Tropical Group Testing

Hsin-Po Wang, Ryan Gabrys, and Alexander Vardy

Abstract—Polymerase chain reaction (PCR) testing is the gold standard for diagnosing COVID-19. PCR amplifies the virus DNA 40 times to produce measurements of viral loads that span seven orders of magnitude. Unfortunately, the outputs of these tests are imprecise and therefore quantitative group testing methods, which rely on precise measurements, are not applicable. Motivated by the ever-increasing demand to identify individuals infected with SARS-CoV-19, we propose a new model that leverages tropical arithmetic to characterize the PCR testing process. Our proposed framework, termed tropical group testing, overcomes existing limitations of quantitative group testing by allowing for imprecise test measurements. In many cases, some of which are highlighted in this work, tropical group testing is provably more powerful than traditional binary group testing in that it requires fewer tests than classical approaches, while additionally providing a mechanism to identify the viral load of each infected individual. It is also empirically stronger than related works that have attempted to combine PCR, quantitative group testing, and compressed sensing.

Index Terms—Group testing, polymerase chain reaction (PCR) testing, tropical arithmetic.

I. INTRODUCTION

COVID-19 pandemic has highlighted the critical role that widely-accessible testing can have in controlling the spread of infectious diseases. Efficient testing schemes have the potential to simultaneously reduce the time to diagnosis while improving both the reliability and accuracy of the testing procedure. This subject has attracted significant attention in literature [1]; however, existing works do not accurately model the semiquantitative information available at the output of the polymerase chain reaction (PCR) testing methods used to detect the presence of SARS-CoV-19.

PCR tests output cycle threshold (Ct) values which, as a result of the testing mechanism itself, are typically represented as semiquantitative measurements in the log domain [2], [3]. Here, the term quantitative refers to the fact that the tests’ readings are non-binary and semi means that the readings are noisy or inaccurate. Previous semiquantitative approaches are ill-suited for modeling the outputs of PCR tests as previous works mostly rely on an implicit assumption that test measurements are reported on a linear (rather than a log) scale [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16].

As an illustration of the potential problem of modeling the PCR test outputs on a linear scale, consider the Ct value of a test as the dB value of a sound wave or the pH value of a liquid. When adding a 50 dB white noise with a 30 dB one, we get a 50.04 dB white noise; that is indistinguishable from 50 dB. When mixing equal volume of pH-1 liquid and pH-3 liquid, we get a result equivalent to combining pH-0.9957 liquid with pure water, where it is hard to tell pH-0.9957 apart from pH-1. Due to the wide range in viral load between infected individuals, the same phenomenon for Ct values has been observed and is often referred to as masking [17], [18], [19].

In order to address the masking issue and also to take advantage of the semiquantitative outputs available from PCR, we propose introducing delays during the DNA amplification process in order to encode extra information. The basic idea will be to generate tests where each of the samples within a test can be inserted at different times. As a simple example of how this would work, suppose we design a test that consists of a single sample from an infected individual that has a Ct value of X. Then, if we delay inserting the sample by δ cycles, the output of the resulting test would be δ + X.

We use tropical multiplication $x \odot y := x + y$ and tropical addition $x \oplus y := \min(x, y)$ to model the behavior of the Ct values that are provided as output of each of the PCR tests. See Table I for a comparison of this model versus others. According to our model, the extra information can be retrieved by matching the pattern of the Ct values against the pattern of the delays. See Figure 1 for an illustration of this process.

In order to address the masking issue and also to take advantage of the semiquantitative outputs available from PCR, we propose introducing delays during the DNA amplification process in order to encode extra information. The basic idea will be to generate tests where each of the samples within a test can be inserted at different times. As a simple example of how this would work, suppose we design a test that consists of a single sample from an infected individual that has a Ct value of X. Then, if we delay inserting the sample by δ cycles, the output of the resulting test would be δ + X.

We use tropical multiplication $x \odot y := x + y$ and tropical addition $x \oplus y := \min(x, y)$ to model the behavior of the Ct values that are provided as output of each of the PCR tests. See Table I for a comparison of this model versus others. According to our model, the extra information can be retrieved by matching the pattern of the Ct values against the pattern of the delays. See Figure 1 for an illustration of this process.

In order to address the masking issue and also to take advantage of the semiquantitative outputs available from PCR, we propose introducing delays during the DNA amplification process in order to encode extra information. The basic idea will be to generate tests where each of the samples within a test can be inserted at different times. As a simple example of how this would work, suppose we design a test that consists of a single sample from an infected individual that has a Ct value of X. Then, if we delay inserting the sample by δ cycles, the output of the resulting test would be δ + X.

We use tropical multiplication $x \odot y := x + y$ and tropical addition $x \oplus y := \min(x, y)$ to model the behavior of the Ct values that are provided as output of each of the PCR tests. See Table I for a comparison of this model versus others. According to our model, the extra information can be retrieved by matching the pattern of the Ct values against the pattern of the delays. See Figure 1 for an illustration of this process. Delaying and matching, hand in hand, leave the masking issue nowhere to stand.

| Regime       | Reading          | Remixing |
|--------------|------------------|----------|
| Binary       | Negative, Positive | Neg \lor Pos = Pos | |
| Tropical     | $2^{-\infty}, 2^{-40}, \ldots, 2^{-12}$ | min(30, 15) = 15 |
| Semiquantitative | $[0, 3], [3, 6], [6, 9], \ldots$ | $[0, 3] + [3, 6] = [3, 9]$ |
| Quantitative | $0, 1, 2, 3, 4, 5, \ldots$ | $8 + 9 = 17$ |

See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
A. Contributions of This Paper

We propose a new model based on two simple techniques—delaying and matching—for handling quantitative measurements that behave like dB values, pH values, or Ct values. With cleverly-crafted delays, we can attack a handful of scenarios as listed below.

- When there is a single \( (D = 1) \) infected person in a population of size \( N \), we can identify her in just \( T = 2 \) nonadaptive tests, where \( N \) can be arbitrarily large (Theorem 10). When delays are limited to \( \ell \) cycles, we show more generally \( N \approx T \ell \log_2 N \) (Theorem 12).
- For the case of \( D = 2 \) infected persons within a population of size \( N \), we give two constructions.
  - The first construction uses \( T \approx 2 \sqrt{N} \) nonadaptive tests (Theorem 15). In this construction, every person is present in only two tests.
  - The second construction uses \( T \approx \log_2(N) + \log_2(\log_2 N) \) nonadaptive tests only (Theorem 19). If the delays are constrained, we can limit it to \( \ell \approx \log_2(N)/\log_2(\log_2 N) \) cycles, and achieve \( T \approx 1.01 \log_2 N \) (Theorem 23). This construction outperforms the information-theoretical bound of binary group testing.
  - For general \( D \), we give one necessary and two sufficient conditions of the existence for group testing schemes (Section VI).
- When adaptive testing is allowed, \( T = 4 \) tests are sufficient to find \( D = 2 \) infected persons among arbitrarily many persons (Theorem 36).
- In general, \( T = 3D + 1 \) adaptive tests are sufficient to locate \( D \) infected persons among arbitrarily many persons (Theorem 41). For this approach, one does not need to know \( D \) beforehand. When delays are limited to \( \ell \) cycles, we show that \( T \approx 4D \log_2 N \) tests suffice (Theorem 42).

- We present simulations that confirm that in many cases matching and delaying outperform related works (Appendix D).
  - Without delaying, matching alone has a comparable sensitivity–specificity tradeoff as existing works.
  - The tradeoff improves as the range of the Ct value increases, i.e., matching works better when the masking issue is supposedly more destructive.
  - Adding random delay improves the tradeoff even further; the greater the variance of the delays, the greater the performance.

All constructions proposed in this paper report the identities of the infected people as well as estimate the extent of infection; we are not sacrificing quantitativity in favor of sensitivity and specificity. The reported Ct values can help further diagnoses [20], [21].

B. Organization of This Paper

Section II reviews PCR and group testing and states the assumptions and goals of tropical group testing. Section III develops optimal strategies to identify a single \( (D = 1) \) infected person. Section IV considers the case of \( D = 2 \) infected persons with minimum pipetting efforts. Section V presents a testing scheme for \( D = 2 \) infected persons with nearly optimal number of tests. Section VI presents necessary and sufficient conditions on non-adaptive testing strategies that hold for general \( D \). Section VII and Section VIII present our proposed adaptive strategies for \( D = 2 \) and general \( D \), respectively. For a detailed breakdown of the organization of the paper and the problems tackled in the respective sections, please see Table II.

II. BACKGROUND AND PROBLEM STATEMENT

In this section, we review the working principles of the PCR process. After that, we review group testing. Next,
we introduce the notion of delays and how this notion is formalized using the tropical semiring. Finally, we conclude this section with our problem statement.

### A. The PCR Process

PCR stands for polymerase chain reaction. It is performed by a machine that holds a collection of tubes, each tube containing some specimens. A PCR machine first heats the tubes up to a temperature $T_h \degree C$; the heat decouples every double-helical DNA molecule into two single-stranded DNA templates. The machine then cools the tubes down to another temperature $T_c \degree C$; at this temperature, the primers (short, single-stranded DNA segments) will attach to the DNA templates, labeling the starting point of DNA replication. Following that, the machine warms the tubes up to an intermediate temperature $T_i \degree C$, which is the working temperature for the polymerases, the enzymes that will be completing the single-stranded DNA templates to form double-stranded DNA molecules. A cycle means that the tubes undergo $T_h \degree C$, $T_c \degree C$, and $T_i \degree C$ once, which implies that the concentration of DNA is doubled. Repeating this process for $r$ cycles increases the concentration of DNA by $2^r$-fold. See Figure 2 for an illustration.

In a *quantitative* PCR\(^1\) test, some fluorescent dyes are added into the tubes; these dye molecules emit light when attaching to DNA. As the process of DNA-amplification continues, more of these fluorescent molecules will attach themselves to the newly-created DNA content. Eventually, the tubes will emit sufficient light that triggers a sensor. When this happens, the current cycle count is reported as the *cycle threshold* (Ct) value of the specimen. Accordingly,

$$Ct = \lfloor -\log_2(\text{viral load}) + \text{constant} \rfloor.$$  

(1)

If all tubes turn out positive or some tubes take too long (40 cycles in practice) to emit a sufficient amount of light, the machine is turned off. For those tubes that did not trigger the sensor, we say the Ct values are $\infty$ and the test results are negative.

According to formula (1), if a sample with Ct value $a$ and another sample with Ct value $b$ are mixed, the mixture will have Ct value $\approx -\log_2(2^{-a} + 2^{-b})$. In practice, it is observed that $a$ and $b$ are almost always not equal. Moreover, as $b - a$ gets larger, it becomes exponentially more difficult to estimate $b$ even if $a$ is precisely known. This is because

$$\frac{\partial}{\partial b} -\log_2(2^{-a} + 2^{-b}) = \frac{2^a}{2^a + 2^b} \approx 2^{a-b}. $$

In this paper, we turn the disadvantage that $\log_2(2^{-a} + 2^{-b})$ is very insensitive to $b$ to an advantage that $-\log_2(2^{-a} + 2^{-b})$ can be approximated by $-\log_2(2^{-a}) = a$. That is, we will use the maximum of the constituent Ct values as a proxy of the true Ct value of the mixture.

For fluorescence-to-cycle plots in real life, see [22], [23], [24], and [25]. For statistics of Ct values, see [18], [26], [27], [28], and [29]. For the relation between Ct values and dilution, see [30], [31], [32], and [33].

---

\(^1\)It is also called *real-time* PCR but still abbreviated as qPCR, not RT-PCR. RT-PCR refers to another technique called reverse transcription PCR. The combination of the two is abbreviated as RT-qPCR, which is the kind that is used in this pandemic.
B. Group Testing

Group testing was introduced by Dorfman [34] and has been of research interest since then. The spirit of group testing is to combine \( q \) specimens, where \( q > 1 \), and then test. If the outcome is negative, all \( g \) specimens are negative; we gain this knowledge by spending only one test, instead of \( g \) tests if we do it individually. If the outcome is positive, at least one specimen is positive; we then test these \( g \) specimens individually to determine who is infected. The latter case costs us \( g + 1 \) tests, which is one more than the \( g \) tests that would have been spent if we do it individually. But if the latter case happens relatively infrequent compared to the former case, we can reduce the number of tests.

More generally, group testing is about recovering an unknown vector \( x \in \{0, 1\}^N \) whose Hamming weight is at most \( D \) by asking as few “questions” as possible. Each “question” is a zero-one vector \( q \in \{0, 1\}^N \) that indicates who do we want to test, and the test result will be

\[
\bigvee_{j=1}^N (q_j \wedge x_j).
\]

So, for instance, if we take \( q = [1 \ldots 1 \ldots 0] \) for the first question and \( q = [0 \ldots 0 \ldots 1] \) for the second question and so on, we can binary-search the ones in \( x \) [35]. But that might not be the most efficient use of tests and many have worked on this. See Du and Hwang’s books [36], [37], Ngo and Rudra’s lecture notes [38], and Aldridge, Johnson, and Scarlett’s survey [39] for more. See [17], [40], [41], [42], [43], and [44] more recent works that are not covered by surveys.

Group testing comes with a variety of flavors. The survey [39, Section 1.1] provides the following categorization of existing group testing works:

- Adaptive vs nonadaptive.
- Zero error probability vs small error probability.
- Exact recovery vs partial recovery.
- Noiseless vs noisy testing.
- Binary vs non-binary outcomes.
- Combinatorial vs iid prior.
- Known vs unknown number of defectives.

The classification of the various scenarios studied throughout this paper is listed in Table II.

We next discuss group testing models that involve non-binary test outcomes, which involves quantitative group testing as well as our tropical group testing model.

The earliest form of non-binary group testing dates back to the 1940s, when balance scales were evaluated as a tool to single out counterfeit coins [45], [46], [47]. A variant of this coin-weighing problem considers the usage of pointer scales to count how many counterfeit coins are in the queried pool. That is, given \( q \in \{0, 1\}^N \) that indicates which coins are to be weighed, the pointer scale outputs

\[
\sum_{j=1}^N (q_j \cdot x_j). \tag{2}
\]

This problem later became known as quantitative group testing and many of the techniques and results have found application in, say, additive multiple access channels [48], [49]. In a very similar field called compressed sensing, \( q \in \mathbb{R}^N \) allows more freedom when asking questions; but at the same time \( x \in \mathbb{R}^N \) becomes more challenging to recover.

On a parallel track, threshold group testing considers the setup where each test outputs “positive” when the number of counterfeit coins exceeds a certain threshold \( \theta \), and it outputs “negative” otherwise [50], [51], [52]. That is, the test outputs whether

\[ \sum_{j=1}^N (q_j \cdot x_j) < \theta. \]

or not, for some known \( \theta \).

A unification of quantitative and threshold group testing, which has received recent attention in [17], [43], [53], [54], and [55] due to its applications to genomic sequence processing, is known as semiquantitative group testing. In semiquantitative group testing, the positive real line are partitioned into intervals

\[ [0, \theta_1), [\theta_1, \theta_2), [\theta_2, \theta_3), \ldots \]

and each test outputs an interval that contain the true value (2). Previous works considered the setup where the endpoints \( \{\theta_j\} \) represent an arithmetic progression. As will be discussed later in the next subsection, motivated by the PCR testing method, we consider a semiquantitative setup where the endpoints \( \{\theta_j\} \) are a geometric progression. See Table I for a summary of existing semiquantitative group testing regimes.

### Table II

| Classification                  | Section | Appendix |
|--------------------------------|---------|----------|
| nonadaptive vs adaptive         | III     | A        |
| error probability               | IV & V  | B        |
| exact vs partial recover         | VI      | C        |
| noisy vs noiseless               | VII     | D        |
| tropical vs quantitative        | A       |          |
| combinatorial vs iid            | B       |          |
| known # of infected             | C       |          |
| free vs limited delay           | D       |          |

main theorem within

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 10 | 15 | 32 | 36 | 41 | 12 |

C. Delay and Tropical Semiring (Aka Min-Plus Algebra)

In this work, we consider a special group testing scenario whereby one adds specimens into each tube at different cycles. Consider the following example procedure that is also illustrated in Figure 1:

- First, we place empty tubes A and B into the PCR machine.
- Next, we put specimens X and Z into tubes A and B, respectively.
- Run the machine for 1 cycle.
• Put specimen Y into tubes A and B.
• Run the machine for 1 cycle.
• Put specimens Z and X into tubes A and B, respectively.
• And finally, run the machine for another 38 cycles or until the detection of light signal, whichever is earlier.

Unlike traditional group testing—which is only concerned with how to distribute specimens into tubes—this paper is also concerned with the question of when specimens should be placed into their respective tubes.

In Figure 1, since specimen Y is put into the tubes after one cycle has elapsed, we say that specimen Y is delayed by 1 cycle. More generally, if a specimen is inserted into a tube after δ cycles have elapsed, we say that the specimen is delayed by δ cycles. If a specimen is never put into a tube, we say that the specimen does not participate in this tube and the corresponding delay is infinity.

We now recall a mathematical framework that will be used to describe the behavior of Ct values under pooling and delaying. The real numbers with positive infinity, we say that the specimen does not participate in this tube and the corresponding delay is infinity.

We now recall a mathematical framework that will be used to describe the behavior of Ct values under pooling and delaying. The real numbers with positive infinity, \( R \cup \{ \infty \} \), with operators \( \oplus \) and \( \circ \), as defined by

\[
\begin{align*}
    x \oplus y &:= \min(x, y) \quad \text{(in particular } x \oplus \infty := x), \\
    x \circ y &:= x + y \quad \text{(in particular } x \circ \infty := \infty),
\end{align*}
\]

is called the tropical semiring or the min-plus algebra. The tropical semiring captures the behavior of addition and multiplication through the lens of \( \log_q \) for some large \( q \), and finds applications in algebraic geometry and combinatorics [56], [57]. As an example, the core of the Floyd–Warshall algorithm (ibid. or [58]) is tropical matrix multiplication, \( A \circ B \), whose \((i, k)\)th entry is defined to be

\[
\bigoplus_{j=1}^{N} (A_{ij} \circ B_{jk}) = \min_{j=1}^{N} (A_{ij} + B_{jk}).
\]

As will be justified by simulations in Appendix D, we use tropical arithmetic to model the Ct values of pooled and delayed specimens.

**Assumption 1 (Tropical Model):** The mixture of specimens with Ct values \( x_1, x_2, \ldots, x_N \), each delayed by \( \delta_1, \delta_2, \ldots, \delta_N \) cycles, respectively, has Ct value

\[
\bigoplus_{j=1}^{N} (\delta_j \circ x_j) = \min_{j=1}^{N} (\delta_j + x_j).
\]

When the context is clear, we write \( \delta \circ x \) to denote a tropical vector–vector multiplication. When there are multiple tests, we write \( S \circ x \) to denote a tropical matrix–vector multiplication. We call \( S \) a schedule matrix.

**Remark 2:** In traditional group testing, the test outcome is either positive or negative. One can, say, treat Ct value 39 or lower as a positive result and Ct value 40 or higher as a negative result. We then have

\[
\left( \bigoplus_{j} x_j \right) < 40 \text{ if and only if } \bigvee_{j} (x_j < 40),
\]

which witnesses how tropical group testing recovers binary group testing. More generally, we always have

\[
\bigoplus_{j} x_j < \theta \text{ if and only if } \bigvee_{j} (x_j < \theta)
\]

regardless which \( \theta \) we have in mind, which witnesses the fact that tropical group testing is conducting binary group testing simultaneously with all thresholds.

**D. Problem Statement**

The goal of this paper is to construct tropical group testing schemes that allow us to diagnose a population efficiently. The following two definitions state our goal precisely.

**Definition 3 (Nonadaptive Testing):** A \((T, N, D)\)-tropical code is a schedule matrix \( S \in (\{0\} \cup N \cup \{\infty\})^{T \times N} \) such that, for any two distinct vectors \( x, y \in (\{0\} \cup N \cup \{\infty\})^{N \times 1} \), each having at most \( D \) finite entries,

\[
S \circ x \neq S \circ y.
\]

A tropical code is said to be within maximum delay \( \ell \) if \( S \in \{0, 1, \ldots, \ell, \infty\}^{T \times N} \).

**Definition 4 (Adaptive Testing):** An \( R\)-\((T, N, D)\)-tropical protocol is a series of \( R \) functions, \( S^{(1)}, S^{(2)}, \ldots, S^{(R)} \), that take earlier test results as inputs and output variable-height schedule matrices

\[
\begin{align*}
    S^{(1)} &= S^{(1)}(A), \\
    S^{(2)} &= S^{(2)}(S^{(1)} \circ x), \\
    S^{(3)} &= S^{(3)}(S^{(2)} \circ x), \\
    &\vdots \\
    S^{(R)} &= S^{(R)}(\left[ \begin{array}{c}
        S^{(1)} \\
        \vdots \\
        S^{(R-1)}
    \end{array} \right] \circ x)
\end{align*}
\]

such that (i) the numbers of rows may depend on past results but the total is \( \leq T \), (ii) the numbers of columns are \( N \), and (iii) the final result

\[
\left[ \begin{array}{c}
    S^{(1)} \\
    \vdots \\
    S^{(R)}
\end{array} \right] \circ x
\]

is unique among all \( x \in (\{0\} \cup N \cup \{\infty\})^{N} \) having at most \( D \) finite entries. A tropical protocol is said to be within maximum delay \( \ell \) if the schedule matrixes use only the alphabet \( \{0, 1, \ldots, \ell, \infty\} \).

**Remark 5:** Here, \( T \) is the number of tests, \( N \) is the number of persons waiting to be tested, and \( D \) is an upper limit on the number of infected persons.

**Remark 6:** When the maximum delay is set to \( \ell = 0 \), a tropical code is just a \( T \times N \) matrix that is \( D \)-separable in the group testing context (after replacing \( 0 \) and \( \infty \) by 1 and 0, respectively). Section VI will discuss this in detail. Similarly, a tropical protocol within maximal delay \( \ell = 0 \) is just the usual adaptive group testing, where \( R \) is the number of rounds.

**Remark 7:** Our setup is more demanding than the traditional group testing setup in that we will be requiring not only
the identities of the infected individuals but also the extent of infection of each infected individual.

Remark 8: When compared to quantitative group testing, our setup is also resilient to the fuzziness innate to PCR testing in that (i) our decoder makes decisions based on integer (instead of real number) Ct values and (ii) every time two specimens are mixed together, the one with fewer virus particles are forgotten. This harsh rule forces us to find workarounds that must not rely on the type of arithmetic that "subtracts 50 dB from 50.04 dB to obtain 30 dB".

Remark 9: The theorems proved in this paper hold true even if we replace the alphabet \( \{0\} \cup \mathbb{N} \cup \{\infty\} \) of the Ct values by \( \mathbb{R} \cup \{\infty\} \). This makes tropical group testing instantly adapted to scenarios with non-integer Ct values. In the latter scenarios, since tropical arithmetic retains the meaning after rescaling, the alphabet \( \{0, 1, \ldots, \ell\} \) of the delays shall be understood as the multiples of the minimum difference of the Ct values that we can tell apart.

The next section is a warmup with the \( D = 1 \) case.

### III. ONE INFECTION AND DIFF-LAYS

This section considers the simplest case whereby at most one person is infected. It demonstrates how tropical group testing can find this infected person using only two tests, regardless of how many people are taking the tests. The key concept in the section is diff-lay, which is the difference \( S_{ij} - S_{uj} \) of delays in the same column of the schedule matrix.

#### A. Two Tests on Three People—a Toy Example

Recall the testing scheme given in Section II-C and Figure 1. The scheme can be summarized by the following schedule matrix

\[
\begin{bmatrix}
 a \\
 b \\
\end{bmatrix} := 
\begin{bmatrix}
 0 & 1 & 2 \\
 2 & 1 & 0 \\
\end{bmatrix} \circ 
\begin{bmatrix}
 x \\
 y \\
 z \\
\end{bmatrix}
\]

(3)

It describes a \((2, 3, 1)\)-tropical code. Instead of delaying by 1 and 2 cycles, one can delay by any positive cycles, including infinity. A larger delay is safer when Ct values are noisy; so one might want to max out the available delays.

Here is how to decode matrix (3): If all three individuals are healthy, then both tubes will report negative results. If X is infected, then we will see \( a-b = -2 \) and we can infer her Ct value \( x = a \). If Y is infected, we will see \( a-b = 0 \) and infer her Ct value \( y = a \). Similarly, if Z is infected, we will see \( a-b = 2 \) and infer \( z = b \). Figure 3 visualizes how to interpret the test results.

The difference of delays, for instance \( a-b = -2 \) for specimen X, is called the diff-lay and will keep playing a central role in the sequel. The next subsection generalizes the \((2, 3, 1)\)-tropical code by generating many more diff-lays.

#### B. Two Tests on Many People—General \((2, N, 1)\)-Tropical Code

An example of a \((2, 4, 1)\)-tropical code is

\[
\begin{bmatrix}
 0 & 0 & 2 & 5 \\
 5 & 2 & 0 & 0 \\
\end{bmatrix}
\]

(4)

where the possible diff-lays are \( a - b \in \{-5, -2, 2, 5\} \). An example of a \((2, 5, 1)\)-tropical code is

\[
\begin{bmatrix}
 0 & 0 & 0 & 3 & 7 \\
 7 & 3 & 0 & 0 & 0 \\
\end{bmatrix}
\]

(5)

Learning from the examples, we know we can test as many people as we want as long as the delays can keep up. Say there is a cap on delays, which we denote by \( \ell \). Then it is possible and optimal to test \( 2\ell + 3 \) persons at once via the following schedule matrix.

\[
\begin{bmatrix}
 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 2 & \cdots & \ell-1 & \ell & \infty \\
 \ell & \ell & \ell-1 & \cdots & 2 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
\end{bmatrix}
\]

(6)

Notice how each column has a unique diff-lay and they exhaust all possible diff-lays from \(-\infty, -\ell, -\ell+1 \) to \( \ell-1, \ell, \infty \). Alternatively, one might prefer the following schedule matrix

\[
\begin{bmatrix}
 0 & 1 & 1 & 2 & \cdots & \ell-1 & \ell-1 & \ell & \ell & \ell & \infty \\
 \ell & \ell & \ell-1 & \ell-1 & \ell & \ell & \ell & \ell & \ell & \ell & \ell \\
\end{bmatrix}
\]

(7)

because the pipetting works are spread out over time (at most four pipets per cycle). The following theorem formalizes this idea and generalizes Figure 1 and matrices (4)–(7).

**Theorem 10 (One Patient, Two Tests):** Any schedule matrix \( S \in \{0, \ldots, \ell, \infty\}^{2 \times N} \) that contains no infinite column \( \{\infty\} \) and satisfies

\[
\{ S_{ij} - S_{uj} \mid j \in [N] \} = N
\]

is a \((2, N, 1)\)-tropical code within maximum delay \( \ell \). Every such \( S \) must satisfy \( N \leq 2\ell + 3 \).

**Proof:** Let \( x \in \{0\} \cup \mathbb{N} \cup \{\infty\} \) have at most one finite entry. If \( x \) is entirely infinite (everyone is healthy), \( S \circ x \) will be entirely infinite (both tubes are negative). Next, suppose \( x_j \) is finite for some \( j \). Let

\[
\begin{bmatrix}
 a \\
 b \\
\end{bmatrix} := S \circ x.
\]
The diff-lay would then be \( a - b = (S_{1j} + x_j) - (S_{2j} + x_j) = S_{1j} - S_{2j} \) and by that diff-lay we can uniquely determine \( j \).
Once \( j \) is known, we can compute her Ct value via \( x_j = a - S_{1j} \) or \( x_j = b - S_{2j} \).

\( N \) must be less than or equal to \( 2\ell + 3 \) because all possible diff-lays are \( -\infty, -\ell, -\ell + 1, \ldots, \ell - 1, \ell, \infty \). Without uniqueness of diff-lays, columns having the same diff-lay are interchangeable.

**Corollary 11:** Matrices (6) and (7) are both \((2, 2\ell + 3, 1)\)-tropical codes within maximum delay \( \ell \) that attain the equality \( N = 2\ell + 3 \).

What is good about schedule matrix (6), the algebraic formulas read
\[
\frac{\ell}{2} + 2 + \frac{\ell}{2} = \ell + 2.
\]
If \( x_j = \min(a, b) \) (unless \( a = b = \infty \)), which means that everyone is uninfected. For schedule matrix (7), the algebraic formulas read
\[
\frac{\ell}{2} + 2 + \frac{\ell}{2} = \ell + 2.
\]

In Theorem 10, \( \ell \) scales linearly in \( N \). As for how to handle the \( D = 1 \) case when \( \ell \ll N \), Appendix A proves the following generalization of Theorem 10.

**Theorem 12 (One Patient):** There is a \((T, N, 1)\)-tropical code within maximum delay \( \ell \) iff \( N \leq (\ell + 2)^T - (\ell + 1)^T \).

In the upcoming sections, we consider the setup where more than a single person can be infected.

### IV. Two Infections and Minimum Pipetting

In the next two sections, we set the number of infected persons to be at most \( D = 2 \). For this section, we impose an extra constraint that each patient should participate in exactly two tubes. This constraint is to minimize the pipetting works that consume labor and suffer from reproducibility issues. For more on this constraint, see\(^3\) [59] and [60].

First, let us walk thorough an \( N = 4 \) example, after which we will give a general construction.

#### A. A 4-Cycle Example

The goal of this subsection is to identify two infected persons using four tests on four persons, i.e., to find a \((4, 4, 2)\)-tropical code. Note that there exists a trivial \((4, 4, 4)\)-tropical code that tests each person individually. Our goal here, a \((4, 4, 2)\)-tropical code, is not meant to be efficient, but to become the building block of a design based on bipartite graphs.

Call the four tubes A, B, C, and D; call the four persons W, X, Y, and Z. Let the lower letters \( a, b, c, d, w, x, y, \) and \( z \) be the corresponding Ct values.

**Encoding:** Prepare the pools as the following.

\[
\begin{bmatrix}
    a \\
    b \\
    c \\
    d \\
\end{bmatrix} =
\begin{bmatrix}
    7 & 0 & \infty & \infty \\
    \infty & 7 & 0 & \infty \\
    \infty & \infty & 7 & 0 \\
    0 & \infty & \infty & 7 \\
\end{bmatrix}
\begin{bmatrix}
    w \\
    x \\
    y \\
    z \\
\end{bmatrix}
\]

This can be graphically paraphrased by Figure 4 (left), in which a vertex is a tube and an edge is a person. An edge's specimen is first put in the arrow tail and then, after 7 cycles, added into the arrow head.

\(^3\)We thank the anonymous reviewer for providing references.
B. A 3-Cycle Non-Example

Knowing how to deal with 4-cycles, we now show that a tropical code cannot contain 3-cycles.

Lemma 14 (No 3-Cycle): There does not exist a tropical code where three persons participate in three tests in total, each person is in two tests, and each test contains two persons.

Proof: Suppose the part of the schedule matrix that concerns the three persons and three tests is
\[
S' = \begin{bmatrix}
\delta & \varepsilon & \infty \\
\zeta & \infty & \eta \\
\infty & \theta & \kappa
\end{bmatrix}
\]
where \(0 \leq \delta, \varepsilon, \zeta, \eta, \theta, \kappa \leq \ell\). Then we cannot distinguish
\[
S' \odot \begin{bmatrix}
2\ell - \xi \\
0 \\
\infty
\end{bmatrix} = \begin{bmatrix}
\varepsilon \\
2\ell \\
0
\end{bmatrix} = S' \odot \begin{bmatrix}
\infty \\
0 \\
2\ell - \eta
\end{bmatrix}
\]
and hence this cannot be a tropical code. \(\square\)

Lemmas 13 and 14 implies that, in order to find a tropical code, we merely want an explicit number. We will treat the blocks as persons. For a test \(t\), we will let \(B_t\) be a bijection that associates blocks with id \(t\) or almost equal partition sizes [61]. These graphs have the potential to give rise to good tropical codes.

Before actually constructing any tropical code out of a complete bipartite graph \(K_{p, q}\), we demonstrate how to encode the delays. Let \(B\) be a \(p \times q\) matrix. Consider the edge that connects \(u \in [p]\) to the left to \(v \in [q]\) to the right. If the specimen is first put in left \(u\) and, after \(\delta\) cycles, in right \(v\), let \(B_{uv}\), be \(\delta\). If the specimen is first put in right \(v\) and, after \(\varepsilon\) cycles, in left \(u\), let \(B_{vu}\) be \(\varepsilon\). This is how we can represent the diff-lays on a bipartite graph using a weighted biadjacency matrix \(B\).

As an example,
\[
B = \begin{bmatrix}
7 & -7 \\
-7 & 7
\end{bmatrix}
\]
represents Figure 4 (right). In a biadjacency matrix representation, the cycle-sum condition becomes whether every \(2 \times 2\) sub-matrix \(\begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}\) satisfies \(\delta - \varepsilon + \zeta - \eta \neq 0\).

This begs the question of whether there exist a systematic way to fill-in larger biadjacency matrices to meet said condition. The answer is positive.

Theorem 15 (Two Patients, min Pipetting): Let \(T \geq 2\). Let \(p \geq T/2\) be an odd prime. Then there exist a graph-based \((T, [T/2])\)-tropical code within maximum delay \((p - 1)/2\).

Proof: Treat the multiplication table
\[
\begin{bmatrix}
1 \cdot 1 \mod p & 1 \cdot 2 \mod p & \ldots & 1 \cdot (p - 1) \mod p \\
2 \cdot 1 \mod p & 2 \cdot 2 \mod p & \ldots & 2 \cdot (p - 1) \mod p \\
\vdots & \vdots & \ddots & \vdots \\
p \cdot 1 \mod p & p \cdot 2 \mod p & \ldots & p \cdot (p - 1) \mod p
\end{bmatrix}
\]
as the biadjacency matrix of a weighted directed bipartite graph. Or use part of the multiplication table when \(p > T/2\). To minimize the delay, use integers between \(\pm (p - 1)/2\) to represent the residue classes modulo \(p\). To verify that the cycle sum along any 4-cycle is nonzero, note that every \(1 \times 2\) sub-matrix has the structure \([1 \ 0 \ 1 \ 0]\). Hence the cycle sum is congruent to \((b - c)(t - u) \neq 0 \mod p\).

According to Theorem 15, there exist tropical codes such that \(T \approx 2\sqrt{N}\) and \(\ell \approx \sqrt{N}\) for large \(N\).

VI. TWO INFECTIONS AND BINARY IT BOUND

This section again considers the \(D = 2\) case. However, in this section we allow people to participate in more than two tests. The goal of this section is to show that under this setup, there exist tropical codes that require less tests than the information-theoretical bound for binary group testing.

We first go over an \((11, 66, 2)\)-example. Then we state a general sufficient condition of the existence of codes. We provide a construction that satisfies this condition. Finally, we briefly discuss the possibility of using concatenation to conserve the delays.

A. An \((11, 66, 2)\)-Example

For any positive integer \(T\), let \([T]\) be \(\{1, 2, \ldots, T\}\). A block design \(\mathcal{H} \subseteq 2^{[T]}\) is a family of subsets of \([T]\). An element of \(\mathcal{H}\) is called a block.

Let us take \(T = 11\) tests as an example. Let \(\mathcal{H} \subseteq 2^{[T]}\) be a block design such that, for any two blocks \(B, Z \in \mathcal{H}\), we have \(|Z \setminus B| \geq 2\). It can be shown that \(|\mathcal{H}| \leq 66\) and the equality holds when \(\mathcal{H}\) is the (unique up to isomorphism) \((4, 5, 11)\)-Steiner system [63, Section 6]. A \((t, k, n)\)-Steiner system, generally speaking, is a block system where each block is a \(k\)-element block of \([n]\) such that any \(t\)-element subset of \([n]\) is contained in one and only one block. For more on Steiner systems, see the standard textbook [64] and a recent breakthrough [65]. For now, fix \(\mathcal{H}\) to be the \((4, 5, 11)\)-Steiner system and \(N = 66\).

We will treat \(\mathcal{H}\)’s underlying set \([T]\) as the set of tests, and treat the blocks as persons. For a test \(t \in [T]\) and a block \(B \in \mathcal{H}\), we will let \(B_t\) join the \(t\)th test if \(t \in B\). Let \(j: \mathcal{H} \rightarrow [66]\) be a bijection that associates blocks with id numbers. Let \(k: \mathcal{H} \rightarrow [37]\) be a coloring of blocks\(^4\) such that \(k(B) \neq k(Z)\) whenever \(|B \cap Z| \geq 2\). Consider a schedule matrix
\[
S_{tj(B)} := \begin{cases}
t \cdot k(B) \mod 37 & \text{if } t \in B, \\
\infty & \text{if } t \notin B.
\end{cases}
\]

Regarding the performance of \(\mathcal{H}\) and \(S\), we have the following.

\(^4\)We found \(k\) by a computer program that runs a greedy algorithm. The optimality of 37 is not the concern here. We merely want an explicit number.
Proposition 16: Schedule matrix 8 is an (11,66,2)-tropical code within maximum delay 36.

Before giving the proof of Proposition 16, we make the following two observations.

Observation 17: For any three distinct individuals $A,B,Z \in \mathcal{H}$, we can differentiate the case where individuals $A,B$ are infected from where $A,Z$ are infected.

Proof of Observation 17: First notice that if $B \backslash A \neq Z \backslash A$, then we can easily distinguish between the two cases based upon the set of tests that contain infected samples. Suppose therefore that $B \backslash A = Z \backslash A$. Since $|B \backslash A| \geq 2$, we further assume that $\{t,u\} \subset B \backslash A$.

Let $x_j$ be the Ct value of the $j$th person. Let $c_t$ be the Ct value of the $t$th tube. When $A,Z$ are infected,

$$c_t - c_u = (S_{tj}(Z) + x_j(Z)) - (S_{uj}(Z) + x_j(Z)) = S_{tj}(Z) - S_{uj}(Z) \equiv (t-u)k(Z) \pmod{37}$$

On the other hand, when $A,B$ are infected,

$$c_t - c_u = (S_{tj}(B) + x_j(B)) - (S_{uj}(B) + x_j(B)) = S_{tj}(B) - S_{uj}(B) \equiv (t-u)k(B) \pmod{37}$$

As $(t-u)k(Z) \equiv (t-u)k(B) \pmod{37}$, this will differentiate $B$ from $Z$.

Observation 18: For any four distinct individuals $A,B,Y,Z \in \mathcal{H}$, we can differentiate the case where individuals $A,B$ are infected from where $Y,Z$ are infected.

Proof of Observation 18: Suppose that $Y,Z$ are infected and we hypothesize that $A,B$ are infected. The goal is to reject this hypothesis using the diff-lays.

First notice that if $A \cup B \neq Y \cup Z$, then we can easily distinguish between the two cases based upon the set of positive tests. Suppose therefore that $A \cup B = Y \cup Z$.

We say a tube is dominated by an individual $B$ if $c_t = S_{tj}(B) + x_j(B)$, i.e., the contribution of $B$ is what makes the $t$th test has Ct value $c_t$. Let $I_{AB}$ be the set of tubes that we think are dominated by $A$ but are actually dominated by $Y$. Define $I_{AZ}$, $I_{BY}$, and $I_{BZ}$ similarly. Then $|I_{AY} \cup I_{AZ} \cup I_{BY} \cup I_{BZ}| = |A \cup B| = |A \backslash B| + |B \backslash A| + 1 \geq 1 + 3 = 5$. This implies that one of $|I_{AY}|$, $|I_{AZ}|$, $|I_{BY}|$, and $I_{BZ}$ has cardinality 2 or higher. Suppose $|I_{AZ}| \geq 2$ and $\{t,u\} \subset I_{AZ}$, then

$$c_t - c_u \equiv (t-u)k(Y) \neq (t-u)k(A) \pmod{37}$$

Now that we can distinguish $A$ from $Y$, we can reject the hypothesis that $\{A,B\}$ are the infected persons.

The moment we have rejected all incorrect hypotheses about who is sick using Observations 17 and 18, whatever remains must be the true patients. That yields the proof of Proposition 16.

Proof of Proposition 16: We claim that the following decoder works. Given any $c \in \{(0) \cup \mathbb{N} \cup \{\infty\}\}^T$, if the number of positive tubes is 5, we know there is only one patient and we can infer her identity as well as her Ct value.

If more than 5 tubes are positive, then we know there are two patients. We blindly guess two blocks $A,B \in \mathcal{H}$ and check if $\{A,B\}$ is compatible with $c$. If so, output the guess; if not, start over and make a new guess.

Here is why the decoder, albeit inefficiently, works. Observation 17 shows that if $|\{A,B\} \cap \{Y,Z\}| = 1$, where $Y,Z \in \mathcal{H}$ are the actual patients, there will be a contradiction. Observation 18 shows that if $|\{A,B\} \cap \{Y,Z\}| = 0$, there will also be a contradiction. Hence the claimed decoder will terminate iff $|\{A,B\} \cap \{Y,Z\}| = 2$, i.e., our guess matches the reality. There are a finite number of combinations to be guessed so the decoder must eventually guess correctly. Once we know who are the patients it is straightforward to infer their Ct values. We conclude that $S$ is a valid tropical code.

B. More Tests and Test Takers

One immediately sees that Proposition 16 can be generalized to host more tests to screen larger population.

Theorem 19 (Two Patients, max Participants): Let $\mathcal{H} \subseteq 2^{|T|}$ be a block design with $N$ blocks. Assume $|Z \backslash B| \geq 2$ and $|B \backslash Z| \geq 3$ for all distinct blocks $B,Z \in \mathcal{H}$. Let $j: \mathcal{H} \rightarrow [N]$ be a bijection. Let $p$ be an integer whose prime divisors are $\geq T$. Assume there exists $k: \mathcal{H} \rightarrow [p]$ satisfying $k(B) \neq k(Z)$ whenever $|B \cap Z| \geq 2$. Then

$$S_{tj}(B) := \begin{cases} t \cdot k(B) \pmod{p} & \text{if } t \in B, \\ \infty & \text{if } t \notin B. \end{cases}$$

is a $(T,N,2)$-tropical code within maximum delay $p-1$.

Proof: Suppose there is one infected person, $Z \in \mathcal{H}$, and we guess there is one, $B \in \mathcal{H}$. If $B \neq Z$, then the tubes in $B \backslash Z$ will be negative while we expect them to be positive. Hence we can reject this guess.

Suppose there is one infected person, $Z \in \mathcal{H}$, but we guess there are two, $A,B \in \mathcal{H}$. Without loss of generality, suppose $A \neq Z$. As $|A \backslash Z| \geq 1$, tubes in $A \backslash Z$ will be negative while we expect them to be positive. Hence we can reject this guess.

Suppose there are two infected persons, $Y,Z \in \mathcal{H}$, but we guess there is one, $B \in \mathcal{H}$. If $B \neq Z$, then the tubes in $B \backslash Z$ will be negative while we expect them to be positive. Hence we can reject this guess.

Suppose there are two infected persons, $Y,Z \in \mathcal{H}$, but we guess there are two, $A,B \in \mathcal{H}$. Then depending on $|\{A,B\} \cap \{Y,Z\}| = 1$ or 0, we will see a contradiction due to the same reasoning in Observation 17 or 18, respectively. Either case, we can reject the incorrect guess that $\{A,B\}$ are infected.

The following lemma prepares block designs whose $N$ is satisfactorily large.

Lemma 20 [66, Theorem 1]: For $T \geq w \geq 1$, there exists a block design $\mathcal{H} \subseteq 2^{|T|}$ such that each block has size $w$,

$$|\mathcal{H}| \geq \left\lceil \frac{|T|}{w} \right\rceil,$$

and $|B \backslash Z| \geq 2$ for all distinct blocks $B,Z \in \mathcal{H}$.

Choose $w := \lceil |T|/2 \rceil$ to maximize the number of blocks. The asymptote of $N$ in terms of $T$ follows the asymptote of the central binomial coefficient:

$$N \approx \frac{2^T}{T^{\sqrt{T}/2}}.$$

This implies $T \approx \log_2(N) + 1.5 \log_2(\log_2 N)$. In contrast, binary group testing has $2^T \geq \binom{N}{2} + N + 1$ because the
number of cases cannot exceed the number of test outcomes. That implies $T \approx 2 \log_2(N)$. We therefore conclude that tropical group testing beats the information-theoretical bound for binary group testing (by a factor of 1.99).

For small parameters, a worthy reference is Brouwer’s table of constant-weight codes of minimum distance 4 [67]. See Table III for a copy of the first few terms. It can be seen that $(T, N) = (11, 66)$ is the first time constant-weight codes surpass the information-theoretical bound; this is why we chose this example.

### C. Kronecker-Amplified Constructions

The previous subsection shows that one can easily beat the information-theoretical bounds using delays growing linear in $N$. This subsection wants to limit the usage of delays without having to introduce too many additional tests.

To facilitate the construction, let us practice how to assemble a code out of existing codes. Heuristically speaking, our approach will allow us to generate new codes from previously constructed codes without increasing the required delay. Let $S$ be a $(t, n, 2)$-tropical code within maximum delay $n$. (For instance, schedule matrix (8) as a $(11, 66, 2)$-tropical code.) Consider the Kronecker product

$$S \otimes 1_{1 \times n}$$

as a $t \times n^2$ schedule matrix, where $1_{1 \times n}$ is the all-one matrix of dimension $1 \times n$. Notice that the first $n$ columns of this matrix are identical, the next $n$ columns of it are identical, and so on. That is to say, this matrix first gathers every $n$ persons into one pool and then tests the resulting $n$ pools using $S$.

If everyone participating the tests is healthy, then all tests will turn out negative. Suppose one person, with id $Z$, $0 \leq Z \leq n^2 - 1$, is infected with Ct value $z_x$. Then the tests will reveal $[Z/n]$ (that is, the “tens digit” of $Z$ in the $n$-ary expression), and $z_x$, the Ct value of $Z$. If two persons, $0 \leq Y, Z \leq n^2 - 1$, are infected, then the tests will reveal $\{Y/n, Z/n\}$, i.e., the most significant digits of $Y$ and $Z$. There are two cases.

- If the digit differ, i.e., $[Y/n] \neq [Z/n]$, then $S \otimes 1_{1 \times n}$ sees two infected pools and will report $y_x$ and $z_x$.
- If the digit agree, i.e., $[Y/n] = [Z/n]$, then $S \otimes 1_{1 \times n}$ sees one infected pool and will report $\min(y_x, z_x)$.

A similar argument shows that $1_{1 \times n} \otimes S$ reveals $Y \mod n$ and $Z \mod n$, the ones digits of $Y$ and $Z$, together with one Ct values $\min(y_x, z_x)$ (if the ones digits agree) or two Ct values $y_x, z_x$ (if the ones digits differ).

We now investigate what happens when we combine $S \otimes 1_{1 \times n} \otimes 1_{1 \times n} \otimes S$. Define $S^{(2)}$ as this $2t \times n^2$ schedule matrix

$$S^{(2)} := \left[ \begin{array}{c} 1_{1 \times n} \otimes S \\ S \otimes 1_{1 \times n} \end{array} \right]$$

and use it to perform the tests. If there are zero infected persons, all tests will be negative. If there is one infected person, then both $1_{1 \times n} \otimes S$ and $S \otimes 1_{1 \times n}$ will report one infected pool; we can infer the index of the infected person by combining the two digits.

Hereafter, assume that there are two infected persons, $0 \leq Y, Z \leq n^2 - 1$. Let their Ct values be $y_x$ and $z_x$. Let $y_{2n} + y_1$ and $z_{2n} + z_1$, where $0 \leq y_1, z_1, y_2, z_2 \leq n - 1$, be the $n$-ary expansions of $Y$ and $Z$, respectively. Then the first $t$ tests can tell us $\{y_1, z_1\} \subseteq \{0, \ldots, n - 1\}$. More precisely, the first $t$ tests tell us $\{(y_1, y_2), (z_1, z_2)\}$ when $y_1 \neq z_1$ and tell us $\{(z_1, \min(y_1, z_1))\}$ when $y_1 = z_1$. Similarly, the last $t$ tests can tell us $\{(y_2, y_1), (z_2, z_1)\}$ when $y_2 \neq z_2$ and tell us $\{(z_2, \min(y_2, z_2))\}$ when $y_2 = z_2$. Now we are only missing one bit of information: we need to differentiate between the case where

$$\{Y, Z\} = \{y_{2n} + y_1, z_{2n} + z_1\}$$

and the case where

$$\{Y, Z\} = \{y_{2n} + z_1, z_{2n} + y_1\}.$$ 

When $y_1 = z_1$, we already can infer $\{Y, Z\} = \{z_1 + y_2, z_1 + y_2\}$. When $y_2 = z_2$, similarly, we already can infer $\{Y, Z\} = \{y_2, z_2\}$. If $\{(y_1, z_1)\} = \{(y_2, z_2)\} = 2$ and $Y$ and $Z$ have different Ct values, we will see that $y_1$ and $y_2$ associate to a Ct value $y_x$, whilst $z_1$ and $z_2$ associate to another different Ct value $z_x$. That will help us link $y_1$ to $y_2$ and $z_1$ to $z_2$ and thus help us recover $Y, Z$. One difficult case remains to be addressed, namely $\{(y_1, z_1)\} = \{(y_2, z_2)\} = 2$ yet $y_x = z_x$.

To overcome the last case, we add one more test to make up the missing information: Let $p$ be the smallest prime $\geq \max(3, n)$. Let $0 \leq \alpha_1, \alpha_2, \beta_1 \leq p - 1$ be distinct numbers modulo $p$. Compute, for any $0 \leq \alpha_1, \alpha_2 \leq n - 1$,

$$[b_1] := \left[ \begin{array}{c} 1 & \beta_1 \\ 1 & \alpha_1 \end{array} \right]^{-1} \left[ \begin{array}{c} \alpha_1 \\ \alpha_2 \end{array} \right] \mod p$$

using modulo $p$ arithmetics. That is, we find $b_1$ such that $(\alpha_1, b_1)$ and $(\alpha_2, b_2)$ are point-evaluation pairs of some degree-one polynomial. In other words, they are colinear on the plane.

Treating $b_1$ as a function in $\alpha_1, \alpha_2$, we want to append $b_1(\alpha_1, \alpha_2)$ at the bottom of the $(\alpha_1 + \alpha_2 n)$th column of $S^{(2)}$. That is, we stack $S^{(2)}$ on top of this $1 \times n^2$ matrix

$$Q^{(2)} := \left[ \begin{array}{c} 1 & \beta_1 \\ 1 & \alpha_1 \end{array} \right]^{-1} \left[ \begin{array}{c} 1_{1 \times n} \otimes M \\ M \otimes 1_{1 \times n} \end{array} \right] \mod p$$

where

$$M := \left[ \begin{array}{cccc} 0 & 1 & \cdots & n - 1 \end{array} \right].$$

| $T$  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|----|----|----|----|----|----|----|----|----|----|
| bipartite | 16 | 20 | 25 | 30 | 36 | 42 | 49 | 56 | 64 |
| modulo   | 9  | 14 | 26 | 42 | 77 | 132| 246| 429| 805|
| const    | 14 | 18 | 36 | 66 | 132| 166| 325| 585| 1170|
| info     | 22 | 31 | 44 | 63 | 89 | 127| 180| 255| 361|
We claim that the appending \( Q^{(2)} \) to \( S^{(2)} \) results in a valid tropical code.

**Proposition 21:** If \( S \) is an \((t, n, 2)\)-tropical code within maximum delay \( n \), then
\[
\begin{bmatrix}
S^{(2)} \\
Q^{(2)}
\end{bmatrix}
\]
is an \((2t+1, n^2, 2)\)-tropical code within maximum delay \( p-1 \), where \( p \) is the least prime \( \geq \max(3, n) \).

**Proof:** When (i) there is \( \leq 1 \) infected person, (ii) there are two infected persons sharing a same digit, or (iii) there are two infected persons having different \( C_t \) values, the first \( 2t \) tests suffice. When the two infected persons have distinct digits yet the same \( C_t \) values, we utilize the last test in the following manner.

Let \( Y \) and \( Z \) be the infected persons, \( 0 \leq t, Z \leq n^2-1 \). Let \( y_2 y_n + y_1 \) and \( z_2 z_n + z_1 \) be the \( n \)-ary expansion of \( Y \) and \( Z \). The result of \( Q^{(2)} \), denoted by \( c_1 \), is \( c_1 = \min(b_1(y_1, y_2) + y_1, b_1(z_1, z_2) + z_1) \). Therefore \( c_1 - z_1 \in \{b_1(y_1, y_2), b_1(z_1, z_2)\} \). Now there are five points on the plane
\[
(\alpha_1, y_1), (\alpha_2, y_2), (\alpha_1, z_1), (\alpha_2, z_2), (\beta_1, c_1 - z_1).
\]

We know the point \((\beta_1, c_1 - z_1)\) is either \((\beta_1, b_1(y_1, y_2))\) or \((\beta_1, b_1(z_1, z_2))\). It must be colinear with two of the four points to its left. By looking at which two are colinear with it, we can correctly link \( y_1 \) to \( y_2 \) and \( z_1 \) to \( z_2 \). This recovers \( Y \) and \( Z \) and finishes the proof.

Next we define the notion of BITE.

**Definition 22:** A \((T, N, 2)\)-tropical code is said to beat the information-theoretical estimate (BITE)\(^5\) if \( N^2 \geq 4 \cdot 2^T \).

BITing is slightly stronger than simply beating the binary bound (by about one test), but it is an inductive invariant: if \( S \) is an \((t, n, 2)\)-tropical code that BITes, then the \((2t+1, n^2, 2)\)-tropical code in Proposition 21 also BITes because \( N^2 \geq (n^2)^2 - 2^T \). The following theorem generalizes the discussion above, of which the full proof is given in Appendix B.

**Theorem 23 (Two Patients, No Patience):** Let \( S \) be a \((t, n, 2)\)-tropical code within maximum delay \( n \), then there is a \((kt+2k-3, n^2, 2)\)-tropical code within maximum delay \( q-1 \), where \( q \geq \max(3k-3, n) \) is a prime power. If \( S \) BITes, so do the putative codes BITE for all \( k \geq 1 \).

**Remark 24:** This Kronecker-based construction is inspired by [68, Theorem 1] (69) and the way it is used in the paper. This shares elements with the grid-based construction [70] that is used to attack the pandemic [33], [71], [72]. This also shares common elements with the fast decoder approach [73]. Other fast decoder approach such as [40] and [74] also contain similar ideas.

We can use very large but almost equal parameters \( q \approx n \approx 3k - 3 \) by consulting Theorem 23 Lemma 20. Hence, as \( q \) goes to infinity, there exist \((T, N, 2)\)-tropical codes within maximum delay \( q \), where \( q \approx 3 \log_2(N)/\log_2(3 \log_2(N)) \) and \( T \approx 1.01 \log_2(N) \).

VI. MANY INFECTIONS AND DISJUNCTION

In this section, we move on to general \( D \). Concerning the existence of tropical codes, we will prove one necessary condition and two sufficient conditions.

Given a schedule matrix \( S \in \{0\} \cup \mathbb{N} \cup \{\infty\} \times \mathbb{N} \), the underlying block design is defined to be a multiset as
\[
\mathcal{H} := \left\{ t \in [T] \mid S_{tj} < \infty \right\},
\]
which registers the tests each person participates in. A nasty edge case is when two individuals participate in exactly the same subset of tests. By letting \( \mathcal{H} \) be a multiset we have \( |\mathcal{H}| = N \). Also by distinct blocks \( B_1, \ldots, B_D \in \mathcal{H} \), we mean that those \( B \)'s originate from distinct individuals. But as subsets of \([T]\) they are not necessarily distinct.

Here is the necessary condition promised. We state the weak version following by the strong version.

**Definition 25:** A block design \( \mathcal{H} \) is said to be \( D \)-disjunct if \( |\mathcal{H}| \geq 1 \) for distinct blocks \( B_1, \ldots, B_D \in \mathcal{H} \).

**Theorem 26:** The underlying block design \( \mathcal{H} \) of a \((T, N, D)\)-tropical code must be \((D-1)\)-disjunct.

**Proof:** Suppose \( \mathcal{H} \) is not \((D-1)\)-disjunct, then there exist distinct blocks \( Z, B_1, \ldots, B_{D-1} \in \mathcal{H} \) such that \( Z \subseteq B_1 \cup \cdots \cup B_{D-1} \). This means that, when the \( B \)'s are severely infected, we cannot tell if \( Z \) is slightly infected or not infected.

**Definition 27:** A block design \( \mathcal{H} \) is said to be \( D \)-uniquely-disjunct if it is \( D \)-disjunct and
\[
\left\{ Z \in \mathcal{H} \mid Z \setminus (B_1 \cup \cdots \cup B_D) = \{t\} \right\} \subseteq 1
\]
for any vertex \( t \in [T] \) and distinct blocks \( B_1, \ldots, B_D \in \mathcal{H} \).

**Theorem 28:** The underlying block design \( \mathcal{H} \) of a \((T, N, D)\)-tropical code must be \((D-1)\)-uniquely-disjunct.

**Proof:** We already see that \( \mathcal{H} \) must be \((D-1)\)-disjunct. Suppose it is not \((D-1)\)-uniquely-disjunct, then there exist distinct blocks \( Y, Z, B_1, \ldots, B_{D-1} \in \mathcal{H} \) such that \( Y \setminus (B_1 \cup \cdots \cup B_{D-1}) = Z \setminus (B_1 \cup \cdots \cup B_{D-1}) \), which contains but one vertex. This means that, when the \( B \)'s are heavily infected, we cannot tell apart if it is \( Y \) or \( Z \) that is infected.

Here are the two sufficient conditions promised.

**Definition 29:** A block design \( \mathcal{H} \) is said to be \( D \)-separable if \( Z_1 \cup \cdots \cup Z_d \neq B_1 \cup \cdots \cup B_e \) for any two different subsets of blocks \( \mathcal{H} \supseteq \{Z_1, \ldots, Z_d\} \neq \{B_1, \ldots, B_e\} \subseteq \mathcal{H} \), where \( d, e \leq D \).

**Theorem 30:** If there is a \( D \)-separable block design \( \mathcal{H} \) with \( N \) blocks on \( T \) vertices, then there exists a \((T, N, D)\)-tropical code within maximum delay \( 0 \).

**Proof:** Use the vanilla schedule
\[
S_{tj}(B) := \begin{cases} 0 & \text{if } t \in B, \\ \infty & \text{if } t \notin B, \end{cases}
\]
where \( j: \mathcal{H} \rightarrow [N] \) is a bijection. To decode, reinterpret every test result as “positive” or “negative”. Use a binary group testing decoder to determine who are infected. For every infected individual \( Z \), the “outcrop” \( Z \setminus (B_1 \cup \cdots \cup B_{D-1}) \)
is never empty because $D$-separability implies $(D - 1)$-disjunct \cite{75}. This further implies that there is at least one tube wherein $Z$ is the only infected participant. The Ct value of this tube is the Ct value of $Z$.

Definition 31: A block design $\mathcal{H}$ is said to be $D$-doubly-disjunct if $|Z \setminus (B_1 \cup \cdots \cup B_D)| \geq 2$ for distinct blocks $Z, B_1, \ldots, B_D \in \mathcal{H}$.

We see\footnote{We thank the anonymous reviewer for providing references.} \cite[Definition 1]{76} and \cite[Definition 8]{77} for a variant of Definition 31, where $|Z \setminus (B_1 \cup \cdots \cup B_D)|$ is demanded to be $\geq$ a constant multiple of $|Z|$.

Theorem 32: If there is a $(D - 1)$-doubly-disjunct block design $\mathcal{H}$ with $N$ blocks on $T$ vertices, there exists a $(T, N, D)$-tropical code.

Proof: We use this schedule

$$S_{ij(B)} := \begin{cases} 2^{l+|B|}T & \text{if } t \in B, \\ \infty & \text{if } t \notin B, \end{cases}$$

where $j$ is a bijection $j: \mathcal{H} \to [N]$. We now verify that this schedule works.

Let $B_1, \ldots, B_D$ be the blocks that we think are infected. Let $Z_1, \ldots, Z_D$ be the blocks that are actually infected. For now, assume that they are all distinct, as otherwise the theorem statement will be similar (perhaps easier) to prove. Let $I_{uv}$ be the subset of tests that we think are dominated by $B_u$ but actually are dominated by $Z_v$, where dominance means $c_t = S_{ij(Z_v)} + x_{ij}(Z_v)$. By that $\mathcal{H}$ is $(D - 1)$-doubly-disjunct,

$$\bigcup_{u \neq \psi} I_{uv} \supseteq \bigcup_{u \neq \psi} B_u \geq 2D.$$

Next, define a bipartite graph $G$ with left part $[D]$ and right part $[D]$; for any $(u, v) \in [D] \times [D]$, connect $u$ to the left to $v$ to the right $|I_{uv}|$ times. Now $G$ has $2D$ vertices and $\geq 2D$ edges, hence it contains a cycle (possibly a 2-cycle). It must be an even cycle (because the graph is bipartite). Let $2\Psi$ be the size of the cycle, and let its vertices be

$$\begin{array}{c}
  u_1 \\
  u_2 \\
  \vdots \\
  u_\Psi \\
\end{array} \begin{array}{c}
  v_1 \\
  v_2 \\
  \vdots \\
  v_\Psi \\
\end{array}$$

where the $u$'s are in the left part and the $v$'s in the right part. Identify $u_{\Psi+1} := u_1$. Every edge involved, be it $u_\psi \rightarrow v_\psi$ or $u_{\Psi+1} \rightarrow v_\psi$, corresponds to a test in $I_{u_\psi v_\psi}$ or in $I_{u_{\Psi+1} v_\psi}$, respectively. Hence we can talk about the test's Ct value, denoted by $c_\psi$ or $d_\psi$, respectively.

Finally, examine the alternating sum along this cycle

$$\sum_{\psi=1}^{\Psi} c_\psi - d_\psi.$$

In our mind, we expect that this is a sum of some of $B$'s diff-lays. But in reality, this is a sum of some of $Z$'s diff-lays. Since the delays are distinct powers of two, there is no way a sum of delays equal to another sum of delays. This shows that we can always reject incorrect guesses. $\Box$

Remark 33: We understand that using exponential delays is rhetorical as the delays already live in the logarithmic realm. This can be avoided. All we need is that the "cycle sums" do not vanish. And so it suffices to use random delays and bound from above the probability they vanish. Since only short cycles are those that are likely to vanish and since there are only polynomially many short cycles, we expect that a $(T, N, D)$-tropical code exists within polynomial delay.

For works that discuss disjunct-ness vs other properties, see \cite{75} and \cite{78}. For how to design block systems with disjunct and/or separable properties, see Kautz–Singleton \cite{79} and the follow-ups \cite{41, 80, 81, 82}.

In the next two sections, we turn our interest to adaptive strategies.

VII. TWO INFECTIONS AND ADAPTIVE STRATEGIES

Recall that the $D = 1$ case was optimally solved in Section III. Recall also that Sections IV and V discussed some nonadaptive strategies of the $D = 2$ case. In this section, we explore adaptive strategies for the $D = 2$ case. We will give a two-round, four-test strategy that finds two infected persons among arbitrarily many. That is, we will construct 2-(4, $N$, 2)-tropical protocols for all $N$. It is rather surprising that, compared to Sections IV and V, allowing a second round saves such a large number of tests.

Recall that an $R$-($T, N, D$)-tropical protocol is an adaptive group testing strategy that uses $R$ rounds and $T$ tests on $N$ individuals to spot $D$ (or less) infected persons. When we have no better bound on the number of rounds, we will speak of $T$-($T, N, D$)-tropical protocol.

A. Five Tests in Three Rounds

Let a population have Ct values

$$x := \begin{bmatrix}
  x_1 \\
  \vdots \\
  x_N \\
\end{bmatrix}$$

where $N$ is the number of persons being tested. For the first round, begin with two tests that imitate matrix (9).

$$\begin{bmatrix}
  a \\
  b \\
\end{bmatrix} := \begin{bmatrix}
  1 & 2 & \cdots & N - 1 & N \\
  N & N - 1 & \cdots & 2 & 1
\end{bmatrix} \odot x$$

If tubes $a$ and $b$ are negative, no one is infected and we are done. Assume the opposite, that $a$ and $b$ are positive. We compute the pointer $j := (a - b + N + 1)/2$ and double-check the $j$th person in the second round

$$\begin{bmatrix}
  c \\
\end{bmatrix} := \begin{bmatrix}
  \infty & \cdots & \infty & 0 & \cdots & N - j & \infty
\end{bmatrix} \odot x,$$

here $0$ is at the $j$th column. If $c$ is positive, we test

$$\begin{bmatrix}
  d^+ \\
  e^+ \\
\end{bmatrix} := \begin{bmatrix}
  0 & \cdots & 0 & 1 & \cdots & N - j \\
  j - 1 & \cdots & 1 & \infty & \cdots & 0
\end{bmatrix} \odot x$$

in the third round. If $c$ is negative, we test

$$\begin{bmatrix}
  d^- \\
  e^- \\
\end{bmatrix} := \begin{bmatrix}
  0 & \cdots & 0 & \infty & \cdots & N - j & \infty \\
  \infty & \cdots & \infty & 0 & \cdots & 0 & \infty
\end{bmatrix} \odot x$$

in the third round. We claim the following.
Proposition 34: Matrices (9)–(12) form a 3-(5, N, 2)-tropical protocol.

Note that schedule matrix (11) is just (6) without the jth person so it can find us the second patient given that j is the first. It remains to show why schedule matrix (12) can find us two infected persons so quickly given that j is not one. A lemma is placed here, before the proof of Proposition 34, to help clarify what we can learn from the test results a and b.

Lemma 35: Given a and b as defined with schedule matrix (9) and \( j := (a-b+N+1)/2 \). Then \( \min_{i \leq j} (x_i + i) = a \) and \( \min_{k > j} (x_k + (N + 1 - k)) = b \).

This lemma encodes the core idea of this section: After a and b, we learn that j is likely to be infected. That is why we query c to check. And even if she comes out healthy, we still know that someone i < j to the left dominates a and someone k > j to the right dominates b. Now i needs one more test to locate; and k needs another test to locate. Those are what d− and e− do, respectively. Note that the following proof does not assume 1 ⩽ j ⩽ N. In fact, 1 ⩽ j ⩽ N is a consequence of this lemma, which is why (10), (11), and (12) are well-defined.

Proof of Lemma 35: By the configuration of the second test, b ⩽ x_i + (N + 1 - k) for all k, which implies \( x_k ≥ b−(N + 1 − k) \). Suppose i is the index that attain the minimum \( a := \min_{i < j} (x_i + i) \). We have 0 = a − x_i − i ⩽ a − b + (N + 1 − i) − i = a − b + N + 1 − 2i = 2j − 2i. This forces 2i ⩽ 2j. As the index that attain the minimum must be ⩽ j, we might as well restrict the domain of the minimum and write a = \( \min_{i ⩽ j} (x_i + i) \). The second statement of the lemma holds by symmetry.

Proof of Proposition 34: Let c be negative and let d−, e− be defined with schedule matrix (12). From the configuration of d− we know \( \min_{i ⩽ j} x_i = d− \). By Lemma 35 we know \( \min_{i < j} x_i = a \). We can replace the domain i ⩽ j with i < j because j is confirmed to be healthy. So far we have collected the following information

\[
\begin{bmatrix}
1 & 2 & \cdots & j-1 \\
0 & 0 & \cdots & 0
\end{bmatrix} \otimes \begin{bmatrix}
x_1 \\
\vdots \\
x_{j-1}
\end{bmatrix} = \begin{bmatrix}
a \\
d−
\end{bmatrix}
\]

and we can infer that one infected person is \( i := a − d− \) with Ct value \( x_i = d− \). By symmetry, the other infected person is \( k := N + 1 − (b − e−) \) with Ct value \( x_k = e− \).

B. Four Tests in Two Rounds

As it turns out, we can superimpose schedule matrices (11) and (12) in a judicious way to optimize test c away. Let

\[
\begin{bmatrix}
d \\
e
\end{bmatrix} := \begin{bmatrix}
0 & 0 & \cdots & \infty & \gamma + 1 & \cdots & \gamma + N - j \\
0 & \gamma + j - 1 & \cdots & \gamma + 1 & \infty & 0 & \cdots & 0
\end{bmatrix} \otimes \mathbf{a}.
\]

where γ is a gargantuan number, e.g., 8(N + a + b).

Theorem 36: Schedule matrices (9) and (13) give a 2-(4, N, 2)-tropical protocol.

Proof: It suffices to show that we can distinguish the following four cases:

- j is indeed infected; everyone else is healthy.
- j is indeed infected; the other patient is < j.
- j is indeed infected; the other patient is > j.

Fig. 5. The configurations space of tests d and e defined with schedule matrix (13). The cloud at the upper right corner is at \((∞, ∞)\).

- j is healthy; one patient is < j and the other > j.

They one-to-one correspond to the following regions of the configuration space of the test results (cf. Figure 5):

- d and e are negative.
- d and e are positive and \( e > γ + d ≥ γ \).
- d and e are positive and \( d > γ + e ≥ γ \).
- d and e are positive and < γ.

The four regions are mutually disjoint and exhaustive so we can and only have to distinguish them. Afterward, the following are how we compute who are the patients and their Ct values for each of the four cases:

\[
\begin{align*}
x_j &= a + b - N - 1 \\
(x_{j−(e−γ−d)}, x_j) &= (d, a + b - N - 1) \\
(x_j, x_{j+(a−γ−e)}) &= (a + b - N - 1, e) \\
(x_{a−d}, x_{N+1−(b−e)}) &= (d, e).
\end{align*}
\]

It is straightforward to derive these formulas. Next, we proceed to how to identify more infected people.

VIII. MORE INFECTIONS AND ADAPTIVE STRATEGIES

Our next goal is to give a construction that finds all D infected persons in 3D + 1 tests regardless of how large D and N are. A pilot construction using 7D + 1 tests is specified in the next subsection.

A. A Search-and-Verify Protocol

Suppose that there are D persons infected among a population of N. Suppose also that we lack the knowledge of D.

To begin, query

\[
[\mathbf{a}] := \begin{bmatrix}
1 & 2 & \cdots & N
\end{bmatrix} \otimes \mathbf{a}.
\]

If a is negative, we conclude that everyone is healthy. We had used \( 1 ⩽ 7 \cdot 0 + 1 \) tests and that is within the budget. This is
the leaf of our recursive algorithm whose definition continues below.

If \( a \) is positive, query
\[
[b] := [N \ N - 1 \ \cdots \ 1] \odot x.
\] (15)
Compute the index \( j := (a - b + N + 1)/2 \). Then query
\[
[c] := [\infty \ j - 1 \ \cdots \ 0 \ \infty \ \cdots \ \infty] \odot x,
\] (16)
where the delay 0 is at the \( j \)th column. Test \( c \) tells us whether \( j \) is really sick or not. If \( c \) ends up positive, \( j \) is infected with \( C \) value \( x_j = c \). We then remove her from the population. Now that there are \( D-1 \) patients left in the remaining \( N-1 \) persons, we start over. Since the number of infected persons decreases, we expect that this recursive algorithm will eventually reach the leaf and return.

If test \( c \) ends up negative, we have a strengthened Lemma 35 that applies to an unknown \( D \).

**Lemma 37:** Let \( a \) and \( b \) be defined with schedule matrices (14) and (15) and \( j := (a - b + N + 1)/2 \). Then \( \min_{i \leq j} \leq j \neq i \leq N + 1 - k \) is equal to \( b \). (Proof is identical to that of Lemma 35 but this time \( x \) need contain more than two patients.)

Being told that \( j \) is healthy, we know that the first \( j-1 \) persons contain at least one infected person and so do the last \( N-j \) persons. We hereby split the population into two halves—[1, \( j-1 \)] and \([j+1, N]\) — and apply the same algorithm to them separately. Since both halves contain less than \( D \) infected persons, the recursive algorithm will eventually reach the leaf and return. The only problem is, How many tests does it consume before termination?

**B. Test Number Analysis**

For every three tests, we spend on the searching matrices (14) and (15) and the confirmation matrix (16), either of the following happens.

- \( a \) and \( b \) indicate that \( j \) is suspicious and \( c \) confirms that she is indeed infected.
- \( a \) and \( b \) indicate that \( j \) is suspicious but \( c \) shows that \( j \) is healthy. We then split the population into two halves, each containing fewer infected persons.

There are \( D \) infected persons so the number of tests spent on the first case, searching and confirming, is exactly 3D. There are \( D-1 \) “gaps” we can split the population at so the number of tests spent on the second case, splitting, is at most 3D - 3. In total, the cost is at most 6D - 3 tests. On top of that, every time we confirm an infected person \( j \) in some interval \([i, k]\), the protocol will then query
\[
[1 \ 2 \ \cdots \ k - i] \odot \begin{bmatrix}
  x_i \\
  \vdots \\
  x_{j-1} \\
  x_{j+1} \\
  \vdots \\
  x_k
\end{bmatrix}
\]
and sometimes the test result says that everyone involved is healthy. This brings the total to 6D - 3 = 7D - 3 tests. The final number is 7D - 3 and it is \( \leq 7D + 1 \).

**Proposition 38:** Schedule matrices (14), (15) and (16) constitute a \((7D+1)-(7D+1, N, D)\)-tropical protocol.

Much to our surprise, this protocol does not depend on how many people are being screened. Moreover, this protocol tells us the number of patients as a part of the output—we do not have to know and tell the protocol the number \( D \) before the protocol begins.

**C. Upgrade the Third Test**

As it turns out, some tests in the \((7D+1)\)-protocol are redundant. The key is that the confirmation matrix (16) can be combined with the searching matrix (14) of the first \( j-1 \) persons in the following way
\[
[c^\sharp] := [j - 1 \ \cdots \ 1 \ \infty \ \cdots \ \infty] \odot x.
\] (17)
This test behaves like a verification of whether \( j \) is really sick. At the same time it prefetches the result of the searching matrix in the next level of the recursion should the verification fail.

**Lemma 39:** Let \( a \), \( b \), and \( c^\sharp \) be defined with searching matrices (14) and (15) and multitask matrix (17). Let \( j := (a - b + N + 1)/2 \). We have

- \( x_j + j \geq a \) is positive, query \( a \).
- \( x_j + (N + 1 - j) \geq b \) is positive, query \( b \).
- \( x_j + j \geq c^\sharp \) is positive, query \( c^\sharp \).

This pair of statements is by Lemma 37 and the third statement by the definition of \( c^\sharp \). Only the last one is nontrivial so let us prove it. Suppose \( i \) is the index that attains the minimum: \( c^\sharp := \min_{i \leq j} x_j + (j - i) \). Then \( a + b - N - 1 \leq x_i + i + x_i + (N + 1 - j) - N - 1 = 2x_i \leq 2x_i + (j - i) = 2c^\sharp \). This finishes the proof.

These four inequalities enjoy a dichotomous behavior.

**Lemma 40:** Let \( a \) and \( b \) and \( c^\sharp \) be defined with searching matrices (14) and (15) and verify—prefetching matrix (17). Let \( j := (a - b + N + 1)/2 \). Either the following four hold

(i) \( x_j + j = a \),
(ii) \( x_j + (N + 1 - j) = b \),
(iii) \( x_j = (a + b - N - 1)/2 \),
(iv) \( c^\sharp = (a + b - N - 1)/2 \),

or the following four hold

- \( x_j + j > a \) (thus \( a = \min_{i < j} x_j + i \)),
- \( x_j + (N + 1 - j) > b \) (thus \( b = \min_{k > j} x_k + (N + 1 - k) \)),
- \( x_j > (a + b - N - 1)/2 \),
- \( c^\sharp > (a + b - N - 1)/2 \).

**Proof:** Due to Lemma 39, it suffices to prove that (i), (ii), (iii), and (iv) are equivalent to each other. (i) is equivalent to \( x_j = a - j = a - (a - b + N + 1)/2 = (a + b - N - 1)/2 \). (ii) is equivalent to \( x_j = b - (N + 1 - j) = b - (N + 1 - (a - b + N + 1)/2) = (a + b - N - 1)/2 \). Hence (i), (ii), and (iii) are equivalent to each other. If they hold, then due to the inequalities \( x_j = (a + b - N - 1)/2 \leq c^\sharp \leq x_j \) (iv) holds. Conversely, suppose that (iv) holds. Let \( i \leq j \) be the index that attains the minimum: \( c^\sharp := \min_{i \leq j} x_j + (j - i) \). Then the inequalities \( 2c^\sharp = a + b - N - 1 \leq x_i + i + x_i + (N + 1 - i) - N - 1 \leq 2x_i \leq 2x_i + 2(j - i) = 2c^\sharp \) squeeze. Hence
\[2(j - i) = 0 \text{ and } (i), (ii), \text{ and } (iii) \text{ hold. This finishes the proof.}\]

Thanks to the dichotomy related to \(c^4\), we either confirm that \(j\) is indeed infected when we see (iv) holds (plus we know her Ct value \(x_j = (a + b - N - 1)/2\) or we know we can split the population at \(j\) when we see (iv) does not hold. This leads to a strategy that only queries \(3D + 1\) times.

**D. 3D + 1 Adaptive Tests Diagnose D Infected Persons**

In what follows, we use the word *position* as in the *winning positions* in the combinatorial game theory, especially in the theory of the game of Nim [83]. In this context, executing a tropical protocol is like playing a game against Mother Nature. A position is a “current state” when we are halfway toward completely understanding everyone’s Ct values. We win the game by picking out all infected individuals in a limited amount of *moves*. A move is analogous to one single test if we care about the total number of tests, or to a batch of parallel tests if we care about the number of rounds.

Denote by a tuple \((D^{(1)}, D^{(2)}, \ldots, D^{(II)})\) a position where

- there are \(\Pi\) piles of people;
- the \(\pi\)th pile contain \(N^{(\pi)}\) persons whose Ct values are denoted by \(x^{(\pi)}_1, \ldots, x^{(\pi)}_{N^{(\pi)}}\);
- the first pile contains \(D^{(1)} \geq 0\) infected persons;
- for each \(\pi \geq 2\), the \(\pi\)th pile contains \(D^{(\pi)} \geq 1\) infected persons; and
- for each \(\pi \geq 2\), we know the search results
  \[
  a^{(\pi)} := \min_j (x^{(\pi)}_j + j) \quad \text{and} \quad b^{(\pi)} := \min_j (x^{(\pi)}_j + (N^{(\pi)} + 1 - j))
  \]
  of the \(\pi\)th pile.

There are two moves that evolves a position into another position.

If \(\Pi = 1\), we perform searching matrix (14) on the one and only pile. Denote the test result by \(a^{(1)}\). If \(a^{(1)}\) is negative, the protocol terminates and reports everyone healthy. If \(a^{(1)}\) is positive, perform searching matrix (15) and denote the result by \(b^{(1)}\). Of this pile of people we now know the “\(a\)” and “\(b\)” ; we migrate these people to the second pile and leave the first pile empty. That is, we evolve the position \((D^{(1)}, D^{(2)}))\) into the position \((0, D^{(2)})\).

If \(\Pi \geq 2\), we focus on the second pile. Of this pile we already know \(a^{(2)}\) and \(b^{(2)}\) (by the definition). Compute \(j^{(2)} := a^{(2)} + b^{(2)} - N^{(2)} - 1\). We perform the verify-prefetching matrix (17) with \(j := j^{(2)}\) and denote the result by \(c^{(2)}\). By Lemma 39, \(c^{(2)}\) is equal to or greater than \((a^{(2)} + b^{(2)} - N^{(2)} - 1)/2\).

If \(c^{(2)}\) is equal to \((a^{(2)} + b^{(2)} - N^{(2)} - 1)/2\), then \(j^{(2)}\) is infected with Ct value \(c^{(2)}\). The remaining of the second pile, with \(D^{(2)} - 1\) patients remained to be found, is merged with the first pile. Now the position \((D^{(1)}), D^{(2)}, \ldots, D^{(II)}\)) is evolved into \((D^{(1)} + D^{(2)} - 1, D^{(3)}, \ldots, D^{(II)})\).

If \(c^{(2)}\) is greater than \((a^{(2)} + b^{(2)} - N^{(2)} - 1)/2\), then we know
\[
\begin{align*}
a^{(2)} &= \min_{i \leq j^{(2)}} (x^{(2)}_i + i), \\
b^{(2)} &= \min_{k > j^{(2)}} (x^{(2)}_k + (N^{(2)} + 1 - k)), \\
c^{(2)} &= \min_{i < j^{(2)}} (x^{(2)}_i + (j^{(2)} - i)).
\end{align*}
\]

For the first \(j^{(2)}\) persons, we now know their “\(a\)” and “\(b\)” ; for the last \(N^{(2)} - j^{(2)}\) persons, we now know their “\(b\)” . It suffices to query the latter’s “\(a\)” :
\[
d^{(2)} := \min_{k > j^{(2)}} (x^{(2)}_k + (k - j^{(2)})).
\]

Consequently, we know the results of the searching matrices for the first half and the second half. Suppose the first half contains \(D^{(2, \leq k)}\) infected persons and the second half contains \(D^{(2, > k)}\) infected persons. (We know nothing about \(D^{(2, \leq k)}\) and \(D^{(2, > k)}\) beyond that they are positive and sum to \(D^{(2)}\). ) Now the position is evolved from \((D^{(1)}, \ldots, D^{(II)})\) into \((D^{(1)}, D^{(2, \leq k)}, D^{(2, > k)}, D^{(3)}, \ldots, D^{(II)})\).

Knowing how positions evolve, we estimate the cost.

**Theorem 41:** The protocol specified in this subsection is a \((3D + 1)-(3D + 1, N, D)\)-tropical protocol.

**Proof:** Declare a budget function
\[
T(D^{(1)}, D^{(2)}, \ldots, D^{(II)}) := 3 - 2\Pi + 3 \sum_{\pi = 1}^{\Pi} D^{(\pi)}.
\]

It is clear that when \(\Pi = 1\), the \(T\)-formula of a singleton collapses to \(T(D^{(1)}) = 3 - 2 + 3D^{(1)} = 3D^{(1)} + 1\). Hence the theorem will be proved if we can show that each position \((D^{(1)}, \ldots, D^{(II)})\) requires at most \(T\)(that position) tests.

We employ induction on the \(T\)-values, the budgets, of the positions. We will show that whenever one position is evolved into another position, the number of tests spent is at most \(T\) (former position) \(- T\) (latter position). That way, we will never overspend the budge. And as the remaining budge decreases, we will eventually reach our end goal, that all infected individuals are identified.

Base case—\(\Pi = 1\) and \(D^{(1)} = 0\): We spend one test on \(a^{(1)}\), the “\(a\)” of the first and only pile. After getting a negative result we conclude that everyone is healthy. Since
\[
T(0) = 1,
\]
the cost meets the budget for the base case.

Now suppose that any position whose \(T\)-value falls below \(T\) leads to successful identification of all infected individuals before the budget runs out. Suppose that \((D^{(1)}, \ldots, D^{(II)})\) is a position with \(T\)-value \(T\). We want to show that it can be cleared before the budget runs out.

Induction step, case one—\(\Pi = 1\) and \(D^{(1)} \geq 1\): We spend two tests on \(a^{(1)}\) and \(b^{(1)}\), the “\(a\)” and “\(b\)” of the first pile. After that we relabel the first pile as the second pile, evolving \((D^{(1)})\) into \((0, D^{(1)})\).

\[
T(D^{(1)}) = 2 + T(0, D^{(1)}),
\]
the cost meets the budget for induction step, case one.
Induction step, case two—$\Pi \geq 2$ and $c^{(2)} = (a^{(2)} + b^{(2)} - N^{(2)} - 1)/2$: We spend one test on $c^{(2)}$ and confirmed that $j^{(2)}$ is infected with Ct value $c^{(2)}$. By doing so, we evolve $(D^{(1)}, \ldots, D^{(\Pi)})$ into $(D^{(1)} + D^{(2)} - 1, D^{(3)}, \ldots, D^{(\Pi)})$. Since $T(D^{(1)}, \ldots, D^{(\Pi)}) = 1 + T(D^{(1)} + D^{(2)} - 1, D^{(3)}, \ldots, D^{(\Pi)})$, the cost meets the budget for induction step, case two.

Induction step, case three—$\Pi \geq 2$ and $c^{(2)} > (a^{(2)} + b^{(2)} - N^{(2)} - 1)/2$: We spend two tests on $c^{(2)}$ and $d^{(2)}$ to complete our knowledge of the “a” and “b” of the first $j^{(2)}$ persons and the last $N^{(2)} - j^{(2)}$ persons. We evolve $(D^{(1)}, \ldots, D^{(\Pi)})$ into $(D^{(1)} + D^{(2)1}, D^{(2), j}, D^{(2), \geq j}, D^{(3)}, \ldots, D^{(\Pi)})$ by doing so. Since $T(D^{(1)}, \ldots, D^{(\Pi)}) = 2 + T(D^{(1)} + D^{(2)1}, D^{(2), j}, D^{(2), \geq j}, D^{(3)}, \ldots, D^{(\Pi)})$, the cost meets the budget for induction step, case three. This is the last piece of the induction and hence completes the proof.

A generalization of Theorem 41 to a delay-limited situation is the following. We defer the proof until Appendix C.

**Theorem 42 (Deep Searching):** Let $T := 4D\lceil \log_2 N \rceil + 1$. There is a $T$-$(T, N, D)$-tropical protocol within maximum delay $\ell$.

**IX. DISCUSSION**

**A. Open Problems**

For the $D = 2$ case with limited delay (Theorem 23), it is straightforward to handle one patient and two unequally infected patients. For two equally infected patients, we use $Q^{(k)}$ to obtain extra information. Can we simplify $Q^{(k)}$? The goal is to find a weaker notion of BIT that is still an inductive invariant.

For general $D$ in nonadaptive case (Theorem 32), we used exponential $\ell$ but remarked that a polynomial $\ell$ should be possible. Is it? Also, do there exist structural constructions that, more or less, generalize Theorem 23 to general $D$?

For the adaptive strategy with limited delay (Theorem 42), the main term of $T$ is $4D\log_2 N$. From an information-theoretical perspective, the main term should be $D\log_2 N$. Is this achievable? Are there tradeoffs between $T$ the number of total tests and $R$ the number of rounds?

Recall that PCR only runs for 40 cycles in real life. Thus, if there is a Ct 35 specimen delayed by 15 cycles, the expected Ct value is 50 but we only see 40. This is a false negative result. Can we increase the maximum delay to the extent where false negative starts showing up but we still benefit from it? Note that in this case, the decoder must perform more a complicated pattern matching, one that treats 40 as a wildcard that can possibly be 40, 45, 50, or infinity.

**B. Concern of Design**

Hong et al. [84] suggested using factorizations of a complete hypergraph to generate balanced pooling designs. Here, balanced means that every person appears in the same number of tubes and every tubes receives (almost) the same number of persons. However, they argued that those conditions make pooling more consistent. We, while agreeing with their argument, want to add that there are other ways to achieve the same goal. Kirkman systems, Steiner systems, BIBDs, and constant weight codes are candidates that sound equally good. To be more specific, we believe that one should (also) optimize for the probability that a block $B$ is covered by the union of a small number of other blocks $Z_1, \ldots, Z_D$ (cf. superimposed codes, $D$-cover-free families).

**C. Origin of Masking**

It is worth mentioning that Hwang and Xu [85] once published a variant of group testing where there are two infected people, one heavily infected and the other slightly infected. The testing result is quantified by three possibilities: (i) The tested pool contains the heavily infected person. (ii) The tested pool contains the lightly infected person, but not the heavily one. (iii) The tested pool contains neither infected person. Notice that (i), (ii), and (iii) can be thought as Ct values 1, 2, and $\infty$, respectively, while the infected persons have Ct values 1 and 2. Hwang and Xu’s problem formulation already suggested that the presence of the heavily infected complicates the identification of the lightly infected. The group testing scheme proposed in Sections IV and V and Appendix B can be interpreted as a solution to a generalization of their setup.

**APPENDIX A**

**PROJECTIVE SPACE (PROOF OF THEOREM 12)**

In this appendix, we factor in the restriction that one often cannot wait for arbitrarily long delays but can afford more tests. This appendix shows how to trade $T$ for $\ell$ and proves Theorem 12.

**A. Three Tests for Higher Volume**

The following is a three-test seven-person example. Name the persons T, U, V, W, X, Y, Z and let $t, u, v, w, x, y, z$ be

---

7In the block design context, every block having the same number of vertices is called uniform or proper; every vertex appearing in the same number of blocks is called balanced. In the hypergraph context, the former is called uniform; the latter is called regular. Authors of [84] want both properties.
their Ct values. Encoding:

\[
\begin{bmatrix}
0 & 0 & 7 & 7 & 7 & 0 \\
0 & 7 & 0 & 0 & 0 & 7 \\
0 & 7 & 7 & 7 & 0 & 0
\end{bmatrix}
\] ⊗

Decoding: Figure 6 generalizes Figure 3 and illustrate how to map the test results back to who is infected and how infected they are.

For a finite \( \ell \), the columns of a schedule matrix are vectors in \([0,1,\ldots,\ell,\infty]^3\) with at least one 0. The number of such lattice points is \((\ell + 2)^3 - (\ell + 1)^3 = 3\ell^2 + 9\ell + 7\). For \( \ell = 0, 1, \) and 2, the number of lattice points are 7, 19, and 37, which are 2x, 6x, and 12x increases in throughput, respectively.

### B. Asymptote of One Infection

Fix an upper bound on delay \( \ell \geq 0 \). How fast can \( N \) grow if the number of tests \( T \) approaches infinity? Clearly we want to select, for each person, a delay column

\[
\delta = \begin{bmatrix}
\delta_1 \\
\vdots \\
\delta_T
\end{bmatrix} \in \{0,1,\ldots,\ell,\infty\}^{T \times 1}
\]

such that the straight lines

\[
\{ \delta \odot [x] = \begin{bmatrix}
\delta_1 + x \\
\vdots \\
\delta_T + x
\end{bmatrix} \mid x \in \mathbb{R} \} \subseteq (\mathbb{R} \cup \{\infty\})^{T \times 1}
\]

are disjoint (far away) from each other. Note that every line contains one and only one delay column

\[
\delta := \delta \odot [-\min(\delta)] = \begin{bmatrix}
\delta_1 - \min(\delta) \\
\vdots \\
\delta_T - \min(\delta)
\end{bmatrix}
\]

that has at least one zero entry and no negative entries. This means that every line passes one and only one point in the “shell”

\[
\text{III} := \{0,1,\ldots,\ell,\infty\}^{T \times 1} \setminus \{0,1,\ldots,\ell,\infty\}^{T \times 1}. 
\]

III has cardinality \((\ell + 2)^T - (\ell + 1)^T \approx T^2.\) We are ready to prove Theorem 12.

**Proof of Theorem 12**: To see \( N \leq (\ell + 2)^T - (\ell + 1)^T \times 1 \), observe that every column vector in \([0,1,\ldots,\ell,\infty]^{T \times 1}\) is congruent to a column vector in III modulo

\[
\begin{bmatrix}
1 \\
\vdots \\
1
\end{bmatrix}
\]

To obtain a tropical code that meets the bound \( N = (\ell + 2)^T - (\ell + 1)^T \), use III per se or use a collection of column vectors that congruent to different column vectors in III. \( \square \)

### APPENDIX B

#### LARGER BITE (PROOF OF THEOREM 23)

Let us first define the schedule matrices. The proof follows.

Fix a \((t,n,2)\)-tropical code \( S \). Define \( S^{(k)} \) to be this \( kt \times n^k \) matrix

\[
S^{(k)} = \begin{bmatrix}
1_{1 \times n} \otimes S \\
1_{1 \times n} \otimes S \otimes 1_{1 \times n} \\
\vdots \\
1_{1 \times n} \otimes S \otimes \cdots \otimes 1_{1 \times n}
\end{bmatrix}
\]

for all \( k \geq 2 \). Also define \( Q^{(k)} \) to be this \((2k - 3) \times n^k \) matrix

\[
\omega \begin{bmatrix}
\begin{bmatrix}
1 \\
\vdots \\
1
\end{bmatrix} \\
\begin{bmatrix}
\beta_1^{k-1} \\
\vdots \\
\beta_{2k-3}^{k-1}
\end{bmatrix}
\end{bmatrix}^{-1} \begin{bmatrix}
1_{1 \times n} \otimes S \\
1_{1 \times n} \otimes S \otimes 1_{1 \times n} \\
\vdots \\
1_{1 \times n} \otimes S \otimes \cdots \otimes 1_{1 \times n}
\end{bmatrix}
\]

where \( \omega : \mathbb{F}_q \rightarrow \{0,\ldots,q-1\} \) is a look-up bijection that applies to matrices entry-wise,

\[
M = \omega^{-1} \left( \begin{bmatrix}
0 & \ldots & n-1
\end{bmatrix} \right),
\]

\( \alpha_1,\ldots,\alpha_k \) and \( \beta_1,\ldots,\beta_{2k-3} \) are distinct elements in \( \mathbb{F}_q \), and \( q \) is the smallest prime power \( \geq \max(3k - 3,n) \). We now use

\[
\begin{bmatrix}
S^{(k)} \\
Q^{(k)}
\end{bmatrix}
\]

to prove Theorem 23.

**Proof of Theorem 23**: First of all, if everyone is healthy or there is merely one infected person, the situation will be trivial. Hereafter we assume that there are two infected people. Let their indices be \( Y = y_1 + y_2 n + \cdots + y_k n^{k-1} \) and \( Z = z_1 + z_2 n + \cdots + z_k n^{k-1} \) in their \( n \)-ary expansions. Let their Ct values be \( y_* \) and \( z_* \), respectively. Then the first \( t \) tests teach us \( \{y_1, y_*\}, \{(z_1, z_*)\} \) if \( y_1 \neq z_1 \) or \( y_1 = z_1 \), respectively. In general, the \((jt - t + 1)\)th to the \((jt)\)th tests teach us \( \{y_1, y_*\}, \{(z_1, z_*)\} \) depending on whether the digits differ or not.

The next step is to sort \( y_1, z_1,\ldots,y_*, z_* \) into two piles, \( y_1,\ldots,y_k \) and \( z_1,\ldots,z_k \), so that we can recover \( Y \) and \( Z \). This can be done when two patients assume different Ct values, \( y_* \neq z_* \), in which case we know the digits of \( Y \) are those that associate to \( y_* \) and the digits of \( Z \) are those that associate to \( z_* \). On the other hand, if we only see one Ct value the whole time, then \( y_* = z_* \). We will utilize the last \( 2k - 3 \) tests as the checksums of a systematic Reed–Solomon code to help sorting. Here is how.

Without loss of generality, we may assume \( y_* = 0 = z_* \). Then the decoding boils down to the following task: Suppose

\[
Y := \omega^{-1} \left( \begin{bmatrix}
y_1,\ldots,y_k, u_1,\ldots, u_{2k-3}
\end{bmatrix} \right)
\]

and

\[
Z := \omega^{-1} \left( \begin{bmatrix}
z_1,\ldots,z_k, v_1,\ldots, v_{2k-3}
\end{bmatrix} \right)
\]

are two codewords of a \([3k-3, k, 2k-2]\)-Reed–Solomon code. Suppose that we know \( \{y_1, z_1\} \) to \( \{y_k, z_k\} \). Suppose also we know \( \min(u_1, v_1) \) to \( \min(v_{2k-3}, v_{2k-3}) \). To recover \( Y \) and \( Z \), make a guess of two codewords

\[
A := \omega^{-1} \left( \begin{bmatrix}
a_1,\ldots,a_k, e_1,\ldots, e_{2k-3}
\end{bmatrix} \right)
\]

and

\[
B := \omega^{-1} \left( \begin{bmatrix}
b_1,\ldots,b_k, f_1,\ldots, f_{2k-3}
\end{bmatrix} \right)
\]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
such that \( \{a_i, b_i \} = \{y_i, z_i \} \) for \( 1 \leq i \leq k \) and \( \min(e_j, f_j) = \min(u_j, v_j) \) for \( 1 \leq j \leq 2k - 3 \). If remains to show that \( \{A, B\} = \{Y, Z\} \).

Let \( I_A \) be the set of coordinates \( i \) where \( a_i = y_i \). Let \( J_A \) be the set of coordinates \( j \) where \( e_j = \min(e_j, f_j) = \min(u_j, v_j) \). Define \( I_A, J_A, I_B, J_B, I_B \) and \( J_B \) similarly. Then \( |I_A| + |J_A| + |I_B| + |J_B| \geq 2k \) and \( |J_A| + |J_B| \geq 2k - 3 \). Hence at least one of \( |I_A| + |J_A| + |I_B| + |J_B| \geq k \). Say \( |I_B| + |J_B| \geq k \). The Reed-Solomon code being \( \{3k - 3, 3k - 2k - 2\} \) forces \( B = Z \). This then forces \( A = Y \) and we finish proving that schedule matrix (18) is a valid tropical code.

Now assume that \( S \) BITEs. Then \( n > 4 \cdot 2^t \). This implies \( n^k > 4 \cdot 2^{kt+2k-3} \), hence schedule matrix (18) BITEs. \( \square \)

**APPENDIX C**

**DEEP SEARCHING (Proof of Theorem 42)**

Let \( \ell \) be the largest available delay. We want to prove that the number of tests needed is at most \( T = 4D[\log_4 N] + 1 \).

Here is an overview of the strategy.

1. Instead of matrix (6), we can only afford this schedule matrix
   \[
   \begin{bmatrix}
   1 & \cdots & 1 & \cdots & \ell & \cdots & \ell & \cdots & 1 & \cdots & 1
   
   \ell & \cdots & \ell & \cdots & 1 & \cdots & 1 & \cdots & 1 & \cdots & 1
   \end{bmatrix}
   \]
   (19)

   where each unique column repeats \( N/\ell \) times. In other words, we divide \( N \) people into \( \ell \) piles, treat each pile as one person, and apply searching matrices (14) and (15).

   Now suppose that the diff-lay \( a-b \) points to the \( j \)th pile. That is, \( j := (a-b+\ell-1)/2 \). Similar to Lemma 40, there are three possibilities.

   (i) The \( j \)th pile contains a patient and her Ct value is \( a \).

   (ii) The \( j \)th pile is healthy. Instead, the first \( j-1 \) piles contain \( \geq 1 \) patient and the last \( \ell-j \) piles contain \( \geq 1 \) patient.

   (iii) The \( j \)th pile does contain some patients but their Ct values are \( > a \). Meanwhile, the first \( j-1 \) piles contain \( \geq 1 \) patient and the last \( \ell-j \) piles contain \( \geq 1 \) patient. Apply the verify-prefetching matrix (17).

   \[
   \left[ c^j \right] := \left[ j-1 \cdots j-1 \cdots 1 \cdots 1 \cdots 0 \cdots 0 \right] \odot \left[ \begin{array}{c} x_1 \\ \vdots \\ x_j \end{array} \right] \odot \left[ \begin{array}{c} x_j \odot \ell^N \\ x_j \odot \ell^{2N} \\ \vdots \\ x_j \odot \ell^{N(k-1)} \end{array} \right]
   \]

   Then \( c^j \) equals \( (a+b-\ell-1)/2 \), then (i) is the case and we can reduce our scope to the \( j \)th pile, which is one order of magnitude smaller. If \( c^j \) is greater then \( (a+b-\ell-1)/2 \), then (ii) is the case and the remaining population splits into \( 2 \) super-piles of piles

   \[
   \begin{bmatrix}
   x_1 \\ \vdots \\ x_{\ell(\ell-1)/2}
   \end{bmatrix}
   \quad \text{and} \quad \begin{bmatrix}
   x_{\ell(\ell+1)/2+1} \\ \vdots \\ x_N
   \end{bmatrix}
   \]

   each having at least one infected person. Now it suffices to apply the recursive algorithm to them separately.

   Overall, our strategy is a digit-by-digit \( \ell \)-ary searching. In sunny days we can confirm a digit and reduce the candidates to a smaller population. In rainy days we split the population into two halves. With these ideas, we can prove Theorem 42.

**Proof of Theorem 42**: Let there be \( N \) persons to be tested. Partition them into \( \ell \) almost-equal piles, treat them as \( \ell \) persons and apply Theorem 41. If there is no patient, the protocol will end the moment the first test comes out negative. Hereafter, we let \( D \geq 1 \). Let there be \( D^* \) infected piles. Then it takes \( 3D^* + 1 \leq 4D \) test to single out the infected piles. To each and every infected pile, apply Theorem 41 recursively. That will single out some infected sub-piles, followed by some infected sub-sub-piles. And so on and so forth. It takes at most \( 4D \cdot \log_\ell N \) tests to split the population down to atomic individuals. This finishes the proof. \( \square \)

**APPENDIX D**

**PROBABILISTIC MODEL AND SIMULATION**

Alongside the development of the theoretical aspects of tropical group testing, we devote the very last appendix to benchmarking the performance of matching and delaying in a practical setup. By practical, we mean that the patients now follow probabilistic distributions and the viral load of a mixture of specimens are the sum of the individual viral loads.

The simulations to be presented here justify why tropical arithmetic—minimum and addition—are good approximation of the additive nature of the viral load in reality. In fact, as to be shown later, assuming tropical arithmetic not only simplifies the decoder (because all the data are just integers and the only operations are + and >) but also performs equally as good as those that use real numbers to model viral loads.

We begin with introducing our setup and decoding algorithm, followed by the charts and analyses.

**A. The Prior**

Let \( p \in [0, 1] \) be the prevalence rate. Let \( N \) denote the number of people to be tested. For every individual \( j \in [N] \), we toss a Bernoulli coin with mean \( p \). If the outcome of the toss is 0, then individual \( j \) is healthy and her Ct value \( x_j \) is set to be 99; if the outcome of the toss is 1, then individual \( j \) is infected and her Ct value \( x_j \) is drawn continuously uniformly from the interval \([16, 32]\) (see [18], [26], [27], [28], [29] for why this interval). Note that \( x_j \) is not necessarily an integer but a random real number. Under this setup, the viral load \( x_j \) is set to be \( 2^{-x_j} \). Denote by

\[
2^{-x} := \begin{bmatrix}
2^{-x_1} \\
\vdots \\
2^{-x_N}
\end{bmatrix}
\]

the column vector that represents the number of virus particles of the specimens from each of the \( N \) individuals.

**B. The Encoder**

Given a block design \( \mathcal{H} \) and a bijection \( j: \mathcal{H} \to [N] \), we will construct the schedule matrix \( S \) as we have always been:

\[
S_{ij}(B) := \begin{cases}
\delta & \text{if } t \in B, \\
\infty & \text{if } t \notin B.
\end{cases}
\]
hence \( x_j \geq 25 \)

\[
\begin{array}{c}
\delta = 8 \\
c_{t_1} = 19 \\
c_{t_2} = 25 \\
c_{t_3} = 30 \\
\end{array}
\]

Fig. 7. Phase I—underestimate: Suppose that the \( j \)th person participates in tests \( t_1, t_2, t_3 \in [T] \). Then \( x_j \) has lower bounds \( c_{t_1} - S_{t_1,j} = x_j = c_{t_2} - S_{t_2,j} \) and \( c_{t_3} - S_{t_3,j} \). They are 19, 25, and 30 – 8 = 22, respectively. The strongest bound is \( x_j \geq 25 \) so we let \( u_j := 25 \).

\[
\begin{array}{c}
\delta = 8 \\
c_{t_1} = 29 \\
c_{t_2} = 17 \\
c_{t_3} = 21 \\
\end{array}
\]

Fig. 8. Phase II—match: If the \( j \)th person is the main contributor of some two tubes \( t_1, t_2 \in [T] \), we will see \( c_{t_1} - S_{t_1,j} = x_j = c_{t_2} - S_{t_2,j} \). Conversely, whenever we see that the greatest two \( c_t - S_{t,j} \) coincide, we can guess with confidence that \( x_j = u_j \). In this figure, \( c_{t_1} - S_{t_1,j} = 30 - 8 = 21 = c_{t_3} - S_{t_3,j} \), so \( x_j \) is likely 21.

But here, instead of some reminder tricks, we simply use random delays \( \delta = \ell \cdot \text{Bernoulli}(1/2) \in \{0, \ell\} \) that are generated unbiasedly and independently. Let \( 2^{-S} \) be the matrix of 2 raised to the power of each entry of \( -S \):

\[
2^{-S} := \begin{bmatrix} 2^{-S_{11}} & \cdots & 2^{-S_{1N}} \\ \vdots & \ddots & \vdots \\ 2^{-S_{T1}} & \cdots & 2^{-S_{TN}} \end{bmatrix}
\]

We implement delaying by a dilution.

Now perform the PCR tests

\[
v := 2^{-S}2^{-x}
\]

implemented by a matrix-vector multiplication. The \( t \)th row of \( v \), denoted by \( v_t \), is the viral load of the \( t \)th tube given the delay schedule \( S \) and the viral loads \( 2^{-x} \). Let

\[
c := \min(40, [- \log_2(v)]) = \begin{bmatrix} \min(40, [- \log_2(v_{t_1})]) \\ \vdots \\ \min(40, [- \log_2(v_{t_N})]) \end{bmatrix}
\]

be the collection of the capped, integral Ct values of the tubes. Its \( t \)th row is denoted by \( c_t \). The decoder will be given \( S \) and \( c \), from which it shall determine who are infected and how severely their infection are.

C. The Decoder

Our decoder has four phases.

Phase I—underestimate: For any \( t \in [T] \) and \( j \in [N] \), the amount of virus in the \( t \)th tube is at least what was contributed by the \( j \)th person, i.e., \( v_t \geq 2^{-S_{ij} - 2^{-x_j}} \). This yields inequalities \( c_t \leq \log_2(v_t) \leq S_{ij} + x_j \). We therefore define

\[
u_j := \max_t c_t - S_{ij}
\]

to be an underestimate of \( x_j \), see Figure 7 for an example.

Phase II—match: For any person with Ct value \( x_j \) and any tube \( t \), either \( j \) has contributed the majority of the virus and hence \( c_t \approx S_{ij} + x_j \), or someone else contributed significantly more and \( c_t \ll S_{ij} + x_j \). Thinking backward, each \( c_t - S_{ij} \) is either \( x_j \) or less than that. We thus look for person \( j \) where

\[
\{ t \mid c_t - S_{ij} = u_j \} \geq 2.
\]

For every such \( j \), the decoder declares that \( j \) is infected and the inferred Ct value is \( u_j \). See Figure 8 for an example.

Phase III—explain: Motivated by the SCOMP algorithm [86], we want to make sure that all positive tubes are explained by some positive person. For any unexplained tube, we look at its participants and look for the one(s) that could have been the main contributor. In detail, recall that the \( t \)th tube has about \( 2^{-c_t} \) and the \( j \)th person could have contributed at most \( 2^{-S_{ij} - u_j} \). Therefore, the more negative the deficit \( c_t - S_{ij} - u_j \), the less likely \( j \) is responsible for tube \( t \). More concisely, we look for the subset of suspects

\[
\{ j \mid c_t - S_{ij} - u_j = \max_k (c_t - S_{ik} - u_k) \}.
\]

We will report everyone in this subset infected, each with her own \( u_j \) as the speculated Ct value. See Figure 9 for an example.

Phase IV—dark room: Consider the following scenario. A person is infected with Ct value 31. However, against our favor, the tubes she is in have Ct values 11, 12, and 13. By no means we can infer whether she is infected or not. For this type of "patients in the dark room", we set a bar parametrized by \( \rho \), say \( \rho = 14 \). If we see \( u_j < \rho \), then the \( j \)th person could have had \( x_j \in [31, 32] \) and we would not notice due to the masking effect. We declare that she is infected with Ct value \( u_j \). In general, the diagnose of each person is a function in \( \rho \). Set a low \( \rho \) then she is infected; set a high \( \rho \) then she is healthy (unless the previous phases found clear evidence of infection).

\[
\begin{array}{c}
\delta = 8 \\
c_{t_1} = 24 \\
\end{array}
\]

\[
\begin{array}{c}
\delta = 8 \\
c_{t_1} = 28 \\
\end{array}
\]

\[
\begin{array}{c}
\delta = 8 \\
c_{t_1} = 16 \\
\end{array}
\]

\[
\begin{array}{c}
\delta = 8 \\
c_{t_1} = 12 \\
\end{array}
\]

Fig. 9. Phase III—explain: If the \( t \)th tube is positive but not explained by any patients reported by the first two phases we will find in \( t \)'s participants the most likely people and report them infected. In the figure, both \( j_1 \) and \( j_2 \) can contribute \( 2^{-24} \), which is what \( t \) has right now, so both are declared infected.

Fig. 10. Phase IV—dark room: Fix a \( \rho \); say \( \rho = 14 \). If we see \( u_j < \rho \), then the \( j \)th person could have had \( x_j \in [31, 32] \) and we would not notice due to the masking effect. We declare that she is infected with Ct value \( u_j \). In general, the diagnose of each person is a function in \( \rho \). Set a low \( \rho \) then she is infected; set a high \( \rho \) then she is healthy (unless the previous phases found clear evidence of infection).
Fig. 11. Assume uniform Ct values on the interval \([16, 32]\), 15\(\times\)35 Kirkman triple system, and no delay (\(\ell = 0\)). We vary the prevalence rate \(p\) and plot the ROC curves.

Fig. 12. Assume prevalence rate \(p = 10\%\), uniform Ct values, 15\(\times\)35 Kirkman triple system, and no delay (\(\ell = 0\)). We vary the range of the Ct values and plot the ROC curves. Surprisingly, larger interval (consequently larger variance) is easier to decode.

D. The Simulation Result

Figure 11 shows how different prevalence rates translate into performances. As one might expect, the ROC curves become worse as the prevalence rate increases. But it does not “blow-up”, i.e., becomes unusable. Instead, the Kirkman design (inspired by [7]) that assumes batch size as small as \(N = 35\) and code rate \(35/15 = 7/3\) maintains 95% specificity and 97% sensitivity even at 10\% prevalence rate. This makes the Kirkman design a competitive candidate for the high-prevalence regime. Note that we did not apply any delay here; this figure is the pure performance of the tropical model.

Figure 12 shows how different ranges of Ct values affect the performance. A larger range leads to a clear improvement in performance. We infer that this is because more rooms for Ct values lead to fewer “collisions” and make matching easier. It shows that the tropical framework is specialized at handling data that span a large range. In other words, the tropical framework addresses the masking issue not by mitigating it, but by turning into an advantage.

Figure 13 shows how delays facilitate matching. Delaying artificially creates variance in the data, reduces the collision probability even further, and improves the ROC tradeoff. A reasonable delay, \(\ell = 3\) or 4, improves either specificity or sensitivity by 1\%. Put it another way, delaying takes advantage of the “unused” part of the Ct values to extract more information about the patients.

Figure 14 shows how the distribution of the random delay is correlated to the performance. Bernoulli (uniform on \(\{0, \ell\}\)) is apparently better than uniform (uniform on \(\{0, \ldots, \ell\}\)). We believe that this is because the former assumes a greater variance and makes collision rarer. This further strengthens the heuristic that we can extract extra information by making the range of Ct values larger and by making the distribution of Ct values more uniform (hence higher entropy).

Figure 15 shows how the size of the matrix can have impact on the performance. At the same code rate \(N/T = 7/3\), a truncation of a larger Kirkman triple system performs better. We guess that this is because the girth (the minimum cycle length) becomes higher as we appeal to a larger block design and hence the cycle sum (Section VII) is a sum of more random variables, making it less likely to collide. That being said, there appeared to be a ceiling on how much the girth can
It is unclear what happens when there are more than 5 simulations (on a computer) that P-BEST can identify.

We can also achieve 99% specificity.

What's more, our setting is more hazardous than [7]. We assume that tapestry assumes uniform viral load on [1, 105] patients, and 1,050,000 test takers. Compare this to Tapestry’s data point and its standard deviations (4.50% ± 2.41%, 99.30% ± 2.55%) (Table S.XII of the preprint version [88]).

help. Or it could be that there are better designs for higher N to be found.

Figure 16 shows a comparison of tropical group testing to Tapestry [7]. Under the same number of patients (D = 10), the same matrix dimension (45 × 105), and no delay (ℓ = 0), we can also achieve 99% sensitivity and 95% specificity. What’s more, our setting is more hazardous than [7]. We assume uniform Ct value on [16, 32]; Tapestry assumes uniform viral load on [1, 32768]. We assume that cℓi is log2(vi) rounded up to an integer; Tapestry assumes that cℓi = log2(vi) shifted by 0.1Z log2(1.95) ≈ 0.096Z, where Z is a standard Gaussian.

Figure 17 shows a comparison of Steiner systems to the block design used in P-BEST [8]. Per the report, P-BEST is applicable when the prevalence rate is < 1.3%. We found that, under tropical group testing, the same matrix performs rather good til p = 2% > 1.3%. Not only that, there are block designs at the same code rate (N/T = 8) that are as competitive. For instance, at p = 2%, the complete graph on 17 vertices, which is by definition a (2, 2, 17)-Steiner system, has a better sensitivity–specificity tradeoff with fewer vertices and simpler pipetting rules. The (2, 3, 21)-Steiner system (returned by the steiner_triple_system function in SageMath [87]) has a better tradeoff when $p \in [1\%, 2\%]$. The (2, 4, 97)-Steiner system [89, Theorem 2.2] has a better tradeoff across all $p \in [0\%, 2\%]$.

Figure 18 shows a comparison of three designs with code rate $N/T = 30$ at the low-prevalence regime $p = 0.5\%$. The Kirkman triple system on 183 vertices (aka a (2, 3, 183)-Steiner system) outperforms the other two at the cost of reasonable batch size—it applies 183 tests to 5551 persons. The complete 3-uniform hypergraph on 15 vertices, using 15 tests on $\binom{15}{3} = 455$ persons, is more to implement and still achieves 98% sensitivity and 95% specificity. The last one, the complete graph on 61 vertices, uses only two pipets per person and achieves a comparable performance. These comparisons

---

*The author of [8] showed with experiments (using robotic arms) and simulations (on a computer) that P-BEST can identify 5 patients among 384 suspects with high probability, hence the prevalence rate $5/384 \approx 1.3\%$. It is unclear what happens when there are more than 5 patients.
are inspired by [84] and show that, at low prevalence regime, there are many block designs that have the potential to be adopted as a good group testing scheme.

ACKNOWLEDGMENT

The authors would like to thank Gerry Myerson, Venkatesan Guruswami, Ching-Hung Hsieh, and Yi Hsiao for pointing out references. The authors would like to thank Chu-Lan Kao for recommending importance sampling and sharing statistics insights. The authors would like to thank Chih-Yang Hsia for programming aids. The authors would like to thank Facebook user kiwi.qin for discussion about the $D = 2$ case.

REFERENCES

[1] M. Aldridge and D. Ellis, Pooled Testing and Its Applications in the COVID-19 Pandemic. Cham, Switzerland: Springer, 2022, pp. 217–249, doi: 10.1007/978-3-030-78334-1_11.

[2] B. Fung et al., “Direct comparison of SARS-CoV-2 analytical limits of detection across seven molecular assays,” J. Clin. Microbiol., vol. 58, no. 9, pp. 20791–20801, Aug. 2020.

[3] A. Cohen, N. Shlezinger, A. Solomon, Y. C. Eldar, and M. Medard, “Multi-level group testing with combinatorial decoding and compressed sensing,” 2020, pp. 605–608, Sep. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0163445320304354.

[4] T. S. Perry, “Researchers are using algorithms to tackle the coronavirus test shortage: The scramble to develop new test kits that deliver faster results—[spectral lines],” IEEE Spectr., vol. 57, no. 6, p. 4, Jun. 2020.

[5] D. Donoho, M. Lotfi, and B. Ozturkler. (2020). The Mathematics of Mass Testing for COVID-19. [Online]. Available: https://sinews.siam.org/Details-Page/the-mathematics-of-mass-testing-for-covid-19.

[6] D. Austin. (Oct. 2020). Pooling Strategies for COVID-19 Testing. [Online]. Available: http://www.ams.org/publicoutreach/feature-column/cfc-2020-10.

[7] S. Ghosh et al., “A compressed sensing approach to pooled RT-PCR testing for COVID-19 detection,” IEEE Open J. Signal Process., vol. 2, pp. 248–264, 2021.

[8] R. Gabrys et al., “AC-DC: Amplification curve diagnostics for SARS-CoV-2 viral load in saliva samples in symptomatic and asymptomatic cases,” medRxiv, vol. 2021, pp. 1–7, Feb. 2021. [Online]. Available: https://www.medrxiv.org/content/early/2021/02/16/2021.02.12.21251229.

[9] F. Wang et al., “Group testing large populations for SARS-CoV-2,” medRxiv, vol. 2021, pp. 1–33, Jun. 2021. [Online]. Available: https://www.medrxiv.org/content/early/2021/06/05/2021.06.03.21258258.

[10] A. Heidarzadeh and K. Narayanan, “Two-stage adaptive pooling with noise-robust SARS-CoV-2 testing,” 2020, arXiv:2007.09171.

[11] Y. Lin, C. Yu, T. Liu, C. Chang, and W. Chen, “Constructions and comparisons of polling matrices for pooled testing of COVID-19,” IEEE Trans. Netw. Sci. Eng., vol. 9, no. 2, pp. 467–480, Mar. 2022.

[12] https://orcid.org/0000-0001-8091-9946

[13] https://scholar.google.com/citations?user=Qmp5CMAAAAJ.
