UNIVERSITY OF LEEDS

This is a repository copy of *Small Error Algorithms for Tropical Group Testing*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/218504/

Version: Accepted Version

Article:
Paligadu, V., Johnson, O. and Aldridge, M. orcid.org/0000-0002-9347-1586 (2024) Small Error Algorithms for Tropical Group Testing. IEEE Transactions on Information Theory, 70 (10). pp. 7232-7250. ISSN 0018-9448

https://doi.org/10.1109/tit.2024.3445271

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose
university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Small error algorithms for tropical group testing

Vivekanand Paligadu[*] Oliver Johnson[†] Matthew Aldridge[‡]

1 October 2024

Abstract

We consider a version of the classical group testing problem motivated by PCR testing for COVID-19. In the so-called tropical group testing model, the outcome of a test is the lowest cycle threshold (Ct) level of the individuals pooled within it, rather than a simple binary indicator variable. We introduce the tropical counterparts of three classical non-adaptive algorithms (COMP, DD and SCOMP), and analyse their behaviour through both simulations and bounds on error probabilities. By comparing the results of the tropical and classical algorithms, we gain insight into the extra information provided by learning the outcomes (Ct levels) of the tests. We show that in a limiting regime the tropical COMP algorithm requires as many tests as its classical counterpart, but that for sufficiently dense problems tropical DD can recover more information with fewer tests, and can be viewed as essentially optimal in certain regimes.

1 Introduction

1.1 Group testing and tropical group testing

Group testing is the problem of reliably recovering a subset $\mathcal{K}$ of 'defective' items from a population of N items, using a relatively small number T of so-called pooled tests to test multiple items at the same time. In the classical noiseless setting, the outcome of such tests is binary, indicating whether or not there is at least one defective item in the pool.

[*] School of Mathematics, University of Bristol, Bristol, UK.

[†] Corresponding author: School of Mathematics, University of Bristol, Fry Building, Woodland Road, Bristol, BS8 1UG, UK. Email: O.Johnson@bristol.ac.uk

[‡] School of Mathematics, University of Leeds, Leeds, UK

Group testing was initially introduced by Dorfman [11] in the context of testing for syphilis. It has now developed into a combinatorial and algorithmic problem with a rich history [12, 4], where one seeks to understand the trade-offs between N, $|\mathcal{K}|$ and T, and to understand how $\mathcal{K}$ can be efficiently recovered using computationally feasible algorithms.

Group testing has applications to many fields such as biology, manufacturing, communications and information technology, as described in [4, Section 1.7]. This framework also recently gained considerable attention during the COVID-19 pandemic – see [3] for a review of its use in this context, and [22] for an early proof of concept that pooled testing can detect the presence of COVID-positive individuals. In general, the efficiency of group testing makes it useful when using PCR machines to test large populations for rare conditions.

In this paper, however, we consider not the standard binary group testing model, but rather the 'tropical group testing' model of Wang *et al.* [20]. While the binary model considers only 'positive or negative' outcomes, the tropical group testing model considers different levels of strength of infection (or 'seriousness of defectiveness'). In the tropical model, the output is given by the strongest infected (or 'most defective') input.

We have three reasons for wanting to study further the tropical model of Wang *et al.* [20]. First, the tropical model is a fascinating mathematical puzzle in its own right, where the generalisation of the binary group testing model creates interesting problems requiring creative solutions. We are pleased to be able to build on the work of [20] to propose and rigorously analyse new algorithms for the tropical model.

Second, tropical group testing is an excellent model for situations where the performance of a set of items is determined by the most extreme performance of a single item in the set. Consider, for example, the capacity of a route through a network, which is typically limited by the lowest bandwidth link on that route. If u_i is the bandwidth of link i, and $x_{ti} = 1$ denotes that link i is on route t, then the capacity of route t would be given by $y_t = \min\{u_i : x_{ti} = 1\}$. This is precisely the tropical group testing model defined below in Definition 2.1.

Third, tropical group testing models how information regarding strengths of infections is combined in PCR testing, for example when testing for COVID. On one side, traditional group testing only considers binary positive-or-negative outcomes, so it does not represent how samples with different strengths of infection combine. On the other side, more complicated models (like that we describe in Subsection 1.2 below) do not seem amenable to theoretical analysis, only to exploration via simulation. In Subsection 1.2 we explain how tropical group testing can be formally seen as an approximation of a more detailed, but less tractable, model of PCR testing. In this sense, we feel that the tropical model of [20] hits the 'sweet spot' between keeping the analytical tractability of binary group testing, while still adopting a model that reflects (even if it does not perfectly capture) how PCR testing works.

Other COVID-inspired pooled testing schemes that are designed to take account of quantitative information through numerical values of test outcomes include Tapestry [15, 14], the work of Bharadwaja and Murthy [6], and the two-stage adaptive scheme of Heidarzadeh and Narayanan [16].

1.2 The tropical model as an approximation of PCR testing

To motivate the tropical model, we will consider a detailed model the PCR test, and show how tropical group testing can be viewed as an approximation of this model. Readers already convinced by the utility of the tropical model who want to read about our new contributions can skip to Subsection 1.3.

In the PCR test (see, for example, [21]), the viral genetic material is extracted from a sample and amplified in 'cycles' of a process using polymerase enzymes. In each cycle, the quantity of this material is multiplied by some factor (usually doubled, or slightly less). The presence of the viral genetic material is detected by fluorescence, indicating a positive result if the quantity present exceeds a certain amount. The *cycle threshold* (Ct) value is the number of cycles after which fluorescence is observed – this represents the number of amplifications (or doublings) required to achieve detection. Hence when using PCR to test for COVID [19], a lower Ct value indicates a higher concentration of viral genetic material in the sample.

The following mathematical description of the PCR protocol follows closely that by Bharadwaja and Murthy [6]. Suppose a single sample has initial viral concentration z. In each cycle it will be amplified by a factor of $1 + q$, say. (Bharadwaja and Murthy [6] suggest values of $q \in [1/2, 1]$ are typical.) So the Ct number, the number of cycles u required for the amplified viral concentration to cross a given threshold τ, will satisfy $(1+q)^u z = \tau$ (see [6, eq. (1)]). Equivalently, $z = \tau(1 + q)^{-u}$.

This extends to a model where multiple items with different viral concentrations $z_i = \tau(1 + q)^{-u_i}$ are tested according to a binary pooling matrix $\boldsymbol{x} = (x_{ti})$, where $x_{ti} = 1$ means item i is in pool t. (See Section 2 for a formal description of the test matrix.) The total initial viral concentration for pool t will be

$$\sum_{i:x_{ti}=1} z_i = \tau \sum_{i:x_{ti}=1} (1 + q)^{-u_i}.$$

This will cross the threshold τ after y_t cycles, where

$$(1+q)^{y_t} \tau \sum_{i:x_{ti}=1} (1+q)^{-u_i} = \tau.$$

Rearranged to make y_t the subject, we have

$$y_t = -\log_{1+q} \left(\sum_{i:x_{ti}=1} (1+q)^{-u_i} \right). \tag{1}$$

This is the model of [6, eq. (4)], except for that also including an additive noise term $\epsilon_t \sim N(0, \sigma^2)$.

It is possible to directly study the model (1) empirically. For example, Bharadwaja and Murthy [6] describe several algorithms based on gradient descent, for both known and unknown q, and validate them empirically for prevalence levels found during the Covid-19 pandemic.

However, an alternative perspective comes through the way that Ct values are explicitly combined together in the so-called *tropical group testing* model of Wang *et al.* [20], which takes the outcome of each test to be equal to the minimum value of u_i among items appearing in it (see Definition 2.1 below). This construction can be motivated using the 'tropicality approximation' to (1):

$$y_t = -\log_{1+q} \left(\sum_{i:x_{ti}=1} (1+q)^{-u_i} \right) \approx \min_{i:x_{ti}=1} u_i. \tag{2}$$

This approximation appears to render the model more amenable to mathematical analysis. It seems to be considerably harder to formally prove performance guarantees in the setting of (1) and [6] than the tropical model of [20] and this paper.

The tropicality approximation (2) is exact when a pool contains 0 or 1 infected items (those with $u_i < \infty$), and is very accurate when the viral concentrations of 2 or more infected items differ by a reasonable amount, but can be slightly inaccurate if 2 or more items have very similar viral concentrations. We believe that the tropical model strikes a good balance between maintaining the analytical tractability of standard binary group testing while still modelling the effect of differing viral concentrations as in (1). In that sense, the contribution of our paper is to prove rigorous bounds in a model which, when it comes to modelling PCR tests, is more realistic than the standard group testing model studied in the vast majority of the group testing literature (see [4] and elsewhere).

Since the outcome of a tropical group test is the minimum Ct value of the infected individuals contained within it, strongly infected items with low Ct values tend to dominate the test outcomes, potentially concealing weakly infected individuals with high Ct values. In an attempt to address this limitation, Wang *et al.* introduce the concept of a 'tropical code' involving 'delays' (controlled by a 'delay matrix'), which correspond to adding some samples partway through a PCR run, so they get amplified fewer times. With this approach, [20] describes adaptive and non-adaptive constructions.

1.3 Our contributions

The key contribution of this paper is the development and analysis of non-adaptive algorithms in the tropical setting to recover the Ct values of defective items under a small-error criterion, and to demonstrate gains in performance relative to the classical group testing setting. These algorithms are tropical generalisations of the classical COMP, DD and SCOMP algorithms [2, 4].

A major strength of our algorithms and results is that they do not require the use of a delay matrix, meaning that all samples are added to the machine simultaneously at the start of the process. This means the tests can be run in parallel simultaneously without needing to pause the machine to load further samples, making the resulting schemes easy to implement in practice on a PCR machine. (Strictly speaking, one could say adding a sample at the beginning of the process is 'delay 0' and not adding a sample at all is 'delay ∞'; so one could say our methods involve no *nontrivial* delays.)

In particular, we identify a sharp threshold for the performance of the tropical DD algorithm (see Section 3.4) in certain parameter regimes. In Theorem 6.1 we give an achievability result by showing that in a limiting regime where the number of tests

$$T > \max\{T_\infty, T_d, T_{d-1}, \ldots, T_1\} \tag{3}$$

then the error probability of this algorithm tends to zero. Here the T_r are explicit expressions in terms of the total number of items with particular defectivity levels. Roughly speaking, T_∞ tests are required to find most (but not necessarily all) of the non-defective items, while T_r tests are required to find all the defective items with Ct value r. Further in Remark 6.2, we argue that in a certain 'uniform' case, this result represents an explicit (albeit second-order) improvement over the performance of classical DD.

In contrast, in Theorems 7.1 and 7.4 we show that in the regime where

$$T < \max\{T_d, T_{d-1}, \ldots, T_1\}$$

then the error probability of tropical DD and even of an optimal algorithm tends to 1. Since apart from the absence of T_∞ this is identical to the expression (3), we can conclude that our tropical DD algorithm is asymptotically optimal in parameter regimes where T_∞ does not give the maximum in (3).

The structure of the rest of the paper is as follows. In Section 2 we introduce the notation used in the paper, and formalise the tropical group testing model. In Section 3 we describe the three tropical algorithms we will study, and briefly mention some of their basic properties. Section 4 gives simulation results indicating the performance of these algorithms. We analyse the theoretical performance of the tropical COMP algorithm in Section 5 and of the tropical DD algorithm in Sections 6 and 7.

2 Notation and tropical model

We adapt the classical group testing notation and algorithms of [2, 4] to the tropical group testing model of [20]. The tropical model replaces the 'OR' operation of standard group testing with a 'min' operation, in a way motivated by the use of PCR testing for COVID-19.

In more detail, for a fixed positive integer value d we define the set $\mathcal{D} = \{1, 2, \ldots, d, \infty\}$ of possible defectivity (or infection) levels. Here level ∞ represents the state of being not defective, and levels $1, 2, \ldots, d$ represent different levels of defectivity. As with Ct values in PCR testing, the lower the numerical value of the defectivity level, the stronger the infection; and the higher the numerical value of the defectivity level, the weaker the infection. The exact values represented in the set do not matter from a mathematical point of view – while it may be closer to medical practice to use Ct values such as $\{20, 21, \ldots, 40, \infty\}$, the choice $\{1, 2, \ldots, d, \infty\}$ provides notational convenience.

Given N items, we represent the defectivity level $U_i \in \mathcal{D}$ of each item i as a vector $\boldsymbol{U} = (U_1, \ldots, U_N) \in \mathcal{D}^N$. We write $\mathcal{K}_r = \{j : U_j = r\}$ for the set of items at each level $r \in \mathcal{D}$, and write $\mathcal{K} = \bigcup_{r=1}^d \mathcal{K}_r$ for the total set of defective items, with finite U_i. We write $K_r = |\mathcal{K}_r|$ for the size of each set, $K = \sum_{r=1}^d K_r = |\mathcal{K}|$ for the total number of defective items, and adopt the notation $\boldsymbol{K} = (K_1, \ldots, K_d)$. For $1 \leq r \leq d$ and $1 \leq s \leq K_r$, we will write $i(r, s)$ for the sth item in set $\mathcal{K}_r$ (labelled arbitrarily within $\mathcal{K}_r$).

We assume a combinatorial model: that is, we fix set sizes $K_1, \ldots, K_d$ in advance and assume that the sets $\mathcal{K}_r$ are disjoint and chosen uniformly at random among sets which satisfy these constraints. We will sometimes consider a limiting sequence of problems where $N \to \infty$ with $K \simeq N^\alpha$ for some fixed $\alpha \in (0, 1)$ and $K_i \simeq \theta_i K$ for some θ_i with $\sum_{i=1}^d \theta_i = 1$.

We use a non-adaptive testing strategy, where we fix the whole test design in advance. We represent the test design in a binary $T \times N$ test matrix $\boldsymbol{x}$, with the standard convention that $x_{ti} = 1$ means that item i appears in test t and $x_{ti} = 0$

means that item i does not appear in test t. Our use of non-adaptive strategies in this context is motivated by the fact that PCR tests can be performed in parallel using plates with large numbers of wells (such as 96 or 384) – see for example [13] – meaning that the test strategy needs to be designed in advance.

We now describe the outcome of a so-called tropical group test.

Definition 2.1. *Tropical group testing* is defined by the outcome Y_t of test t being given by the lowest defectivity level U_i among items i that appear in the test:

$$Y_t = \min_i \{U_i : x_{ti} = 1\}. \tag{4}$$

For $d = 1$, there are only two defectivity levels possible for an item i, namely $U_i = 1$ (defective) and $U_i = \infty$ (non-defective). In this case, Definition 2.1 reduces to Dorfman's standard binary group testing model [11], with the outcome of a negative test t denoted by $Y_t = \infty$ (rather than the usual $Y_t = 0$). We refer to this as 'classical group testing'.

For any value of d, if a test contains no defective items (that is, if $U_i = \infty$ for all items i in the test) then the outcome is $Y_t = \infty$, which we regard as a negative test, just as in classical group testing. However, unlike classical group testing, we also receive information about the defectivity levels of the items through the outcomes of positive tests being a number from 1 to d.

In order to analyse tropical group testing, we make some definitions that will be useful, and which extend the definitions and terminology of [2].

Definition 2.2. Write μ_i for the highest outcome of any test that item i appears in:

$$\mu_i \coloneqq \max_t \{Y_t : x_{ti} = 1\}. \tag{5}$$

If item i is not tested, so that $\{Y_t : x_{ti} = 1\} = \emptyset$, we use the convention $\mu_i \coloneqq 1$.

A key deduction is that μ_i is the lowest possible defectivity level for item i.

Lemma 2.3. *For each item i, we have $U_i \geq \mu_i$.*

In particular, if $\mu_i = \infty$ (that is, if the item appears in a negative test) then we can recover with certainty that $U_i = \infty$.

Proof. If an item i is not tested at all, then by Definition 2.2 we know that $\mu_i = 1$, and so the result trivially holds.

Otherwise, if an item i is tested, then for each t such that $x_{ti} = 1$, by Definition 2.1, we know that $U_i \geq Y_t$. So $U_i \geq \max_t\{Y_t : x_{ti} = 1\} = \mu_i$. □

Definition 2.4. We define the following:

1. For each $1 \leq r \leq d$, we refer to an item i that has $\mu_i = r$ as PD(r) ('Possibly Defective at levels $\{r, \ldots, d, \infty\}$').

$$\begin{array}{c} \begin{array}{cccccccc} K_1 & K_2 & \cdots & K_d & \cdots & H_{d-1} & H_d & H_\infty \end{array} \\ \left(\begin{array}{cccc|cccc} +1 & ? & \cdots & ? & \cdots & ? & ? & ? \\ 0 & +1 & \cdots & ? & \cdots & ? & ? & ? \\ & & & & \vdots & & & \\ 0 & 0 & \cdots & ? & \cdots & +1 & ? & ? \\ 0 & 0 & \cdots & +1 & \cdots & 0 & +1 & ? \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 0 & +1 \end{array}\right) \end{array} \Longrightarrow \begin{array}{c} \boldsymbol{Y} \\ \left(\begin{array}{c} 1 \\ 2 \\ \vdots \\ d-1 \\ d \\ \infty \end{array}\right) \end{array}.$$

Figure 1: Schematic illustration of test matrix and outcomes sorted into block form. Here a 0 represents a submatrix of all zeroes, a +1 represents a submatrix which has at least one entry equal to 1 in each column, and ? represents a submatrix which could be of any form. The defective items are sorted by level to the left of the vertical line. The column labels above the matrix represents the number of elements of each type; the vector represents the outcomes of the test.

2. For $r \in \mathcal{D}$, we say that an item i of defectivity level $U_i = r$ is *intruding* if it never appears in a test of outcome r (in which case strict inequality $U_i > \mu_i$ holds in Lemma 2.3).

3. For $r \in \mathcal{D}$, write H_r for the number of tested non-defective items in $\mathrm{PD}(r)$ (those that have $\mu_i = r$). For convenience, also define H_0 to be the number of untested non-defective items.

If $d = 1$, then, in the notation of [2], the number G of intruding non-defectives (i.e. non-defectives that don't appear in any negative tests) corresponds here to those items i with $\mu_i = 1$, tested or untested; so G in [2] corresponds to $H_0 + H_1$ here.

To aid understanding, it can be helpful to sort the rows and columns of the test matrix as illustrated in Figure 1.

The algorithms we describe in Section 3 will be effective for a variety of matrix designs. However, as in [2], in the theoretical analysis in Sections 5–7 we assume that the matrix $\boldsymbol{x}$ is sampled according to a Bernoulli design with parameter p; that is, that the elements x_{ti} are equal to 1 independently of one another with a fixed probability p. As in [4, Section 2.1], we consider a probability $p = \nu/K$ for some fixed ν.

In fact, as justified in [2] and in Section 4 it is often reasonable to take $\nu = 1$. It remains possible that some other choice may be better in some situations, although simulation evidence in Figure 3 shows that the performance of our algorithms is relatively robust to choices of ν close to 1. This means that while in theory we need to know the number of defective items in the sample to design the matrix, for practical purposes it is enough to have a good estimate of this number.

The paper [17] proves that performance is improved in the classical case when using matrix designs with near-constant column weights $L = \lfloor \nu T/K \rfloor$, and simulation evidence in Figure 6 suggests that the same might well be true in the tropical case. However the analysis involved in [17] is significantly more complicated than that in [2], so here we restrict ourselves to the Bernoulli case for the sake of simplicity of exposition, and leave alternate matrix designs for future work.

3 Description of tropical algorithms

3.1 General remarks concerning algorithms

In this section, we describe three algorithms which estimate the true vector of defectivity levels $\boldsymbol{U}$, given the test design matrix $\boldsymbol{x}$ and the vector of test outcomes $\boldsymbol{Y}$. These are the tropical COMP, tropical DD and tropical SCOMP algorithms, adapted from the classical algorithms of the same names in [7, 2] (see also [4, Chapter 2] for a more detailed description).

We first define what is meant by an algorithm in this setting.

Definition 3.1. A decoding (or detection) algorithm is a function $\widehat{\boldsymbol{U}} : \{0,1\}^{T\times N} \times \mathcal{D}^T \to \mathcal{D}^N$ which estimates the defectivity level of each of the items, based only on knowledge of the test design $\boldsymbol{x}$ and outcomes $\boldsymbol{Y}$.

We write $\mathbb{P}(\text{err})$ for the error probability of an algorithm, and $\mathbb{P}(\text{suc}) = 1 - \mathbb{P}(\text{err})$ for the success probability. We define

$$\mathbb{P}(\text{err}) = \mathbb{P}(\widehat{\boldsymbol{U}} \neq \boldsymbol{U}) \tag{6}$$

to be the probability that the algorithm fails to recover all the defectivity levels exactly, where the randomness comes through the design of the matrix and the value of $\boldsymbol{U}$ itself. Sometimes for emphasis we will include the name of the algorithm and the number of tests, for example by writing $\mathbb{P}(\text{err}; \text{DD}, T)$.

Recovering $\boldsymbol{U}$ exactly represents a strong success criterion for this problem. For example, in some cases, we might be happy to simply recover the defective set $\mathcal{K} = \{i : U_i < \infty\}$. We later show that recovering $\boldsymbol{U}$ and recovering $\mathcal{K}$ represent equivalent success criteria for tropical DD and tropical SCOMP, but not for tropical COMP. From a clinical point of view, since lower Ct levels are generally associated with higher infectiousness [19], it might be sufficient to recover all the items with defectivity level below a certain threshold t, that is to find $\bigcup_{r<t} \mathcal{K}_r = \{i : U_i < t\}$.

In this setting, we say that a *false positive error* is an error of underestimating the defectivity level U_i of an item i – that is, of setting $\widehat{U}_i < U_i$ – and a *false negative error* is an error of overestimating the defectivity level U_i of an item i – that is, of setting $\widehat{U}_i > U_i$.

In the remainder of this section, we define the tropical COMP (Subsection 3.3), tropical DD (Subsection 3.4) and tropical SCOMP (Subsection 3.5) algorithms as tropical equivalents of their established classical counterparts. All of these algorithms are relatively simple: they do not require exhaustive search over possible values of $\boldsymbol{U}$ (in contrast to the classical SSS algorithm [2], for example), can be implemented with a small number of passes through the data, and require an amount of storage which is proportional to the number of items and tests.

Despite this simplicity, in the classical case, the DD algorithm has performance close to optimal for certain parameter ranges. This can be seen by comparing [10, Eq. (1.1), (1.2)], which show that DD under a constant column weight design achieves an asymptotic performance which matches that achievable by any algorithm and any test design in the case where $K \simeq N^\alpha$ and $1/2 \leq \alpha < 1$.

Also, note that while simulations show that classical SCOMP outperforms classical DD for a range of finite size problems, Coja-Oghlan *et al.* [9] prove that it requires the same number of tests in an asymptotic sense, with SCOMP having the same rate (in the sense of [2]) as classical DD.

3.2 Counting bounds

For classical group testing, a lower bound on the number of tests required is given by the so-called 'magic number' $T^*_{\text{class}} := \log_2 \binom{N}{K}$, which can be justified on information-theoretic grounds. In fact below this number of tests there is exponential decay in performance of any algorithm, adaptive or non-adaptive, and for any test strategy. Specifically, [5, Theorem 3.1] shows that if T tests are used then in any scenario the success probability for classical group testing satisfies

$$\mathbb{P}(\text{suc}) \leq 2^{-(T^*_{\text{class}} - T)} = \frac{2^T}{\binom{N}{K}}, \tag{7}$$

sometimes referred to as the counting bound.

It may not be *a priori* obvious how the difficulty of the tropical decoding problem with success criterion (6) compares with the corresponding classical problem. In the tropical setting, we receive more information from each test through the more diverse test outcomes, which suggests the problem could be easier; but we also need to recover more information (to find the levels $\boldsymbol{U}$), which suggests the problem could be harder. Nonetheless, if for given parameters any tropical algorithm can demonstrate performance exceeding the classical counting bound (7) then we can be sure that the corresponding tropical problem is easier than its classical counterpart.

By closely mimicking the proof of the classical counting bound (7) given in [5] we can prove its tropical counterpart.

Theorem 3.2. *Write* $T^*_{\mathrm{trop}} := \log_{d+1} \binom{N}{\boldsymbol{K}}$*, where*

$$\binom{N}{\boldsymbol{K}} = \binom{N}{K_1, K_2, \ldots, K_d, N-K} = \frac{N!}{K_1!K_2! \cdots K_d!(N-K)!}$$

is the multinomial coefficient. Then

$$\mathbb{P}(\mathrm{suc}) \leq (d+1)^{-(T^*_{\mathrm{trop}} - T)} = \frac{(d+1)^T}{\binom{N}{\boldsymbol{K}}}. \tag{8}$$

Proof. See Appendix A. □

Writing $\binom{K}{\boldsymbol{K}} = K!/(K_1!K_2! \ldots K_d!)$ and $H(\boldsymbol{\theta}) = -\sum_{i=1}^d \theta_i \log_2(\theta_i)$, we expand

$$T^*_{\mathrm{trop}} = \log_{d+1} \binom{N}{K} + \log_{d+1} \binom{K}{\boldsymbol{K}} \simeq \frac{T^*_{\mathrm{class}}}{\log_2(d+1)} + K \frac{H(\boldsymbol{\theta})}{\log_2(d+1)}. \tag{9}$$

Compared with the classical case, the scaling factor $1/\log_2(d+1) < 1$ on the first term of (9) represents the fact that we potentially gain more information through each test, while the second additive term represents the extra information we are required to recover.

3.3 Tropical COMP

We now describe the tropical COMP algorithm, which extends the classical COMP algorithm described in [7] (see also [8]) – although the idea of the algorithm dates back at least to the work of Kautz and Singleton [18].

We first describe the classical COMP algorithm, which simply declares any item that appears in a negative test as non-defective. All other items are declared defective. In the notation of this paper, classical COMP can be described in the following way. For each item i with $\mu_i = \infty$, we set $\widehat{U}_i = \infty$; otherwise, $\mu_i = 1$ and we set $\widehat{U}_i = 1$. In other words, we set $\widehat{U}_i = \mu_i$ for each item i.

The same rule $\widehat{U}_i = \mu_i$ can also be used in tropical group testing. This is what we call the tropical COMP algorithm.

Algorithm 1: Tropical COMP algorithm

```
**Input:** Test design matrix $\boldsymbol{x}$ and vector of test outcomes $\boldsymbol{Y}$
**Output:** Estimated vector of defectivity levels $\widehat{\boldsymbol{U}}$
**for** each item $i$ **do** set $\widehat{U}_i = \mu_i$;
```

While both classical and tropical COMP mark items appearing in negative tests as non-defective, the tropical COMP algorithm further classifies items into estimated defectivity levels. Note that the two algorithms operate identically when $d = 1$, and have some analogous properties in general. To aid terminology, we first define the notion of unexplained tests in this setting.

Definition 3.3. Fix a test matrix $\boldsymbol{x}$ and an estimate $\widehat{\boldsymbol{U}}$ of $\boldsymbol{U}$. Write

$$\widehat{Y}_t = \min_i \{\widehat{U}_i : x_{ti} = 1\}$$

to be the outcome of test t using matrix $\boldsymbol{x}$ if the true defectivity vector were equal to $\widehat{\boldsymbol{U}}$. We say that test t is *unexplained* by $\widehat{\boldsymbol{U}}$ if $\widehat{Y}_t \neq Y_t$, where Y_t is the actual test outcome, or *explained* if $\widehat{Y}_t = Y_t$.

We call an estimate vector $\widehat{\boldsymbol{U}}$ a *satisfying vector* if it explains all T tests.

The terminology 'satisfying vector' here is the tropical group testing equivalent of the classical group testing notion of a satisfying set [2, 4]. For classical COMP, the estimate given is a satisfying set [4, Lemma 2.3]) – indeed, the largest satisfying set. We have a similar result for tropical COMP.

Lemma 3.4. *The estimate $\widehat{\boldsymbol{U}}^{\mathrm{COMP}}$ given by tropical COMP is a satisfying vector.*

Further, $\widehat{\boldsymbol{U}}^{\mathrm{COMP}}$ is the least satisfying vector, in that if $\boldsymbol{V} \in \mathcal{D}^N$ is also a satisfying vector, then $U_i^{COMP} \leq V_i$ for all items i.

Proof. For the first part, take an arbitrary test t with outcome Y_t. All items i included in this test have $U_i \geq \mu_i \geq Y_t$. Further, there must be an item j with $U_j = Y_t$, otherwise the test outcome would be greater than Y_t. For that item, $\mu_j = Y_t$. Hence,

$$\widehat{Y}_t = \min_i \{\mu_i : x_{ti} = 1\} = \mu_j = U_j = \min_i \{U_i : x_{ti} = 1\} = Y_t,$$

and the test is explained. Since the test t was arbitrary, we have $\widehat{\boldsymbol{Y}} = \boldsymbol{Y}$, and hence $\widehat{\boldsymbol{U}}^{\mathrm{COMP}}$ explains all the tests.

For the second part, note that any satisfying vector $\boldsymbol{V}$ must have $V_i \geq \mu_i = U_i^{\mathrm{COMP}}$ for all i. To see this, consider a vector $\boldsymbol{V}$ and item j with $V_j < \mu_j$. Then let t be a test containing item j for which $Y_t = \mu_j$. There must be at least one such test, by the definition of μ_j, unless j is never tested. If j is never tested, then by assumption $V_j \in \mathcal{D}$ has $V_j \geq 1 = \mu_j$. For this test t,

$$\min_i \{V_i : x_{ti} = 1\} \geq \mu_j > V_j,$$

so $\boldsymbol{V}$ is not satisfying. □

We know that classical COMP never makes false negative errors. The same is true for tropical COMP – recall that we use this terminology to refer to an error of the form $\widehat{U}_i > U_i$.

Lemma 3.5. *Tropical COMP never makes false negative errors.*

Proof. This follows directly from Lemma 2.3, which tells us that $U_i \geq \mu_i$, where μ_i is the tropical COMP estimate. □

For tropical COMP, the success criterion given by (6) to recover the whole vector $\boldsymbol{U}$ is not equivalent to the success criterion of merely recovering the defective set $\mathcal{K}$. It is true that if any algorithm correctly recovers $\boldsymbol{U}$, so that $\widehat{\boldsymbol{U}} = \boldsymbol{U}$, then it also recovers $\mathcal{K}$ as $\widehat{\mathcal{K}} = \{i : \widehat{U}_i < \infty\} = \mathcal{K}$. But the following example shows that for tropical COMP the converse does not hold; that is, just because COMP fails to recover $\boldsymbol{U}$, that does not necessarily mean it also fails to recover $\mathcal{K}$:

Example 3.6. Suppose we have two items, with true defectivity levels $\boldsymbol{U} = (1, 2)$. Suppose further that we run just one test, which contains both items, so $\boldsymbol{x} = (1, 1)$ and $\boldsymbol{Y} = (1)$. Then both items are in just one test with outcome $Y_1 = 1$, so have $\mu_1 = \mu_2 = 1$. Tropical COMP therefore incorrectly estimates $\widehat{\boldsymbol{U}} = (1, 1) \neq \boldsymbol{U}$. However, it does succeed in recovering the defective set $\widehat{\mathcal{K}} = \mathcal{K} = \{1, 2\}$.

Despite this, we show in Section 5 that tropical COMP asymptotically requires the same number of tests to recover $\boldsymbol{U}$ as classical COMP does to recover $\mathcal{K}$.

3.4 Tropical DD

We now describe the tropical DD algorithm. This extends the classical DD algorithm introduced in [2], which works in three steps:

1. Every item appearing in a negative test is non-defective. (All other items are 'possibly defective'.)

2. If a positive test contains a single possibly defective item, that item is defective.

3. All remaining items are assumed non-defective.

Tropical DD works the same way, except in step 2, it takes account of the different levels of defectivity in the tropical testing. Recalling that a PD(r) item is one with $\mu_i = r$, the tropical DD algorithm is as follows:

Algorithm 2: Tropical DD algorithm

```
Input: Test design matrix $\boldsymbol{x}$ and vector of test outcomes $\boldsymbol{Y}$
Output: Estimated vector of defectivity levels $\widehat{\boldsymbol{U}}$
for each item $i$ with $\mu_i = \infty$ do set $\widehat{U}_i = \infty$;
for each test $t$ with $Y_t = r < \infty$ do
    if there exists only one PD($r$) item $i$ in test $t$ then set $\widehat{U}_i = r$;
end
Declare all remaining unclassified items to have $\widehat{U}_i = \infty$;
```

To understand why this algorithm works, consider a test t with outcome $Y_t = r$. Observe that (by Definitions 2.1 and 2.2 respectively):

(a) test t cannot contain any items i with $U_i < r$, and must contain at least one 'special item' j with $U_j = r$;

(b) every item i appearing in test t has $\mu_i \geq r$.

Suppose that (apart from j) all the other items i in test t have $\mu_i > r$, so none of them are in PD(r). Then we know (by Lemma 2.3) that each such item has $U_i \geq \mu_i > r$, and cannot be a special item. Hence the remaining item j must be the special item that we seek. In other words, the sole PD(r) item in the test is marked as definitely defective at level r. This mirrors the classical case where if there is a single PD item in a test, it is marked as definitely defective.

We can think of the problem of finding $\mathcal{K}$ in the classical case as now being split into sub-problems of finding $\mathcal{K}_1, \ldots, \mathcal{K}_d$ in the tropical case. It is helpful to think of moving upwards through the rows in the block formulation of Figure 1:

1. By examining the tests with outcome ∞, we can identify H_∞ non-defective items and remove them from consideration for tests of outcome $r < \infty$.

2. In general, for $r = d, d-1, \ldots, 1$, by examining all the tests with outcome r, we hope to find all the defective items $i(r,1), \ldots, i(r, K_r)$ and to find the H_r non-defective items that are in PD(r) and remove them from consideration for tests of outcome lower than r.

We note that the operation of classical DD is the same as the operation of tropical DD when $d = 1$. We know that classical DD never makes false positive errors [4, Lemma 2.2]. The same is true for tropical DD:

Lemma 3.7. *Tropical DD never makes false positive errors. Indeed the only errors it can make is wrongly declaring a defective items of some finite level $U_i = r$ to be non-defective $\widehat{U}_i = \infty$.*

Proof. The first step finds some non-defective items from negative tests, and so is error-free. The second step identifies the sole defective item that can explain the outcome of the test it is in. It is thus also error-free. The final step is the only point at which errors can be made; specifically, false negative errors where a defective item is marked non-defective can occur. □

For tropical DD, the success criteria of recovering the vector $\boldsymbol{U}$ and recovering the defective set $\mathcal{K}$ are equivalent. We know that if an algorithm recovers $\boldsymbol{U}$, then it recovers $\mathcal{K}$. To prove equivalence of the success criteria, it suffices to show that if tropical DD fails to recover $\boldsymbol{U}$, then it fails to recovers $\mathcal{K}$. This is done in the following paragraph.

Suppose that tropical DD fails to recover $\boldsymbol{U}$. Then by Lemma 3.7, the only errors that could have been made are false negative errors where a defective item is wrongly marked non-defective. Hence, tropical DD also fails to recover $\mathcal{K}$.

3.5 Tropical SCOMP

We now describe the tropical SCOMP algorithm, which extends the classical SCOMP algorithm introduced in [2].

Classical SCOMP starts with the estimate given by the DD algorithm, but considers the fact that DD can leave some positive tests unexplained, since they can contain some defective items whose status we are unsure about. Since each such unexplained test must contain at least one such defective, we might consider possible defective items which appear in many unexplained tests as most likely to be defective. Hence the classical SCOMP algorithm greedily adds such items to the estimated defective set $\hat{\mathcal{K}}$ until all tests are explained.

Similarly, the tropical SCOMP algorithm starts with the estimate given by tropical DD. It then greedily adds items to the sets $\hat{\mathcal{K}}_r$, for each r such that there are unexplained tests of outcome r. This is done until all tests are explained.

```
Algorithm 3: Tropical SCOMP algorithm
Input: Test design matrix, $\boldsymbol{x}$, and vector of test outcomes $\boldsymbol{Y}$
Output: Estimated vector of defectivity levels, $\widehat{\boldsymbol{U}}$
Initialize $\widehat{\boldsymbol{U}}$ as the estimate $\widehat{\boldsymbol{U}}_{\mathrm{DD}}$ of $\boldsymbol{U}$ produced by the DD algorithm;
while unexplained tests exist do
    Choose a test outcome $r$ from the unexplained tests;
    Retrieve all the tests with outcome $r$;
    Find the PD($r$) item $i$ that occurs the most times in those tests (ties
      can be broken arbitrarily);
    Set $\widehat{U}_i = r$ and update the list of unexplained tests;
end
```

Note that tropical SCOMP attempts to solve the sub-problems of finding $\mathcal{K}_1, \ldots, \mathcal{K}_d$ that are not solved by tropical DD. The action of classical SCOMP is the same as that of tropical SCOMP when $d = 1$.

Remark 3.8. If tropical DD succeeds, then so does tropical SCOMP. This is because tropical SCOMP starts with the estimate produced by tropical DD. If tropical DD succeeds, then no tests are unexplained and tropical SCOMP also succeeds (cf. [4, Theorem 2.5]).

We show that the success criteria of recovering $\boldsymbol{U}$ and recovering $\mathcal{K}$ are equivalent for tropical SCOMP. Similar to the case of tropical DD, it suffices to show that if tropical SCOMP fails to recovers $\boldsymbol{U}$, then it fails to recovers $\mathcal{K}$. This is done in the following paragraph.

Suppose that tropical SCOMP fails to recover $\boldsymbol{U}$. Then necessarily, tropical DD also fails to recover $\boldsymbol{U}$. Then there exists an item $i \in \mathcal{K}$ such that there are no tests in which it is the only PD(μ_i) item. Since tropical SCOMP fails to recover $\boldsymbol{U}$, at least one such item i was not chosen to explain the test outcomes of any of the tests that it is in and is marked as non-defective. Hence, $\widehat{\mathcal{K}} \neq \mathcal{K}$.

3.6 Comparison of tropical algorithms

Table 1 summarises the features of the tropical algorithms, while comparing them to the classical algorithms (cf. [4, Table 2.1]). We now present a worked example which illustrates the operation of the various tropical algorithms:

Example 3.9. Suppose we use the test design $\boldsymbol{x}$ and receive the outcomes $\boldsymbol{Y}$ as

	satisfying	**no false +**	**no false −**
COMP	✓	✗	✓
DD	✗	✓	✗
SCOMP	✓	✗	✗
tropical COMP	✓	✗	✓
tropical DD	✗	✓	✗
tropical SCOMP	✓	✗	✗

Table 1: Summary of features of algorithms in the classical and tropical case: (i) whether the output estimate $\widehat{\boldsymbol{U}}$ is guaranteed to explain all test outcomes; (ii)–(iii) guarantees on false positives or false negatives.

follows:

$$\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad \boldsymbol{Y} = \begin{bmatrix} \infty \\ 37 \\ \infty \\ 29 \\ \infty \end{bmatrix}.$$

It is convenient to first calculate $\boldsymbol{\mu}$. For example, item 1 occurs in tests $1, 2, 5$. We then deduce

$$\mu_1 = \max_{t \in \{1,2,5\}} Y_t = \max\{\infty, 37, \infty\} = \infty.$$

Proceeding similarly for the other items, we obtain

$$\boldsymbol{\mu} = \big(\infty, \infty, 37, \infty, \infty, 29, 37\big).$$

Tropical COMP: We set $\widehat{\boldsymbol{U}} = \boldsymbol{\mu}$, obtaining the following:

$$\widehat{\boldsymbol{U}}^{\text{COMP}} = \big(\infty, \infty, 37, \infty, \infty, 29, 37\big).$$

Tropical DD: In the first step, we find the items with $\mu_i = \infty$. These are items $1, 2, 4$ and 5. We declare these to be non-defective, so $\widehat{U}_1^{\text{DD}} = \widehat{U}_2^{\text{DD}} = \widehat{U}_4^{\text{DD}} = \widehat{U}_5^{\text{DD}} = \infty$.

In the second step, we check each positive test t and look to see if they contain a single $\mathrm{PD}(Y_t)$ item. For test 2, there are two $\mathrm{PD}(Y_2) = \mathrm{PD}(37)$ items in the test, items 3 and 7, so DD does nothing. For test 4, items

$2, 5$ and 6 appear, but only item 6 is a $\mathrm{PD}(Y_4) = \mathrm{PD}(29)$ item. Hence, the tropical DD algorithm sets $\widehat{U}_6^{\mathrm{DD}} = 29$.

Finally, in the third step, items 3 and 7, which have not yet been classified, get assigned a defectivity level of $\widehat{U}_3^{\mathrm{DD}} = \widehat{U}_7^{\mathrm{DD}} = \infty$.

Hence, the output of the tropical DD algorithm is:

$$\widehat{\boldsymbol{U}}_{\mathrm{DD}} = \big(\infty, \infty, \infty, \infty, \infty, 29, \infty\big).$$

Tropical SCOMP: The algorithm initializes with the tropical DD estimate $\widehat{\boldsymbol{U}} = \widehat{\boldsymbol{U}}_{\mathrm{DD}}$. The corresponding outcome would be (written as the transpose, a row vector)

$$\widehat{\boldsymbol{Y}} = (\infty, \infty, \infty, 29, \infty)^\top,$$

where $\widehat{Y}_2 = \infty \neq 37 = Y_2$. Hence, test 2 is the only unexplained test. We retrieve the PD(37) items in test 2. These are items 3 and 7. Because these items both appear in the same number of tests with outcome 37, namely one, the tropical SCOMP algorithm chooses between them arbitrarily – let's say it chooses item 7 – and assigns the defectivity level of $\widehat{U}_7^{\mathrm{SCOMP}} = 37$ to it. Now no tests remain unexplained, and the algorithm terminates.

Hence the algorithm returns

$$\widehat{\boldsymbol{U}}^{\mathrm{SCOMP}} = \big(\infty, \infty, \infty, \infty, \infty, 29, 37\big).$$

4 Simulation results

In this section, we present some simulation results. We empirically compare the performance of the tropical and classical algorithms, and investigate how changing the probability p and the sequence $\boldsymbol{K} = (K_1, \ldots, K_d)$ affects their performance. We also investigate the effect of using a combinatorial model with random defectivity levels for defective items, as opposed to the model with fixed K_r introduced in Section 2. Finally, we compare the Bernoulli design and the near-constant column weight design, described in [4, Section 2.7].

Figure 2 shows the performance of the tropical algorithms, relative to the performance of the classical algorithms and to the counting bounds (7) and (8). Figure 2 shows for the chosen set of parameters that the tropical COMP algorithm performs almost identically to its classical counterpart (the lines are so close that they are difficult to distinguish), and the tropical DD and SCOMP algorithms perform better than their classical counterparts. We also notice that in this case tropical SCOMP beats the classical counting bound (7) for small values of T,

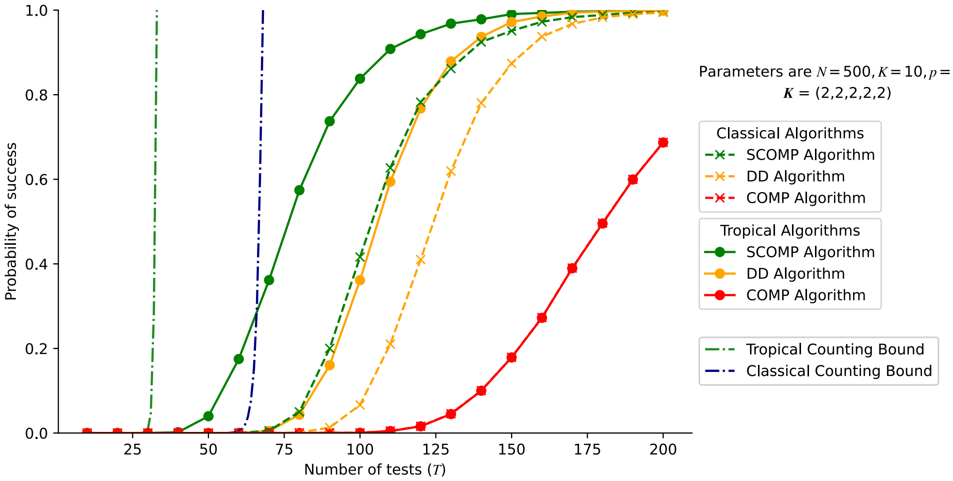

Figure 2: Empirical performance of the classical COMP, DD and SCOMP algorithms together with their tropical counterparts, through simulation with a Bernoulli design. For comparison, we plot the classical and tropical counting bounds of (7) and (8). The parameters chosen are $N = 500, K = 10, p = 0.1, \boldsymbol{K} = (2, 2, 2, 2, 2)$. Each point is obtained through 10^4 simulations.

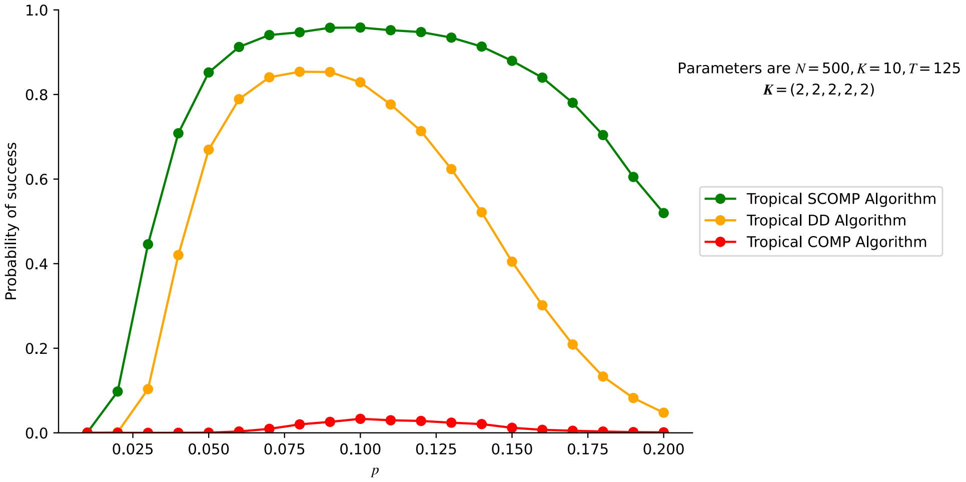

Figure 3: Simulation of the tropical COMP, DD and SCOMP algorithms with a Bernoulli design to investigate the effect of changing the parameter p for the Bernoulli design. The parameters are $N = 500, K = 10, T = 125$ and $\boldsymbol{K} = (2, 2, 2, 2, 2)$. Each point is obtained through 10^4 simulations.

showing that the tropical model can allow genuine performance gains over even adaptive classical group testing algorithms.

Figure 3 shows how the performance of the tropical algorithms vary with p, for fixed N, T and $\boldsymbol{K}$. Figure 3 shows that the performance of tropical DD and tropical SCOMP have a relatively wide plateau near their peak, indicating some robustness to misspecification of p, and showing that in general the choice $p = 1/K$ is close to optimal for each algorithm.

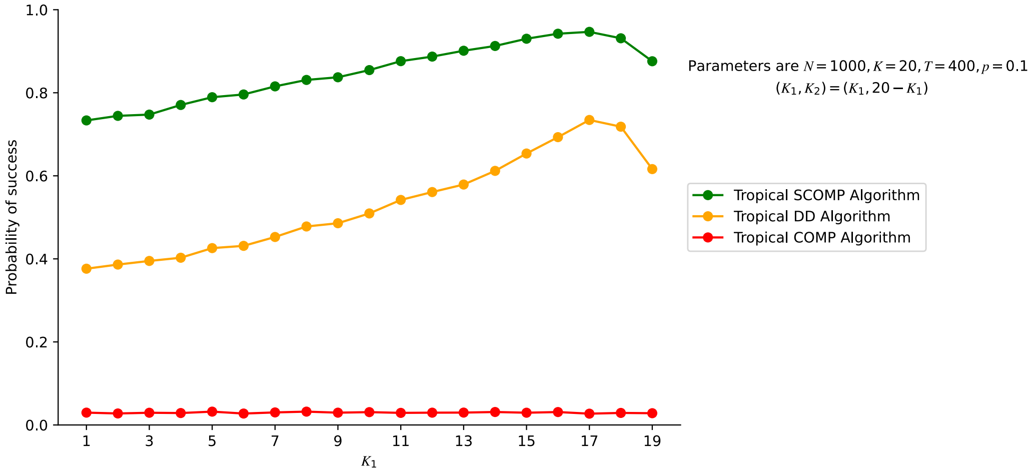

Figure 4: Simulation of the tropical COMP, DD and SCOMP algorithms with a Bernoulli design to investigate the effect of changing $\boldsymbol{K}$ on the performance. The parameters are $N = 1000, K = 20, T = 400, p = 0.1$ and $\boldsymbol{K} = (K_1, 20 - K_1)$. Each point is obtained through 10^4 simulations.

Figure 4 shows how the performance of the tropical algorithms vary as $\boldsymbol{K}$ is varied, for fixed N, K, T and d. We note that there are $\binom{K-1}{d-1}$ possible vectors $\boldsymbol{K}$ that sum to K while having each $K_i > 0$. Also, a d-dimensional plot is required to visualize the performance of the algorithms for all the $\boldsymbol{K}$ simultaneously. Hence, for simplicity of exposition, we only present the case $d = 2$. Figure 4 shows that changing $\boldsymbol{K}$ has very little effect on the performance of tropical COMP. This is quantified later in Section 5, where we find that, for tropical COMP, the error contribution of the K defective items is small compared to that of the $N - K$ non-defective items. Figure 4 also shows that the performance of tropical DD and tropical SCOMP improves as K_1 increases, until reaching a peak.

Figure 5 shows the effect on the performance of the tropical algorithms when the defective set $\mathcal{K}$ is chosen with a combinatorial prior, and the defectivity level for each defective item is drawn uniformly and independently from $\{1, \ldots, d\}$. We note that that the performance of tropical DD and tropical SCOMP improves as d increases, until reaching a peak, while the performance of tropical COMP does not change.

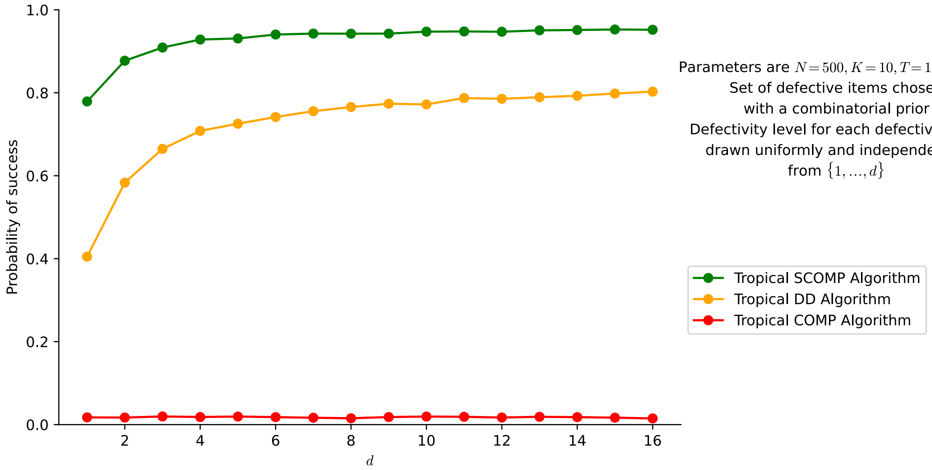

Figure 5: Simulation of the tropical COMP, DD and SCOMP algorithms to investigate their performance with a Bernoulli design when the set of defective items, $\mathcal{K}$, is chosen with a combinatorial prior, and the defectivity level for each defective item is drawn uniformly and independently from $\{1, \dots, d\}$. The parameters are $N = 500, K = 10, T = 120, p = 0.1$. Each point is obtained through 10^4 simulations.

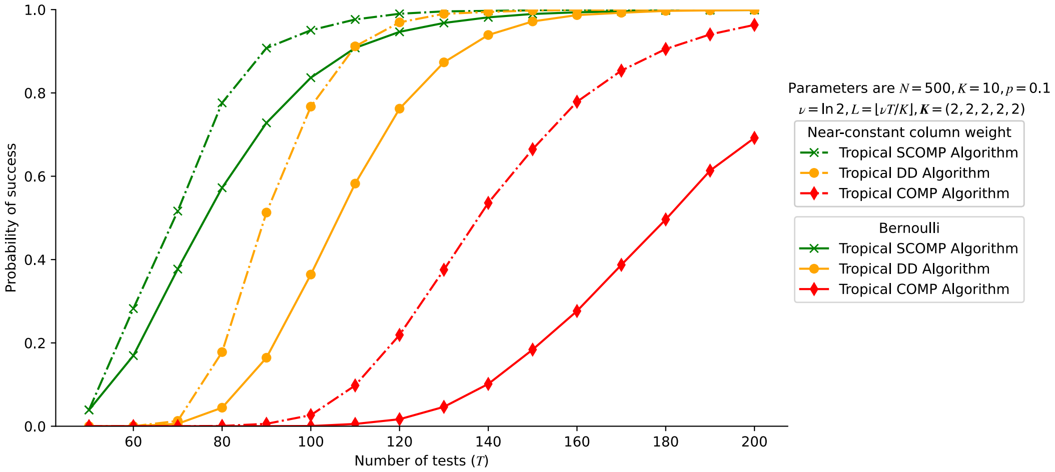

Figure 6: Simulation of the tropical COMP, DD and SCOMP algorithms to investigate their performance with a Bernoulli design as well as with a near-constant column weight design. The parameters are $N = 500, K = 10, p = 0.1, \nu = \ln 2, L = \lfloor \nu T/K \rfloor, \boldsymbol{K} = (2, 2, 2, 2, 2)$. Each point is obtained through 10^4 simulations.

Finally, Figure 6 compares the performance of the tropical algorithms with the Bernoulli design and with a near-constant column weight design. We see that the performance mirrors the classical case (cf. [4, Figure 2.3]).

5 Analysis of tropical COMP algorithm

In this section, we give an analysis of the performance of tropical COMP. The main result of this section (Theorem 5.1) shows that the number of tests needed to ensure a vanishing error probability using a Bernoulli design is asymptotically identical to that needed in the classical case.

5.1 Achievability result

Our main result for tropical COMP is the following (cf. [4, Eq. (2.9)]):

Theorem 5.1. *Let $\delta > 0$. Let $p = \nu/K$, for $0 < \nu < K$. Taking*

$$T \geq (1+\delta)\,\frac{\mathrm{e}^{\nu}}{\nu} K \ln N$$

ensures that the tropical COMP error probability $\mathbb{P}(\mathrm{err})$ is asymptotically at most $N^{-\delta}$.

Remark 5.2. Note that $T = (1+\delta)\frac{\mathrm{e}^{\nu}}{\nu} K \ln N$ is minimised over ν when $\nu = 1$. This corresponds to choosing the same optimal p as in the classical case. We note that tropical COMP, similar to classical COMP, is reasonably robust to misspecification of p (cf. [4, Remark 2.3]).

To reach the result of Theorem 5.1, we find a bound on the error probability of tropical COMP using a Bernoulli design. This bound, given below, extends the corresponding bound by Chan *et al.* for classical COMP, given in [7, Eq. (8)], to the tropical setting.

Lemma 5.3. *For a Bernoulli test design with parameter p, we can bound the error probability of Tropical COMP from above by*

$$\mathbb{P}(\mathrm{err}; \mathrm{COMP}, T) \leq \sum_{r \in \mathcal{D}} K_r (1 - p(1-p)^{\sum_{i<r} K_i})^T. \tag{10}$$

Proof. To obtain an upper bound on the error probability of tropical COMP, we consider each item in turn, using the fact that the union bound

$$\mathbb{P}(\mathrm{err}) = \mathbb{P}\left(\bigcup_i \{\widehat{U}_i \neq U_i\}\right) \leq \sum_i \mathbb{P}(\widehat{U}_i \neq U_i), \tag{11}$$

tells us that we only need to control the individual probabilities that an item is misclassified.

Any given item i with $U_i = r$ is misclassified only if it is intruding. This happens if every test which it appears in contains at least one of the $\sum_{i<r} K_i$ items of lower level. For a given test, the chance that it contains i but doesn't contain such an item is $p(1-p)^{\sum_{i<r} K_i}$. Hence using independence between tests, we have that

$$\mathbb{P}(\widehat{U}_i \neq U_i) \leq (1 - p(1-p)^{\sum_{i<r} K_i})^T. \tag{12}$$

The result follows on substituting (12) in (11). □

We can now prove the main result.

Proof of Theorem 5.1. This proof is adapted from the one given in the classical case by Chan *et al.* in [7]. Let $T = \beta K \ln N$. The key is to observe for a given T and p that the function $f(\ell) = (1 - p(1-p)^{\ell})^T$ is increasing in ℓ.

Hence we can write (10) as

$$\mathbb{P}(\text{err}) \leq \sum_{r \in \mathcal{D}} K_r f\left(\sum_{i<r} K_i\right) \leq \sum_{r \in \mathcal{D}} K_r f\left(K\right) = N f(K). \tag{13}$$

Then, setting $p = \nu/K$ in Lemma 5.3, we have

$$\begin{aligned}\mathbb{P}(\text{err}) &\leq N \exp(-Tp(1-p)^K) \\ &= N \exp(-\beta\nu(1-\nu/K)^K \ln N) \\ &\simeq N \exp(-\beta\nu \mathrm{e}^{-\nu} \ln N) \qquad \text{as } K \to \infty \\ &= N^{1-\beta\nu\mathrm{e}^{-\nu}}.\end{aligned}$$

Hence, taking $\beta = (1+\delta)\frac{\mathrm{e}^{\nu}}{\nu}$ ensures that $\mathbb{P}(\text{err})$ is asymptotically at most $N^{-\delta}$. □

5.2 Contributions to the error probability

Figure 7 illustrates the contribution of each summand to the bound (10) for a range of values of T. It is clear that the dominant term contributing to the error bound is $r = \infty$ (that the dominant error event is for a non-defective item to be wrongly classified as defective, and that the defective items are more typically correctly classified).

Indeed, (13) implies that the proportion of the bound (10) arising from the $r = \infty$ term is at least $1 - K/N$, since this result gives

$$\frac{(N-K)f(K)}{\sum_{r\in\mathcal{D}} K_r f\left(\sum_{i<r} K_i\right)} \geq \frac{(N-K)f(K)}{Nf(K)} = 1 - \frac{K}{N}.$$

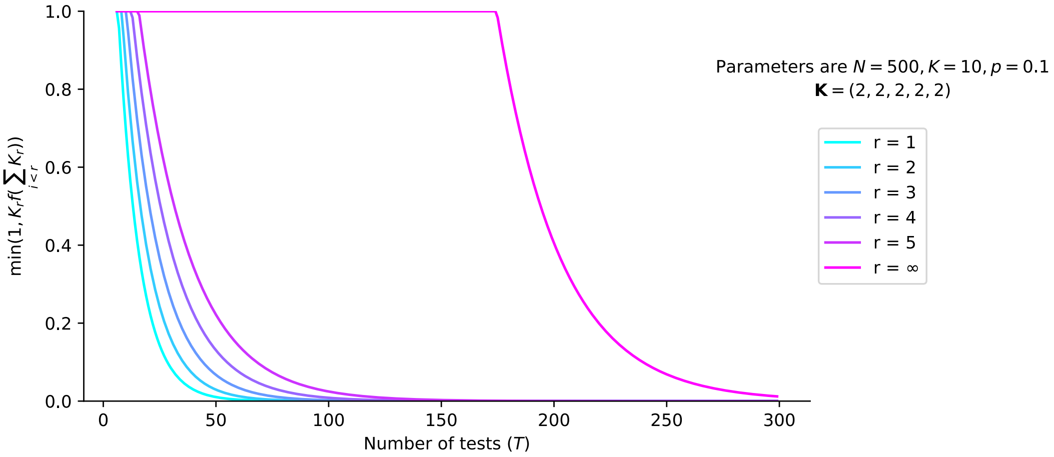

Figure 7: Plot illustrating the variation of $\min\{1, K_r f(\sum_{i<r} K_r)\}$ with T, for each $r \in \mathcal{D}$. The parameters chosen are $N = 500, K = 10, p = 0.1$ and $\boldsymbol{K} = (2, 2, 2, 2, 2)$.

In fact in the 'uniform' case where $\boldsymbol{K} = (K/d, \dots, K/d)$, the contributions to the bound

$$K_r f\left(\sum_{i<r} K_i\right) \simeq K_r \exp\left(-Tp(1-p)^{(r-1)K/d}\right) \simeq \frac{K}{d} \exp\left(-Tp\mathrm{e}^{-p(r-1)K/d}\right) \quad (14)$$

decay doubly-exponentially as r gets smaller, meaning that the errors are overwhelmingly likely to arise from wrongly classifying items with high r.

5.3 Error probability for different defectivity sequences

For a fixed number of defective items K, it would be interesting to know what defectivity sequences $\boldsymbol{K}$ make the tropical group testing problem hardest or easiest. This is explored via simulation in Figure 4, but we would also like to understand which sequences $\boldsymbol{K}$ lead to the largest and smallest error probability for tropical COMP. Unfortunately, we cannot directly control the error probability in this way. However, we can use the COMP error bound (10) to induce a partial order on sequences $\boldsymbol{K}$ with the same sum. This will show that the error bound is smallest in the classical case where all items have the same level and largest in the case where each item has distinct levels.

Given a sequence $\boldsymbol{K} = (K_1, \dots, K_d)$, we can sort the items in increasing order of level, and for each item k write L_k for the number of items with a strictly lower level. For example, with $K = 8$, the sequence $\boldsymbol{K}^{(1)} = (2, 2, 2, 2)$ induces the sequence $\boldsymbol{L}^{(1)} = (0, 0, 2, 2, 4, 4, 6, 6)$, while the sequence $\boldsymbol{K}^{(2)} = (1, 1, 1, \ldots, 1)$ induces the sequence $\boldsymbol{L}^{(2)} = (0, 1, 2, 3, 4, 5, 6, 7)$.

We can compare two sequences $\boldsymbol{K}^{(i)} = \big(K_1^{(i)}, \ldots, K_d^{(i)}\big)$, for $i = 1, 2$, and define a partial order $\boldsymbol{K}^{(1)} \preceq \boldsymbol{K}^{(2)}$ if the corresponding sequences $L_k^{(1)} \leq L_k^{(2)}$ for all k. Hence in the example above, $\boldsymbol{K}^{(1)} \preceq \boldsymbol{K}^{(2)}$. In this partial ordering, for a given K the minimal sequence will be $\boldsymbol{K} = (K)$, giving $\boldsymbol{L} = (0, 0, \ldots, 0)$, and the sequence $\boldsymbol{K} = (1, 1, \ldots, 1)$ as seen above will be maximal.

Now, note that the bound on the RHS of (10) respects this partial order. That is, since the $r = \infty$ term will be the same for all such sequences, we can regard the variable part of the bound (13) as a sum over defective items

$$\sum_{k \in \mathcal{K}} f(L_k), \tag{15}$$

and use the fact that the function $f(\ell)$ is increasing in ℓ to deduce that:

Lemma 5.4. *If $\boldsymbol{K}^{(1)} \preceq \boldsymbol{K}^{(2)}$ then the corresponding error bound (15) is lower for $\boldsymbol{K}^{(1)}$ than $\boldsymbol{K}^{(2)}$.*

Hence, for fixed K the error bound (15) is smallest for the minimal sequence $\boldsymbol{K} = (K)$ corresponding to the classical case and largest for the sequence $\boldsymbol{K} = (1, 1, \ldots, 1)$ where each defective item has its own unique level.

6 Analysis of tropical DD algorithm: achievability

In this section we give an analysis of the performance of tropical DD, which extends that given in [2] for the classical $d = 1$ case by taking advantage of the information provided by the more varied test outcomes of the tests.

Our main achievability result ensures success with high probability when we have a number of tests above a certain threshold.

Theorem 6.1. *For $\nu > 0$, write*

$$\psi_r := \psi_r(\nu) = \left(1 - \frac{\nu}{K}\right)^{\sum_{t \leq r} K_t}.$$

Also define

$$T_\infty(\nu) := \frac{1}{\nu \psi_d} K \ln \frac{N}{K}$$

and

$$T_r(\nu) := \frac{1}{\nu \psi_r} K \ln K_r.$$

If we take

$$T \geq (1+\delta) \max \big\{ T_\infty(\nu), T_d(\nu), T_{d-1}(\nu), \dots, T_1(\nu) \big\},$$

tests, then the error probability of the tropical DD algorithm for a Bernoulli design with $p = \nu/K$ tends to 0.

Remark 6.2. Note that in the uniform case where $K_r = K/d$ for all r, in regimes where $K = N^\alpha$ and $\alpha > 1/2$, the dominant term in Theorem 6.1 is

$$T_d(\nu) = \frac{1}{\nu(1-\nu/K)^K} K \ln \frac{K}{d};$$

that is, the maximum over r is achieved at $r = d$, since ψ_r is decreasing in r and $\ln K_r$ is constant.

Further, in this case, since we are able to choose ν to minimise $T_{\text{fin}}(\nu)$, we can maximise $\nu(1-\nu/K)^K$ by taking $\nu = K/(K+1)$ or $p = 1/(K+1)$. Note that this does not necessarily mean that we minimise the error probability, since we are minimising the number of tests to control a bound on $\mathbb{P}(\text{err})$, not $\mathbb{P}(\text{err})$ itself. In other words, asymptotically we require $\mathrm{e}K \ln(K/d)$ tests.

This means that the tropical performance bound of Theorem 6.1 represents an asymptotic reduction of $\mathrm{e}K \ln d$ tests over the classical bound of $\mathrm{e}K \ln K$ tests (see [1, Theorem 1]). While this is a second-order asymptotic term compared with the leading order $\mathrm{e}K \ln K$ term, it may still represent a valuable improvement in problems of finite size.

In the next section, we will see a converse result Theorem 7.1, showing that success probability can't be high with a number of tests below a threshold. Further, we show that for certain parameter regimes these two thresholds coincide, showing that we have sharp performance bounds on tropical DD.

6.1 Proof outline

To prove Theorem 6.1, we need a general upper bound on the error probability of DD. The key idea is that DD succeeds if and only if each defective item is proven to be such in the second stage of the algorithm.

Definition 6.3. For any $1 \leq s \leq K_r$, we write $L_{r,s}$ for the number of tests that contain item $i(r,s)$, no other defective item $i(t,u)$ with $t \leq r$, and also no non-defective $\mathrm{PD}(r)$ item.

A test that counts towards $L_{r,s}$ is precisely one that discovers $i(r,s)$ to be defective at level r. So with this definition, we can say that the tropical DD algorithm succeeds if and only if $L_{r,s} \geq 1$ for all (r,s). Hence we have

$$\mathbb{P}(\text{err}) = \mathbb{P}\left(\bigcup_{r,s} \{L_{r,s} = 0\} \right) \leq \sum_{r=1}^{d} \mathbb{P}\left(\bigcup_{s=1}^{K_r} \{L_{r,s} = 0\} \right), \tag{16}$$

One way we could get $L_{r,s} = 0$ is if there is a large number of potentially intruding non-defectives at this outcome level. We define $G_r = \sum_{j=0}^{r} H_j$ for the number of intruding non-defectives (where H_j is introduced in Definition 2.4.3) and note that there are $(N - K) - G_r = H_{r+1} + \ldots + H_d + H_\infty$ non-defective items with $\mu_i > r$. We observe that, provided we use sufficiently many tests, G_r is unlikely to be large. Hence, we will condition on G_r being no larger than some threshold level g_r^*, to be chosen later. So for each level r, the summand in (16) can be bound as

$$\begin{aligned}
\mathbb{P}\left(\bigcup_{s=1}^{K_r}\{L_{r,s}=0\}\right) &= \mathbb{P}\left(\bigcup_{s=1}^{K_r}\{L_{r,s}=0\} \,\middle|\, G_r \le g_r^*\right)\mathbb{P}(G_r \le g_r^*) \\
&\quad + \mathbb{P}\left(\bigcup_{s=1}^{K_r}\{L_{r,s}=0\} \,\middle|\, G_r > g_r^*\right)\mathbb{P}(G_r > g_r^*) \\
&\le \mathbb{P}\left(\bigcup_{s=1}^{K_r}\{L_{r,s}=0\} \,\middle|\, G_r \le g_r^*\right) + \mathbb{P}(G_r > g_r^*) \\
&\le K_r\,\mathbb{P}\left(L_{r,s}=0 \mid G_r \le g_r^*\right) + \mathbb{P}(G_r > g_r^*), \qquad (17)
\end{aligned}$$

where we used the union bound in the last line.

We need to show that both terms in (17) are small. The first term being small tells us we're likely to find the level-r defectives provided G_r is not too big; we will show this happens when $T \ge (1+\delta)\max\{T_\infty, T_r\}$. The second term being small tells us that G_r is unlikely to be too big; we will show this happens when T is big; for example, $T \ge (1+\delta)T_r$ will be plenty.

In Subsection 6.2 we will analyse the first term $\mathbb{P}\left(L_{r,s}=0 \mid G_r \le g_r^*\right)$. In Subsection 6.3 we will bound the second term $\mathbb{P}(G_r > g_r^*)$. Then in Subsection 6.4 we will put the pieces together to prove Theorem 6.1.

6.2 Finding defectives

We first describe the joint distribution of certain random variables arising in the analysis of DD. We provide additional notation to that used in Section 2.

Definition 6.4. We define the following random variables:

1. Write M_∞ for the number of tests which contain no defectives – and so are negative tests with outcome ∞.

2. For $1 \le r \le d$, write M_r for the total number of positive tests with outcome r.

3. Further, decompose $M_r = \sum_{s=1}^{K_r} M_{r,s} + M_{r,+}$ as follows:

(a) For $1 \leq s \leq K_r$, write $M_{r,s}$ for the number of tests that contain a single item $i(r,s)$ at level r and no other defective item $i(t,u)$ with $t \leq r$; note that each such test has outcome r.

(b) Write $M_{r,+}$ for the number of tests which have outcome r but contain multiple defective items at level r.

Write $\boldsymbol{M}$ for the collection of random variables

$$\boldsymbol{M} = (M_{1,1}, M_{1,2}, \dots M_{1,K_1}, M_{1,+}, \dots, M_{d,1}, M_{d,2}, \dots, M_{d,K_d}, M_{d,+}, M_\infty)$$

(noting this includes the terms in the decompositions of the M_r variables, but not the M_r themselves).

Note that $M_{r,s} = 0$ means necessarily that $L_{r,s} = 0$, and this is the event we wish to avoid. But first, let us note the joint distribution of $\boldsymbol{M}$.

Lemma 6.5. *The random vector $\boldsymbol{M}$ is multinomial with parameters T and $\boldsymbol{q}$, where*

$$\boldsymbol{q} = (q_{1,1}, q_{1,2}, \dots q_{1,K_1}, q_{1,+}, \dots, q_{d,1}, q_{d,2}, \dots, q_{d,K_d}, q_{d,+}, q_\infty) .$$

Here for each r and s:

$$\begin{aligned} q_\infty &:= (1-p)^K, \\ q_{r,s} &:= \prod_{t<r} (1-p)^{K_t} \left((1-p)^{K_r - 1} p \right), \\ q_r &:= \prod_{t<r} (1-p)^{K_t} \left(1 - (1-p)^{K_r} \right), \\ q_{r,+} &:= q_r - K_r q_{r,s}. \end{aligned}$$

Proof. First, a test is negative if all defective items are absent, with happens with probability $q_\infty = (1-p)^K$. Second, $q_{r,s}$ is the probability all items at levels $t < r$ are absent, that item $i(r,s)$ is present, and that the $K_r - 1$ other items at level r are absent. Third, q_r is the probability of outcome r, which happens all items at level $t < r$ are absent, and also it's not the case that all items at level r are absent. Fourth, $q_{r,+}$ is the probability q_r of an outcome r minus the probabilities of a single level-r item being the cause. □

Although the distribution of the crucial variable $L_{r,s}$ seems tricky to derive from first principles, it is much easier once we know $M_{r,s}$ and the number of potentially intruding non-defectives. This is because a test counting towards $M_{r,s}$ will count also towards $L_{r,s}$ provided that no non-defectives intrude on the test too.

Lemma 6.6. *The conditional distribution of $L_{r,s}$, given $M_{r,s}$ and G_r, is*

$$L_{r,s} \mid \{M_{r,s} = m_{r,s}, G_r = g_r\} \sim \mathrm{Bin}(m_{r,s}, (1-p)^{g_r}). \tag{18}$$

Proof. There are $m_{r,s}$ tests which contain item $i(r,s)$ and no other defective item $i(t,u)$ with $t \leq r$. Each of these $m_{r,s}$ tests independently contributes to $L_{r,s}$ if and only if none of the g_r potentially intruding non-defective items appear in the test. Because of the Bernoulli design, each of those g_r non-defectives appears in the test with probability p, so the probability none of them appear is $(1-p)^{g_r}$, independent over the $m_{r,s}$ tests. □

We can now bound the probability that of undesirable event that $L_{r,s} = 0$.

Lemma 6.7. *Using a Bernoulli design with parameter p, for any g_r^*, we have the bound*

$$\mathbb{P}(L_{r,s} = 0 \mid G_r \leq g_r^*) \leq \exp\big(-q_{r,1}(1-pg_r^*)T\big), \tag{19}$$

where as in Lemma 6.5 we write $q_{r,1} = p(1-p)^{\sum_{t \leq r} K_t - 1}$.

Proof. We start by conditioning on equality $G_r = g_r$. Noting that by Lemma 6.5 $M_{r,s} \sim \mathrm{Bin}(T, q_{r,1})$ and that $\mathbb{P}(\mathrm{Bin}(m,q) = 0) = (1-q)^m$, we can write

$$\begin{aligned}
\mathbb{P}(L_{r,s} = 0 \mid G_r = g_r) &= \sum_{m=0}^{T} \mathbb{P}(M_{r,s} = m)\, \mathbb{P}(L_{r,s} = 0 \mid G_r = g_r, M_{r,s} = m) \\
&= \sum_{m=0}^{T} \binom{T}{m} q_{r,1}^m (1-q_{r,1})^{T-m} \left(1-(1-p)^{g_r}\right)^m \\
&= \left(1 - q_{r,1}(1-p)^{g_r}\right)^T \\
&\leq \exp\left(-q_{r,1}(1-p)^{g_r} T\right) \\
&\leq \exp\left(-q_{r,1}(1-pg_r)T\right). \qquad (20)
\end{aligned}$$

From the second to the third line, we used the binomial theorem, and then we used Bernoulli's inequality in the form $(1-p)^g \geq 1 - pg$.

Note that (20) is increasing in g_r. Thus we can bound (19) by the worst-case conditioning, where $G_r = g_r^*$. □

6.3 Intruding non-defectives

Recall that G_r is the number of non-defectives that could intrude into tests with outcome r. Our goal is to bound that tail probability of G_r in (17).

Lemma 6.8. *Write*

$$\overline{M}_r := M_\infty + M_d + M_{d-1} + \cdots + M_{r+1}$$

for the number of tests with outcomes higher than r*. Then* $\overline{M}_r$ *has distribution*

$$\overline{M}_r \sim \text{Bin}\left(T,\ \prod_{t\le r}(1-p)^{K_t}\right) \tag{21}$$

Further, the conditional distribution of G_r *given* $\boldsymbol{M}$ *is*

$$G_r \mid \{\boldsymbol{M} = \boldsymbol{m}\} \sim \text{Bin}(N-K, (1-p)^{m_r^*}), \tag{22}$$

where $m_r^* = m_{r+1} + \cdots + m_d + m_\infty$.

Proof. By standard properties of the multinomial (see [2, Lemma 30]),

$$\overline{M}_r \sim \text{Bin}\left(T, q_{r+1} + \ldots + q_d + q_\infty\right).$$

But

$$q_{r+1} + \ldots + q_d + q_\infty = \prod_{t\le r}(1-p)^{K_t}, \tag{23}$$

since it forms a collapsing sum. This proves the first distribution.

Given $\overline{M}_r = m_r^*$, each of the $N-K$ non-defectives will be independently counted in G_r provided they don't appear in any of the m_r^* tests with outcomes higher than r. By the Bernoulli design structure, each item is independently counted with probability $(1-p)^{m_r^*}$. This proves the second distribution. □

We can calculate the expectation of G_r by conditioning on $\overline{M}_r$.

Lemma 6.9. $\mathbb{E}G_r \le (N-K)\exp(-p\psi_r T)$.

Proof. We use the facts that (21) gives that $\overline{M}_r \sim \text{Bin}(T, \psi_r)$ and Lemma 6.8 gives that $G_r \mid \{\overline{M}_r = m_r^*\} \sim \text{Bin}\left(N-K, (1-p)^{m_r^*}\right)$. Hence we can use the binomial theorem to write

$$\begin{aligned}
\mathbb{E}G_r &= \sum_{m=0}^{T} \mathbb{P}(\overline{M}_r = m)\,\mathbb{E}[G_r \mid \overline{M}_r = m] \\
&= \sum_{m=0}^{T} \binom{T}{m} \psi_r^m (1-\psi_r)^{T-m}\,(N-K)(1-p)^m \\
&= (N-K)(1-\psi_r + \psi_r(1-p))^T,
\end{aligned}$$

and the result follows. □

We will choose the threshold g_r^* to be just slightly bigger than this expectation; specifically, we take $g_r^* = (N-K)\exp(-p\psi_r T(1-\epsilon))$ for some $\epsilon > 0$ to be determined later.

Lemma 6.10. *With $g_r^* = (N-K)\exp(-p\psi_r T(1-\epsilon))$, we have*

$$\mathbb{P}(G_r > g_r^*) \leq \exp(-p\psi_r T\epsilon).$$

Proof. This is a simple application of Markov's inequality. Using Lemma 6.9, we get

$$\begin{aligned}\mathbb{P}(G_r > g_r^*) &\leq \frac{\mathbb{E}G_r}{g_r^*} \\ &\leq \frac{(N-K)\exp(-p\psi_r T)}{(N-K)\exp(-pT\psi_r(1-\epsilon))} \\ &= \exp(-p\psi_r T\epsilon).\end{aligned}$$

□

6.4 Completing the proof

We are now ready to complete the proof of our main result.

Proof of Theorem 6.1. From (16) and (17), we had got as far as the bound

$$\mathbb{P}(\text{err}) \leq \sum_{r=1}^{d} \big[K_r\,\mathbb{P}\left(L_{r,s} = 0 \mid G_r \leq g_r^*\right) + \mathbb{P}(G_r > g_r^*)\big], \tag{24}$$

and in Subsection 6.3 we had chosen $g_r^* = (N-K)\exp(-p\psi_r T(1-\epsilon))$, with ϵ still to be fixed. We need to show that, in each summand of (24), both the first and second terms tend to 0.

We begin with the first term. From Lemma 6.7, we have the bound

$$\begin{aligned}K_r\,\mathbb{P}(L_{r,s} = 0 \mid G_r \leq g_r^*) &\leq K_r \exp\big(-q_{r,1}(1-pg_r^*)T\big) \\ &= \exp\left(\ln K_r - T\psi_r p\frac{1-pg_r^*}{1-p}\right),\end{aligned}$$

where we have used that $q_{r,1} = \psi_r p/(1-p)$. The condition $T \geq (1+\delta)T_r$ means that $T\psi_r p \geq (1+\delta)\ln K_r$, so we get

$$K_r\,\mathbb{P}(L_{r,s} = 0 \mid G_r \leq g_r^*) \leq \exp\left(\ln K_r\left(1-(1+\delta)\frac{1-pg_r^*}{1-p}\right)\right).$$

This tends to 0 so long as pg_r^* tends to 0, which we will now check.

Since $T \geq (1+\delta)T_\infty$ and $\psi_r \geq \psi_d$, we know that $Tp\psi_r \geq (1+\delta)\ln(N/K)$. With $g_r^* = (N-K)\exp(-p\psi_r T(1-\epsilon))$, we therefore have

$$pg_r^* \leq \left(\frac{N}{K}\right)\exp\left(-Tp\psi_r(1-\epsilon)\right) \leq \left(\frac{N}{K}\right)^{1-(1-\epsilon)(1+\delta)}.$$

This means that $pg_r^* \to 0$ is indeed guaranteed by choosing $\epsilon < \delta/(1+\delta)$.

Now the second term. From Lemma 6.10, we have

$$\mathbb{P}(G_r > g_r^*) \leq \exp(-p\psi_r T\epsilon) < \exp(-p\psi_r T\delta/(1+\delta)),$$

since we have just chosen $\epsilon < \delta/(1+\delta)$. The condition $T \geq (1+\delta)T_r$ gives us $p\psi_r T/(1+\delta) = \ln K_r$, so this term does indeed tend to 0.

Since we have shown that all the terms in (24) tend to zero, the proof is complete. □

7 Converse results

Our achievability result Theorem 6.1 shows that tropical DD can succeed with

$$T \geq (1+\delta)\max\{T_\infty(\nu), T_d(\nu), T_{d-1}(\nu), \dots, T_1(\nu)\} \qquad \text{tests.} \tag{25}$$

We can use similar ideas to provide a converse result for the tropical DD algorithm.

Theorem 7.1. *For a given $\nu > 0$, in the limiting regime where*

$$T \leq (1-\delta)\max\{T_d(\nu), T_{d-1}(\nu), \dots, T_1(\nu)\}$$

then the error probability of the tropical DD algorithm for a Bernoulli design with $p = \nu/K$ tends to 1.

Note that the difference between the achievablility and the converse results is the lack of the T_∞ term in the converse.

We will prove Theorem 7.1 for tropical DD in Subsection 7.1. In addition, we will show in Subsection 7.2 that this same bound acts as a more general 'algorithm-independent' converse for Bernoulli designs.

7.1 Proof for tropical DD

The key to proving Theorem 7.1 is to observe that tropical DD will definitely fail if any $M_{r,s} = 0$, since that means that item $i(r,s)$ never appears without at least one other defective item with which it could be confused, so $L_{r,s}$ is certainly 0 too.

Thus we start with the bound

$$\mathbb{P}(\mathrm{err}) \geq \mathbb{P}\left(\bigcup_{r,s}\{M_{r,s}=0\}\right). \tag{26}$$

By picking out just the defective items at a given level $r = r^*$, we have

$$\mathbb{P}(\mathrm{err}) \geq \mathbb{P}\left(\bigcup_{s=1}^{K_{r^*}}\{M_{r^*,s}=0\}\right) \tag{27}$$

As in [2, Eq. (13)], we define the function

$$\phi_K(q,T) := \sum_{j=0}^{K}(-1)^j\binom{K}{j}(1-jq)^T. \tag{28}$$

We will bound the error probability in terms of ϕ_K as follows.

Lemma 7.2. *For $1 \leq r^* \leq d$, the error probability of tropical DD is bounded below by*

$$\mathbb{P}(\mathrm{err}) \geq 1 - \phi_{K_r^*}(q_{r^*,1},T), \tag{29}$$

Proof. We follow the general idea from [2].

We can calculate (27) using the inclusion–exclusion formula

$$\mathbb{P}\left(\bigcup_{s=1}^{K_{r^*}}\{M_{r^*,s}=0\}\right) = \sum_{|S|=0}^{K_{r^*}}(-1)^{|S|}\,\mathbb{P}\left(\bigcap_{s\in S}\{M_{r^*,s}=0\}\right), \tag{30}$$

where the sum is over subsets S of $\mathcal{K}_r$. By the multinomial distribution form of Lemma 6.5, we have

$$\begin{aligned}\mathbb{P}\left(\bigcap_{s\in S}\{M_{r^*,s}=0\}\right) &= \binom{T}{0,0,\dots,0,T}\left(\prod_{s\in S}q_{r^*,s}^0\right)\left(1-\sum_{s\in S}q_{r^*,s}\right)^T \\ &= \left(1-\sum_{s\in S}q_{r^*,s}\right)^T \\ &= (1-|S|q_{r^*,1})^T. \end{aligned} \tag{31}$$

Substituting (31) into (30) gives

$$\mathbb{P}\left(\bigcup_{s=1}^{K_{r^*}}\{M_{r^*,s}=0\}\right) = \sum_{|S|=0}^{K_{r^*}}(-1)^{|S|}\,(1-|S|q_{r^*,1})^T.$$

Collecting together the summands according to the value of $|S| = j$ gives the result. □

We bound this quantity from below by deducing an upper bound on ϕ_K.

Lemma 7.3. *For all K, q and T we can bound*

$$\phi_K(q,T) \leq \exp\left(-\frac{K(1-q)^{T+1}}{1+Kq(1-q)^T}\right) \tag{32}$$

Proof. See Appendix B.1. □

We can now finish our proof of the converse for tropical DD.

Proof of Theorem 7.1. By hypothesis, $T \leq (1-\delta)\max_r T_r$. So pick some level r^* such that $T \leq (1-\delta)T_{r^*}$.

We had already reached the bound (29):

$$\mathbb{P}(\text{err}) \geq 1 - \phi_{K_r^*}(q_{r^*,1}, T).$$

We now combine this with Lemma 7.3. We deduce that

$$\mathbb{P}(\text{err}) \geq 1 - \exp\left(-\frac{K_{r^*}(1-q_{r^*,1})^{T+1}}{1+K_{r^*}q_{r^*,1}(1-q_{r^*,1})^T}\right). \tag{33}$$

The exponential term here is of the form $\exp(-(1-q)u/(1+qu))$, with $u = K_{r^*}(1-q_{r^*,1})^T$. Since $\exp(-(1-q)u/(1+qu))$ increases as u decreases (for fixed q), it suffices to bound $K_{r^*}(1-q_{r^*,1})^T$ from below, which we do now.

Since $q_{r^*,1} = p\psi_{r^*}/(1-p)$ we know that

$$\frac{q_{r^*,1}}{1-q_{r^*,1}} = \frac{p\psi_{r^*}}{1-p(1+\psi_{r^*})}.$$

Combining this with $T \leq (1-\delta)T_{r^*} = (1-\delta)K\ln K_{r^*}/(\nu\psi_{r^*})$ gives

$$\begin{aligned}\frac{Tq_{r^*,1}}{1-q_{r^*,1}} &\leq \frac{(1-\delta)P}{\nu\psi_{r^*}}\frac{p\psi_{r^*}}{1-p(1+\psi_{r^*})}\ln K_{r^*} \\ &= \frac{(1-\delta)}{1-p(1+\psi_r^*)}\ln K_r^* \\ &\leq (1-c)\ln K_r^*, \end{aligned} \tag{34}$$

for some $c > 0$, for K_{r^*} sufficiently large. This gives us the lower bound

$$\begin{aligned} K_{r^*}(1-q_{r^*,1})^T &= K_{r^*}\exp(T\log(1-q_{r^*,1})) \\ &\geq K_{r^*}\exp\left(-\frac{Tq_{r^*,1}}{1-q_{r^*,1}}\right) \\ &\geq K_{r^*}^c, \end{aligned} \tag{35}$$

where we used $\log(1-q) \geq -q/(1+q)$ for $q > -1$ and (34).

Using the bound (35) in (33), we get

$$\begin{aligned} \mathbb{P}(\text{err}) &\geq 1 - \exp\left(-\frac{K_{r^*}(1-q_{r^*,1})^{T+1}}{1+q_{r^*,1}K(1-q_{r^*,1})^T}\right) \\ &\geq 1 - \exp\left(-\frac{K^c_{r^*}(1-q_{r^*,1})}{1+q_{r^*,1}K^c_{r^*}}\right) \\ &= 1 - \exp\left(-\frac{(1-\psi_{r^*}/(K-1))K^c_{r^*}}{1+\psi_{r^*}K^c_{r^*}/(K-1)}\right). \end{aligned} \tag{36}$$

where (36) follows since $q_{r^*,1} = p\psi_{r^*}/(1-p) = \psi_{r^*}/(K-1)$.

Finally, that bound (36) is asymptotically equivalent to $1-\exp(-K^c_{r^*})$, which tends to 1 as $K \to \infty$. This completes the proof. □

7.2 Algorithm-independent converse

In fact, our DD-specific converse, Theorem 7.1, helps give a converse bound for *all* algorithms with a Bernoulli design.

We can write $\mathbb{P}(\text{err}; \text{optimal}, T)$ for the minimum error probability that can be achieved by any algorithm. The key observation is that in Theorem 7.1 we find that with too few tests there is a good chance that some item $i(r^*, s)$ appears in no tests without other items of the same level, so a satisfying vector can be formed without it.

Theorem 7.4. *For a given $\nu > 0$ and any $T \leq (1-\delta)T_{\text{fin}}(\nu)$, the error probability of the optimal algorithm $\mathbb{P}(\text{err}; \text{optimal}, T)$ for a Bernoulli design with parameter ν/K is bounded away from zero, even for algorithms which are given access to the values of K_i.*

Proof. First note that for any $T' \geq 1$ the $\mathbb{P}(\text{err}; \text{optimal}, T) \geq \mathbb{P}(\text{err}; \text{optimal}, T + T')$, since there exists a (possibly suboptimal) algorithm using $T + T'$ tests which simply ignores the last T' tests and applies the optimal T-test algorithm to the remaining tests. Hence it will be sufficient to bound $\mathbb{P}(\text{err}; \text{optimal}, T)$ away from zero for $T = (1-\delta)\max_r T_r$, as the same bound will hold for all $T \leq (1-\delta)\max_r T_r$.

We argue as in [1]. Recall that $M_{r,s}$ is the number of tests that have a chance of proving $i(r,s)$ is defective at level r, and H_r is the number of non-defective items in $\text{PD}(r)$. The key idea is this: Suppose for some r^* that both $A = \bigcup_{s=1}^{K_{r^*}}\{M_{r^*,s} = 0\}$ and $B = \{H_{r^*} \geq 1\}$ occur. The event A means that there is some item $i(r^*, t)$ which never appears without some other item j of level $U_j \leq r$, and the event B means that there is some non-defective item which is a possible defective at that level. So we could form an alternative satisfying vector from the true vector $\boldsymbol{U}$ by swapping the entries in $\boldsymbol{U}$ of these two items. Hence, if $A \cap B$ occurs, then there are at least two satisfying vectors with the correct number of items at each level, so we the success probability can only be at most $1/2$.

The probability of an intersection can always be bounded with $\mathbb{P}(A \cap B) \geq \mathbb{P}(A) - \mathbb{P}(B^c)$, so the error probability for any algorithm satisfies

$$\mathbb{P}(\mathrm{err}) \geq \frac{1}{2}\,\mathbb{P}\left(\bigcup_{s=1}^{K_{r^*}} \{M_{r^*,s} = 0\}\right) - \frac{1}{2}\,\mathbb{P}(H_{r^*} = 0).$$

Now the first term involves exactly the term we have controlled in Theorem 7.1, so we know it is larger than $1/4$ in the regime of interest for K sufficiently large. Hence, to bound the error probability away from zero it will be sufficient to prove that $\mathbb{P}(H_{r^*} = 0) \leq 1/4$.

We will prove this in a series of technical lemmas in Appendix B.2:

1. In Lemma B.4, we will show that

   $$\mathbb{P}(H_{r^*} = 0) \leq \mathbb{P}(G_{r^*} = 0) + \mathbb{E}(1-p)^{M_{r^*}}.$$

   We deal with the two terms separately.

2. In Lemma B.5, we will show that the first term is bounded by

   $$\mathbb{P}(G_{r^*} = 0) \leq \left(1 - (1-p)^{m^*_{r^*}}\right)^{N-K} + \exp\left(-\frac{\delta^2 T \psi_{r^*}}{2}\right), \tag{37}$$

   where $m^*_\ell = T\psi_\ell(1+\delta)$.

3. In Lemma B.6, we show that the second term is bounded by

   $$\mathbb{E}(1-p)^{M_{r^*}} \leq \exp(-p\psi_{r^*} d_{r^*} T), \tag{38}$$

   where $d_\ell = (1-p)^{-K_\ell} - 1$.

Recall that we consider $T = (1-\delta)T_{r^*} = (1-\delta)K \ln K_{r^*}/(\nu\psi_{r^*})$, for the maximising r^*. Since $p\psi_{r^*}T = (1-\delta)\ln K^*_r$, we know that $(1-p)^{m_{r^*}} \simeq K^{-(1-\delta^2)}$, so that both terms in (37) tend to zero. Similarly, (38) also tends to zero for this choice of ℓ and T. This completes the proof. □

8 Discussion

In this paper, we have considered the tropical group testing model of Wang *et al.* [20] in a small-error setting. We have described small-error algorithms in Section

3. We demonstrated the empirical performance of these algorithms in Section 4, showing that tropical DD and tropical SCOMP outperform their classical counterparts. We performed theoretical analysis of the tropical COMP algorithm in Section 5 and of the DD algorithm in Sections 6 and 7, proving that in certain parameter regimes the tropical DD algorithm is asymptotically optimal.

We briefly mention some open problems. Further work could explore test designs with near-constant column weights in the tropical setting, as these designs show a gain in performance in the classical case (see [4]), and Figure 6 suggests the same may well be true here. The results could be made more practically valuable by developing bounds in a noisy setting, under a variety of noise models similar to those described in [4, Chapter 4]. Also, there is potential to extend the results in this paper by considering models with random defectivity levels, as illustrated in Figure 5. It may also be mathematically interesting to develop small-error algorithms and bounds using the delay matrix approach of [20].

Acknowledgements

This work was carried out while Vivekanand Paligadu was on a University of Bristol School of Mathematics undergraduate summer bursary placement, funded by Mark Williams Alumni Funds.

References

[1] M. P. Aldridge. The capacity of Bernoulli nonadaptive group testing. *IEEE Transactions on Information Theory*, 63(11):7142–7148, 2017.

[2] M. P. Aldridge, L. Baldassini, and O. T. Johnson. Group testing algorithms: bounds and simulations. *IEEE Transactions on Information Theory*, 60(6):3671–3687, 2014.

[3] M. P. Aldridge and D. Ellis. Pooled testing and its applications in the COVID-19 pandemic. In M. d. C. Boado-Penas, J. Eisenberg, and Ş. Şahin, editors, *Pandemics: Insurance and Social Protection*, pages 217–249. Springer International Publishing, 2022.

[4] M. P. Aldridge, O. T. Johnson, and J. M. Scarlett. Group testing: an information theory perspective. *Foundations and Trends in Information Theory*, 15(3–4):196–392, 2019.

[5] L. Baldassini, O. T. Johnson, and M. P. Aldridge. The capacity of adaptive group testing. In *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul Turkey, July 2013*, pages 2676–2680, 2013.

[6] H. S. Bharadwaja and C. R. Murthy. Recovery algorithms for pooled RT-qPCR based Covid-19 screening. *IEEE Transactions on Signal Processing*, 70:4353–4368, 2022.

[7] C. L. Chan, P. H. Che, S. Jaggi, and V. Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing*, pages 1832 –1839, Sept. 2011.

[8] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri. Non-adaptive group testing: Explicit bounds and novel algorithms. *IEEE Transactions on Information Theory*, 60(5):3019–3035, 2014.

[9] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick. Information-theoretic and algorithmic thresholds for group testing. *IEEE Transactions on Information Theory*, 66(12):7911–7928, 2020.

[10] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick. Optimal group testing. In J. Abernethy and S. Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1374–1388. PMLR, 2020.

[11] R. Dorfman. The detection of defective members of large populations. *The Annals of Mathematical Statistics*, pages 436–440, 1943.

[12] D. Du and F. Hwang. *Combinatorial Group Testing and Its Applications*. Series on Applied Mathematics. World Scientific, 1993.

[13] Y. Erlich, A. Gilbert, H. Ngo, A. Rudra, N. Thierry-Mieg, M. Wootters, D. Zielinski, and O. Zuk. Biological screens from linear codes: theory and tools. *bioRxiv*, page 035352, 2015.

[14] S. Ghosh, R. Agarwal, M. A. Rehan, S. Pathak, P. Agarwal, Y. Gupta, S. Consul, N. Gupta, R. Goenka, A. Rajwade, et al. A compressed sensing approach to pooled RT-PCR testing for COVID-19 detection. *IEEE Open Journal of Signal Processing*, 2:248–264, 2021.

[15] S. Ghosh, A. Rajwade, S. Krishna, N. Gopalkrishnan, T. E. Schaus, A. Chakravarthy, S. Varahan, V. Appu, R. Ramakrishnan, S. Ch, M. Jindal, V. Bhupathi, A. Gupta, A. Jain, R. Agarwal, S. Pathak, M. A. Rehan,

S. Consul, Y. Gupta, N. Gupta, P. Agarwal, R. Goyal, V. Sagar, U. Ramakrishnan, S. Krishna, P. Yin, D. Palakodeti, and M. Gopalkrishnan. Tapestry: A single-round smart pooling technique for COVID-19 testing. *medRxiv*, 2020. doi:10.1101/2020.04.23.20077727.

[16] A. Heidarzadeh and K. Narayanan. Two-stage adaptive pooling with RT-QPCR for COVID-19 screening. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, pages 8148–8152, 2021.

[17] O. T. Johnson, M. P. Aldridge, and J. Scarlett. Performance of group testing algorithms with near-constant tests-per-item. *IEEE Transactions on Information Theory*, 65(2):707–723, 2019.

[18] W. H. Kautz and R. C. Singleton. Nonrandom binary superimposed codes. *IEEE Transactions on Information Theory*, 10(4):363–377, 1964.

[19] S. N. Rao, D. Manissero, V. R. Steele, and J. Pareja. A narrative systematic review of the clinical utility of cycle threshold values in the context of COVID-19. *Infectious Diseases and Therapy*, 9:573–586, 2020.

[20] H.-P. Wang, R. Gabrys, and A. Vardy. Tropical group testing. *IEEE Transactions on Information Theory*, 69(9):6098–6120, 2023.

[21] T. J. White, N. Arnheim, and H. A. Erlich. The polymerase chain reaction. *Trends in Genetics*, 5:185–189, 1989.

[22] I. Yelin, N. Aharony, E. S. Tamar, A. Argoetti, E. Messer, D. Berenbaum, E. Shafran, A. Kuzli, N. Gandali, O. Shkedi, et al. Evaluation of COVID-19 RT-qPCR test in multi sample pools. *Clinical Infectious Diseases*, 71(16):2073–2078, 2020.

A Proof of the counting bound

We now prove Theorem 3.2, using an argument closely adapted from [5, Proof of Theorem 3.1].

Proof of Theorem 3.2. Given N, $\mathcal{D}$ and $\boldsymbol{K} = (K_1, \dots, K_d, K_\infty)$, we write

$$\Sigma_{N,\boldsymbol{K}} = \big\{ \boldsymbol{u} \in \mathcal{D}^N : |\{j : u_j = r\}| = K_r \text{ for all } r \in \mathcal{D} \big\}$$

for the collection of vectors with the correct number of each type of component. Note that $|\Sigma_{N,\boldsymbol{K}}| = \binom{N}{\boldsymbol{K}}$ is precisely the multinomial term arising in Theorem 3.2. Further, we write $\mathcal{U}$ for the true defectivity vector.

The testing procedure naturally defines a mapping $\theta : \Sigma_{N,\boldsymbol{K}} \to \mathcal{D}^T$. That is, given a putative defectivity vector $\boldsymbol{u} \in \Sigma_{N,\boldsymbol{K}}$, write $\theta(\boldsymbol{u})$ to be the vector of test outcomes. For each vector $\boldsymbol{y} \in \mathcal{D}^T$, write $\mathcal{A}_{\boldsymbol{y}} \subseteq \Sigma_{N,\boldsymbol{K}}$ for the inverse image of $\boldsymbol{y}$ under θ,

$$\mathcal{A}_{\boldsymbol{y}} = \theta^{-1}(\boldsymbol{y}) = \{\boldsymbol{u} \in \Sigma_{N,\boldsymbol{K}} : \theta(\boldsymbol{u}) = \boldsymbol{y}\}\,.$$

As described in [5], a decoding algorithm aims to mimic the inverse image map θ^{-1}. Exactly as in [5], if size $|\mathcal{A}_{\boldsymbol{y}}| \geq 1$ we cannot do better than to pick uniformly among $\mathcal{A}_{\boldsymbol{y}}$, with success probability $1/|\mathcal{A}_{\boldsymbol{y}}|$. (We can ignore $\boldsymbol{y}$ with size $|\mathcal{A}_{\boldsymbol{y}}| = 0$ because these are outcomes which do not occur for any possible defectivity vector).

We can find the success probability by conditioning over all the equiprobable values of the defective set:

$$\begin{aligned}
\mathbb{P}(\mathrm{suc}) &= \sum_{u \in \Sigma_{N,\boldsymbol{K}}} \frac{1}{\binom{N}{\boldsymbol{K}}} \mathbb{P}\left(\mathrm{suc} \mid \mathcal{U} = \boldsymbol{u}\right) \\
&= \frac{1}{\binom{N}{\boldsymbol{K}}} \sum_{\boldsymbol{u} \in \Sigma_{N,\boldsymbol{K}}} \left(\sum_{\boldsymbol{y} \in \mathcal{D}^T} \mathbb{I}(\theta(\boldsymbol{u}) = \boldsymbol{y}) \right) \mathbb{P}\left(\mathrm{suc} \mid \mathcal{U} = \boldsymbol{u}\right) \\
&\leq \frac{1}{\binom{N}{\boldsymbol{K}}} \sum_{\boldsymbol{u} \in \Sigma_{N,\boldsymbol{K}}} \sum_{\boldsymbol{y} \in \mathcal{D}^T : |\mathcal{A}_{\boldsymbol{y}}| \geq 1} \mathbb{I}(\theta(\boldsymbol{u}) = \boldsymbol{y}) \frac{1}{|\mathcal{A}_{\boldsymbol{y}}|} \\
&= \frac{1}{\binom{N}{\boldsymbol{K}}} \sum_{\boldsymbol{y} \in \mathcal{D}^T : |\mathcal{A}_{\boldsymbol{y}}| \geq 1} \frac{1}{|\mathcal{A}_{\boldsymbol{y}}|} \left(\sum_{\boldsymbol{u} \in \Sigma_{N,\boldsymbol{K}}} \mathbb{I}(\theta(\boldsymbol{u}) = \boldsymbol{y}) \right) \\
&= \frac{1}{\binom{N}{\boldsymbol{K}}} \sum_{\boldsymbol{y} \in \mathcal{D}^T : |\mathcal{A}_{\boldsymbol{y}}| \geq 1} \frac{1}{|\mathcal{A}_{\boldsymbol{y}}|} |\mathcal{A}_{\boldsymbol{y}}| \\
&= \frac{|\{\boldsymbol{y} \in \mathcal{D}^T : |\mathcal{A}_{\boldsymbol{y}}| \geq 1\}|}{\binom{N}{\boldsymbol{K}}} \\
&\leq \frac{(d+1)^T}{\binom{N}{\boldsymbol{K}}},
\end{aligned}$$

since $\mathcal{D}^T$ is a set of size $(d+1)^T$. □

B Lemmas for converse results

B.1 Properties of ϕ_K

We seek to prove Lemma 7.3, which gives an upper bound on ϕ_K and hence a lower bound on $\mathbb{P}(\mathrm{err})$. We first need two technical results.

First, recall from [2, Lemma 32] that $\phi_K(q, T)$ is increasing in q for fixed K and T. We can provide a similar analysis to show that:

Lemma B.1. *For fixed T and q, the function $\phi_K(q, T)$ is decreasing in K.*

Proof. We use Pascal's identity $\binom{K+1}{j} = \binom{K}{j} + \binom{K}{j-1}$ to expand

$$\begin{align*}
\phi_{K+1}(q,T) - \phi_K(q,T) &= \sum_{j=0}^{K+1} (-1)^j \left(\binom{K}{j} + \binom{K}{j-1} - \binom{K}{j} \right) (1-jq)^T \\
&= \sum_{j=1}^{K+1} (-1)^j \binom{K}{j-1} (1-jq)^T \tag{39} \\
&= \sum_{\ell=0}^{K} (-1)^j \binom{K}{\ell} (1-(\ell+1)q)^T \\
&= -(1-q)^T \sum_{\ell=0}^{K} (-1)^\ell \binom{K}{\ell} \left(1 - \ell \frac{q}{1-q} \right)^T \\
&= -(1-q)^T \phi_K(q/(1-q), T) \leq 0,
\end{align*}$$

using the fact (as in [2]) that $(1-(\ell+1)q) = (1-q)(1-\ell q/(1-q))$. □

Note that an alternate proof of this is to compare (39) with terms in the proof of [2, Lemma 32] to deduce that

$$\phi_{K+1}(q,T) - \phi_K(q,T) = -\frac{1}{(K+1)(T+1)} \frac{\partial}{\partial q} \phi_{K+1}(q, T+1) \leq 0.$$

The second technical result is the following:

Lemma B.2. *The function $\phi_K(q, T)$ satisfies*

$$\phi_K(q, T+1) \geq \phi_K(q,T) \left(1 + Kq(1-q)^T \right). \tag{40}$$

Proof. As in [2, Proof of Lemma 32], since $j\binom{K}{j} = K\binom{K-1}{j-1}$ we know that

$$\begin{aligned}
&\phi_K(q, T+1) - \phi_K(q, T) \\
&= -\sum_{j=0}^{K} (-1)^j \binom{K}{j} (1-jq)^T jq \\
&= -Kq \sum_{j=1}^{K} (-1)^j \binom{K-1}{j-1} (1-jq)^T \\
&= Kq(1-q)^T \sum_{j=1}^{K} (-1)^{j-1} \binom{K-1}{j-1} \left(1 - (j-1)\frac{q}{1-q}\right)^T \\
&= Kq(1-q)^T \phi_{K-1}\left(\frac{q}{1-q}, T\right),
\end{aligned}$$

since $1 - jq = (1-q)(1-(j-1)q/(1-q))$. Now, using the fact that $\phi_K(q, T)$ is increasing in q and decreasing in K, we know that

$$\phi_{K-1}\left(\frac{q}{1-q}, T\right) \geq \phi_K\left(\frac{q}{1-q}, T\right) \geq \phi_K(q, T),$$

and the result follows. □

We can now prove the bound on ϕ_K.

Proof of Lemma 7.3. Using Lemma B.2 on the terms of a collapsing product and the fact that $\phi_K(q, S) \leq 1$ for all S, we can consider $S \to \infty$ to write

$$\begin{aligned}
\phi_K(q, T) &= \phi_K(q, S+1) \prod_{t=T}^{S} \frac{\phi_K(q, t)}{\phi_K(q, t+1)} \\
&\leq \phi_K(q, S+1) \exp\left(-\sum_{\ell=T}^{S} \log(1 + Kq(1-q)^\ell)\right) \\
&\leq \exp\left(-\sum_{\ell=T}^{\infty} \frac{Kq(1-q)^\ell}{1 + Kq(1-q)^\ell}\right), \qquad (41) \\
&\leq \exp\left(-\int_T^\infty \frac{Kq(1-q)^\ell}{1 + Kq(1-q)^\ell} d\ell\right), \qquad (42) \\
&= \exp\left(-\frac{\log(1 + Kq(1-q)^T)}{-\log(1-q)}\right) \qquad (43)
\end{aligned}$$

where a) (41) follows using the fact that $-\log(1+x) \leq -x/(1+x)$, b) (42) follows since $f(x) := Kq(1-q)^x/(1 + Kq(1-q)^x)$ is decreasing in x so we can bound $\sum_{\ell=T}^{\infty} f(\ell) \geq \int_T^\infty f(\ell) d\ell$, c) (32) follows using the fact that (in the numerator) $\log(1+x) \geq x/(1+x)$ and (in the denominator) $-\log(1-q) \leq q/(1-q)$. □

B.2 Lemmas for the algorithm-independent converse

This section consists of various lemmas used in the proof of Theorem 7.4.

Lemma B.3. *For any ℓ, the conditional distribution of H_ℓ is given by*

$$H_\ell \mid \{M_\ell = m_\ell, G_\ell = g_\ell\} \sim \mathrm{Bin}(g_\ell, 1-(1-p)^{m_\ell}).$$

Proof. Suppose one has already examined the tests with outcomes $\infty, d, \ldots, \ell+1$. There remain $G_\ell = g_\ell$ non-defective items which have not appeared in any of these tests. They will each independently contribute to H_ℓ unless they avoid appearing in each of the m_ℓ tests with outcome ℓ. Because of the Bernoulli design of the matrix, this will happen with probability $1-(1-p)^{m_\ell}$. □

Lemma B.4. *The probability we have no non-defectives in* $\mathrm{PD}(\ell)$ *is bounded above by*

$$\mathbb{P}(H_{r^*} = 0) \leq \mathbb{P}(G_{r^*} = 0) + \mathbb{E}(1-p)^{M_{r^*}}.$$

Proof. We know from Lemma B.3 that $H_\ell \mid \{M_\ell = m_\ell, G_\ell = g_\ell\} \sim \mathrm{Bin}(g_\ell, 1-(1-p)^{m_\ell})$. Since $\mathbb{P}(\mathrm{Bin}(g, 1-Q) = 0) = (1-Q)^g$ we know that for any g_ℓ

$$\begin{aligned}
\mathbb{P}(H_\ell = 0) &= \sum_{m_\ell, g_\ell} \mathbb{P}(H_\ell = 0 \mid M_\ell = m_\ell, G_\ell = g_\ell)\, \mathbb{P}(M_\ell = m_\ell, G_\ell = g_\ell) \\
&= \sum_{m_\ell, g_\ell} (1-p)^{m_\ell g_\ell}\, \mathbb{P}(M_\ell = \ell, G_\ell = g_\ell) \\
&= \mathbb{E}(1-p)^{M_\ell G_\ell} \qquad (44) \\
&\leq \mathbb{P}(G_\ell = 0) + \mathbb{E}(1-p)^{M_\ell}. \qquad (45)
\end{aligned}$$

Here, we got (45) by considering separately the case where $G_\ell = 0$, and by noting that if $G_\ell \geq 1$ then $(1-p)^{M_\ell G_\ell} \leq (1-p)^{M_\ell}$. □

Lemma B.5. *Writing $m_\ell^* = T\psi_\ell(1+\delta)$ we can bound*

$$\mathbb{P}(G_\ell = 0) \leq \left(1-(1-p)^{m_\ell^*}\right)^{N-K} + \exp\left(-\frac{\delta^2 T \psi_\ell}{2}\right).$$

Proof. Using Lemma 6.8 we know $\mathbb{P}(G_\ell = 0 \mid \overline{M}_\ell = m) = (1-(1-p)^m)^{N-K}$, which is increasing in m. Therefore we can write

$$\begin{aligned}
\mathbb{P}(G_\ell = 0) &= \mathbb{P}(G_\ell = 0 \mid \overline{M}_\ell \leq m_\ell^*)\, \mathbb{P}(\overline{M}_\ell = m_\ell^*) \\
&\qquad + \mathbb{P}(G_\ell = 0 \mid \overline{M}_\ell > m_\ell^*)\, \mathbb{P}(\overline{M}_\ell > m_\ell^*) \\
&\leq \mathbb{P}(G_\ell = 0 \mid \overline{M}_\ell \leq m_\ell^*) + \mathbb{P}(\overline{M}_\ell > m_\ell^*) \\
&\leq \mathbb{P}(G_\ell = 0 \mid \overline{M}_\ell = m_\ell^*) + \mathbb{P}(\overline{M}_\ell > m_\ell^*) \\
&\leq \left(1-(1-p)^{m_\ell^*}\right)^{N-K} + \mathbb{P}(\overline{M}_\ell > m_\ell^*). \qquad (46)
\end{aligned}$$

Recall that $\overline{M}_\ell \sim \mathrm{Bin}(T, \psi_\ell)$, so $m_\ell^* = T\psi_\ell(1+\delta)$ is slightly bigger than the expectation, and we can bound the second term by Chernoff's inequality. □

Lemma B.6. *We have the bound*

$$\mathbb{E}(1-p)^{M_{r^*}} \leq \exp(-p\psi_{r^*} d_{r^*} T),$$

where $d_\ell = (1-p)^{-K_\ell} - 1$.

Proof. From Lemma 6.5 we know that $M_\ell \sim \mathrm{Bin}(T, \psi_\ell d_\ell)$. Hence, using the binomial theorem:

$$\begin{aligned}\mathbb{E}(1-p)^{M_\ell} &= \sum_{m=0}^{T} \binom{T}{m} (\psi_\ell d_\ell)^m (1-\psi_\ell d_\ell)^{T-m} (1-p)^m \\ &= (1 - p\psi_\ell d_\ell)^T \\ &\leq \exp(-p\psi_\ell d_\ell T).\end{aligned}$$

□