Improving Group Testing via Gradient Descent

Sundara Rajan Srinivasavaradhan*, Pavlos Nikolopoulos†, Christina Fragouli*, Suhas Diggavi*
*University of California, Los Angeles, Electrical and Computer Engineering,
{sundar, christina.fragouli, suhasdiggavi}@ucla.edu †EPFL, Switzerland, pavlos.nikolopoulos@epfl.ch

***Abstract*—We study the problem of group testing with non-identical, independent priors. So far, the pooling strategies that have been proposed in the literature take the following approach: a hand-crafted test design along with a decoding strategy is proposed, and guarantees are provided on how many tests are sufficient in order to identify all infections in a population. In this paper, we take a different, yet perhaps more practical, approach: we fix the decoder and the number of tests, and we ask, given these, what is the *best* test design one could use? We explore this question for the Definite Non-Defectives (DND) decoder. We formulate a (non-convex) optimization problem, where the objective function is the expected number of errors for a particular design. We find approximate solutions via gradient descent, which we further optimize with informed initialization. We illustrate through simulations that our method can achieve significant performance improvement over traditional approaches.**

I. Introduction

Group testing has recently attracted significant attention in the context of COVID [1]–[6], and several countries (including India, Germany, US, and China) have already deployed preliminary group-testing strategies [7], [8].

Group testing has a rich history in academia and a number of variations and setups have been examined so far [9]–[12]. Simply stated, group testing assumes a population of N individuals out of which some are infected, and the goal is to design testing strategies and corresponding decoding algorithms to identify the infections from the test results. Most works revolve around proposing a particular hand-crafted test design (e.g., random Bernoulli design) coupled with a decoding strategy (e.g. Definite Defectives, Definite Non-Defectives), and guarantees are provided on the number of tests required to achieve vanishing probability of error. Additionally, order-optimality results have been proved for the asymptotic regime, where the population size tends to infinity.

This paper examines instead the following complementary question: Given a fixed decoding strategy and a given number of tests T (perhaps smaller than what is needed to achieve zero error), what is the *best* test design one may use? [1] We examine this question in the context of nonadaptive group testing, and under the assumption of a Definite Non-Defectives (DND) decoder, which eliminates false negatives by construction.

Our main contribution in this paper is to derive a novel to way to approach group test matrix design: we reduce the matrix design problem to a non-convex continuous-time optimization problem that can be solved using gradient descent methods. We believe this approach is interesting, because it provides an easy way to construct test matrices tailored to particular scenarios, such as specific number of tests, known prior probabilities of infection, and/or other parameters.

More specifically, our problem requires finding a test-design matrix G that minimizes the expected number of erroneous identifications (i.e. false positives). This, however, presents two challenges: (a) the analytical computation of the expected number of false positives turns out to be computationally difficult; and (b) because $G \in \{0,1\}^{T\times N}$, we are faced with a combinatorial optimization problem.

To address these challenges, we proceed as follows: First, we provide a lower bound on the expected number of errors, which we use as a proxy in the optimization problem; that bound can be computed in $O(N^2)$ runtime. We then relax the combinatorial optimization problem based on an equivalence result; the objective function in that relaxed formulation as well as its gradient can be computed in $O(N^2)$, thus enabling the use of Gradient Descent (GD). To further improve the performance of our method, we propose two approaches to GD: (i) an informed initialization with information from classic test designs, such as the Constant Column Weight (CCW) design and the Coupon Collector Algorithm (CCA); (ii) a stochastic re-initialization of the state of the solution every few gradient iterations (e.g. 100 iterations), in a way that allows GD to explore across various neighborhoods, while also ensuring that the objective value does not increase by much with each re-initialization.

Numerical evaluations show that the GD based approaches can significantly outperform classical test designs, achieving up to 58% fewer errors with the DND decoder on simulated infection models. Rather surprisingly, GD based designs also significantly outperform classical test designs when the decoder is switched to definite defectives (DD), indicating transferability to other decoders.

We underline that our contribution is not the specific matrices design, but the novel approach to design them. As we discuss in the conclusion section, given that this is a first paper proposing this approach, we hope that it can be extended beyond the cases we already examine in this paper - to other decoders, other group test models, even to the design of the group test decoders themselves.

II. Related work

We here give a brief overview of group testing; the exact problem we consider is detailed in Section III-A.

Three infection models are usually studied in the group testing literature: (i) in the *combinatorial priors model*, a fixed

[1] Interestingly, a discussion of one of the authors with the General Secretary of Public Health in an EU state has revealed that this question is perhaps the most relevant in practice, as both private and public lab facilities have limited testing capacity per day, and what actually matters is how to use the available tests most efficiently.

number of individuals k (selected uniformly at random), are infected; (ii) in *i.i.d probabilistic priors model*, each individual is i.i.d infected with some probability p; (iii) in the *non-identical probabilistic priors model*, each item i is infected independently of all others with prior probability p_i, so that the expected number of infected members is $\bar{k} = \sum_{i=1}^{N} p_i$. Infection models (i) and (ii) have received attention from researchers for the most part (see for example, [13]–[23]). Infection model (iii) is the least studied one [24]; we refer the reader to [10] for an excellent summary of existing work on the above infection models. Tangentially, recent works have considered correlated infection models; see, for example, [25]–[31].

Typically, the goal of this line of research is dual: (a) compute lower bounds for the number of tests T that is needed to identify the infection statuses of all individuals; and (b), provide test designs that can achieve zero- or small-error recovery by also asymptotically matching the lower bounds or at least being order-optimal in particular asymptotic regimes for k (or $\bar{k}$). Perhaps, the most representative (non-adaptive) designs, which we also use as baselines in our numerical evaluation, are the well-known Constant column weight (CCW) design [17], [32], and the Coupon Collector Algorithm (CCA) [24]. CCW has been proved to be order-optimal[2] in model (ii) and in a sparse regime, where $\bar{k} = \Theta(N^{\alpha})$ and $\alpha \in [0, 0.409]$ [17], [20], [23]. CCA has been shown to achieve a vanishing error probability in infection model (iii) of non-identical priors, given that the number of tests used satisfies $T = \omega\left(4e \ln N \sum_{i=1}^{N} p_i\right)$ [15], [24].

Our work is closely related to these test designs, but takes a different approach: instead of focusing on asymptotic regimes, we consider a fixed number of tests, which may also be lower than the lower bound for exact recovery and we are interested in minimizing the overall error rate via gradient descent. We examine this in the context of infection model (iii) with non-identical probabilistic priors, which accepts (ii) as a special case and in our opinion resembles better the information available in practice – the prevalence/prior infection probability varies across individuals depending on many factors such as age, geographical location, health condition, contact with other infected individuals, etc.

III. Preliminaries

In this section, we first precisely formulate the problem of interest, and then state a simple lemma on combinatorial optimization that is used in our work.

A. Problem formulation

We consider the noiseless nonadaptive group testing problem with non-identical priors. There are N individuals in the population, where individual i is infected independently with probability p_i. We assume that the value of p_i is known a priori[3]. Let U_i be the infection status of individual i: $U_i = \mathbb{1}\{\text{Individual } i \text{ is infected}\}$. As a result, $U_i \sim \text{Ber}(p_i)$. We will denote by $\mathbf{U} = (U_1, U_2, ..., U_N)$ the vector of infection statuses.

Testing matrix: A testing matrix $G \in \{0,1\}^{T \times N}$ is a $T \times N$ binary matrix. Row t in the testing matrix represents the individuals participating in test t, i.e., $G_{ti} = 1$ represents individual i participating in test t. The test results corresponding to a particular realization of $\mathbf{U} = (U_1, U_2, ..., U_N)$ and G are defined as the vector $\mathbf{Y} = (Y_1, Y_2, ..., Y_T)$ where

$$Y_t = 1 - \prod_{i=1}^{N} (1 - G_{ti} U_i). \tag{1}$$

In words, the test t gives a positive result if any of the individuals participating in the test are infected, otherwise it gives a negative result[4]. In (1) $Y_t = 1$ if and only if there exists i such that both $G_{ti} = 1$ and $U_i = 1$ (individual i is infected). In order to infer $\mathbf{U}$ from Y, a *decoding algorithm* $r : \{0,1\}^T \to \{0,1\}^N$ constructs an estimate $\widehat{\mathbf{U}}$ of the infection statuses from the test results. In this work, we fix the decoding algorithm, which we describe next.

DND decoder: The definite non-defective (DND) decoder is a well-known decoding algorithm that forms an estimate of $\mathbf{U}$ by identifying those individuals who have participated in at least one negative test as healthy and labeling every other individual as infected – i.e., it operates under the principle "every item is defective unless proved otherwise". More precisely, it outputs an estimate $\widehat{\mathbf{U}}$ where

$$\widehat{U}_i = \prod_{t=1}^{T} Y_t^{G_{ti}}. \tag{2}$$

$\widehat{\mathbf{U}}$ has zero false negatives by construction – it can be seen that $\widehat{U}_i = 1$ whenever $U_i = 1$. The number of errors (false positives) that the DND decoder makes for a particular realization $\mathbf{U}$ is given by

$$\sum_{i=1}^{N} \mathbb{1}\{\widehat{U}_i \neq U_i\} = \sum_{i=1}^{N} \mathbb{1}\{U_i = 0\} \mathbb{1}\{\widehat{U}_i = 1 | U_i = 0\},$$

and as a result the expected number of errors $\mathcal{E}(G)$ under the DND decoder for a given G is

$$\begin{aligned} \mathcal{E}(G) &\triangleq \mathbb{E}\left[\sum_{i=1}^{N} \mathbb{1}\{\widehat{U}_i \neq U_i\}\right] \\ &= \sum_{i=1}^{N} \Pr(U_i = 0) \Pr\left(\widehat{U}_i = 1 | U_i = 0\right) \\ &= \sum_{i=1}^{N} (1 - p_i) \mathbb{E}\left[\widehat{U}_i | U_i = 0\right]. \end{aligned} \tag{3}$$

Further, when U_i is fixed to be 0, $\widehat{U}_i$ is a function of G and $\mathbf{U} \setminus \{i\}$, where $\mathbf{U} \setminus \{i\} \triangleq (U_1, ..., U_{i-1}, U_{i+1}, ..., U_N)$ denotes

[2] In fact, in the same regime, [20] has provided the precise constants for optimal non-adaptive group testing.

[3] This is a standard assumption in group testing. Otherwise, epidemiological models for disease spread can be used to estimate these probabilities ([33]–[35]).

[4] Most works in group testing express the right-hand side of (1) as a Boolean expression. However, we use this particular form (similar expression was given in [21]) as it easily admits continuous-valued relaxations of the composing variables.

the vector $\mathbf{U}$ without its i^{th} entry. Thus, fixing $U_i = 0$, and using (1) and (2) we have,

$$\widehat{U}_i = \prod_{t=1}^{T} \left(1 - \prod_{\substack{j=1:\\ j\neq i}}^{N} (1 - G_{tj}U_j)\right)^{G_{ti}} \overset{(a)}{=} \prod_{t=1}^{T} \left(1 - G_{ti} \prod_{\substack{j=1:\\ j\neq i}}^{N} (1 - G_{tj}U_j)\right),$$

where (a) follows because of the following fact: $(1-x)^y = 1 - xy$ if $y \in \{0,1\}$. Now, denoting $\gamma_{t,i} \triangleq \left(1 - G_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1 - G_{tj}U_j)\right)$ in the above expression, we rewrite (3) as:

$$\mathcal{E}(G) = \sum_{i=1}^{N} (1-p_i)\mathbb{E}_{\mathbf{U}\setminus\{i\}} \prod_{t=1}^{T} \gamma_{t,i}. \tag{4}$$

Our Goal: We want to minimize $\mathcal{E}(G)$ across all binary matrices G of size $T \times N$, i.e., solve

$$G_{opt} = \underset{G\in\{0,1\}^{T\times N}}{\arg\min}\ \mathcal{E}(G). \tag{5}$$

Discussion: We first observe that $\gamma_{t,i}$ is not independent of $\gamma_{t',i}$ for $t \neq t'$ as they potentially share common U_j terms. As a result, the expectation of the product term in (4) is not trivially the product of expectations, which makes the computation of $\mathcal{E}(G)$ intractable in general (indeed one could estimate $\mathcal{E}(G)$ using Monte-Carlo methods, belief propagation etc., but this is, in general, computationally expensive and hence possible only for small values of T and N, which are not realistic in our context.). In Section IV we provide a lower bound for $\mathcal{E}(G)$ which can be computed efficiently, and which we use as a proxy for $\mathcal{E}(G)$. Our numerics show that the lower bound is a very good approximation to $\mathcal{E}(G)$.

We also note that in principle, (5) could be formulated for any decoder, not just the DND decoder. However, the particular nature of $\mathcal{E}(G)$ for the DND decoder admits a nice form, for which we can propose an approximate solution using lower bounding techniques (Section IV).

Finally, note that (4) shows something which is also intuitively correct: the lower the T (for fixed N) or the larger the N (for fixed T), the larger the expected number of errors $\mathcal{E}(G)$.

B. A combinatorial relaxation result

We now take a detour to prove a simple result that allows one to relax combinatorial optimization problems that aim to optimize over the vertices of an n-dimensional hypercube. One could extend this technique for optimization over other finite sets as well.

Lemma 1. *In order to solve*

$$\underset{\mathbf{x}\in\{0,1\}^n}{\arg\min}\ g(\mathbf{x}), \tag{6}$$

it is sufficient to solve

$$\underset{\mathbf{q}\in[0,1]^n}{\arg\min}\ f(\mathbf{q}), \tag{7}$$

$$\textit{where } f(\mathbf{q}) \triangleq \mathbb{E}_{\mathbf{X}\sim Ber(\mathbf{q})}\ g(\mathbf{X})$$

can be envisioned as a continuous extension of $g(\mathbf{x})$. The expectation in the above expression is taken w.r.t the distribution where each $X_i \sim Ber(q_i)$, and the X_is are independent of each other.

We refer the reader to Appendix A for the proof but provide an intuition next. First, we observe that (6) aims to minimize a function g over the vertices of the n-dimensional unit hypercube. Instead, what Lemma 1 says is that one could minimize a different continuous-valued function f over the entire n-dimensional unit hypercube and obtain a solution to the discrete optimization problem. To do this, it is sufficient to ensure that f satisfies the following properties:

1) The value taken by f in the hypercube is always greater or equal to the minimum value taken by g.
2) The value taken by f at each vertex of the hypercube is equal to the value taken by g.
3) If f attains minima at $\mathbf{q}^*$, one can easily obtain a corresponding minima $\mathbf{x}^*$ of g. This point is just for algorithmic convenience as we essentially would like to easily transform a minima of f to a minima of g.

If the above 3 properties are satisfied by a function f, it is clear that one could minimize f in the hypercube and obtain a solution to the combinatorial optimization problem (6).

One choice of f that satisfies the above conditions is the following: at each point $\mathbf{q}$ in the interior of the hypercube, represent $f(\mathbf{q})$ as a convex combination of the values taken by g, i.e.,

$$f(\mathbf{q}) = \sum_{\mathbf{x}} \alpha_{\mathbf{q},\mathbf{x}}\ g(\mathbf{x}),$$

for some $\alpha_{\mathbf{q},\mathbf{x}}$ that are non-negative and sum to 1. This ensures that property 1 above is satisfied. Additionally, whenever $\mathbf{q}$ is a vertex, we fix $\alpha_{\mathbf{q},\mathbf{q}} = 1$ so that $f(\mathbf{q}) = g(\mathbf{q})$, ensuring property 2. Moreover, for any minimizer $\mathbf{q}^*$ of f, pick any $\mathbf{x}$ such that $\alpha_{\mathbf{q}^*,\mathbf{x}} > 0$ – such an $\mathbf{x}$ minimizes g. This is precisely what Lemma 1 does, where instead of defining f by explicitly picking coefficients $\alpha_{\mathbf{q},\mathbf{x}}$, we just define it via an expectation[5].

Example: To see an example of what f looks like, assume that $n = 2$ in (6). Then, we have

$$f(q_1,q_2) \triangleq q_1q_2g(1,1) + q_1(1-q_2)g(1,0) + (1-q_1)q_2g(0,1) + (1-q_1)(1-q_2)g(0,0).$$

Remark: There is a long history of using relaxation techniques to approximate solutions of combinatorial optimization problems (see [38] for an overview). Most of these focus on linear programming relaxation techniques. In Lemma 1, there is no assumption on $g(\cdot)$ whatsoever and the resulting relaxation may not be a linear program. Moreover, it may not be easy to compute $f(\cdot)$ in all cases and it may also not be easy to compute the gradient $\nabla f(\cdot)$ as well. In cases where exactly

[5]A similar but more restricted version of Lemma 1 was presented in [36], [37] in the context of deletion channels, where we proved the result for a particular example of $g(\cdot)$ that arose in the context of deletion channels.

computing or approximating the gradient is easy (as is indeed the case in this work), one can use first-order optimization techniques such as GD.

IV. Main results

In this section, we delineate our approach to find an approximate solution to (5). Following the discussion at the end of Section III-A, our approach is three-fold: First, we lower bound $\mathcal{E}(G)$ by another function $\mathcal{E}_{LB}(G)$, whose computation turns out to be tractable; we then use $\mathcal{E}_{LB}(G)$ as a proxy for $\mathcal{E}(G)$. Next, we use Lemma 1 to show that it is sufficient to consider a continuous relaxation of the resulting combinatorial optimization problem. Finally, we show that the objective function in the continuous relaxation and its gradient can also be computed efficiently, thus enabling gradient descent.

A. A lower bound for $\mathcal{E}(G)$

As a first step, the following theorem states and proves a lower bound for $\mathcal{E}(G)$.

Theorem 1. *Consider a random vector $\mathbf{U} = (U_1, U_2, ..., U_N)$ where $U_i \sim Ber(p_i)$. For a given testing matrix G, and under the DND decoder, the expected number of errors (see (4)) satisfies*

$$\mathcal{E}(G) \geq \mathcal{E}_{LB}(G),$$

where

$$\mathcal{E}_{LB}(G) \triangleq \sum_{i=1}^{N}(1-p_i)\prod_{t=1}^{T}\left(1 - G_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-G_{tj}p_j)\right).$$

Proof. First we recall the expression for $\mathcal{E}(G)$ in (4):

$$\mathcal{E}(G) = \sum_{i=1}^{N}(1-p_i)\mathbb{E}_{\mathbf{U}\setminus\{i\}}\prod_{t=1}^{T}\gamma_{t,i}.$$

Using the FKG inequality (see [39]–[41] or proof of Lemma 4 in [18]) one could show that

$$\mathbb{E}_{\mathbf{U}\setminus\{i\}}\prod_{t=1}^{T}\gamma_{t,i} \geq \prod_{t=1}^{T}\mathbb{E}_{\mathbf{U}\setminus\{i\}}\gamma_{t,i}.$$

A rigorous proof of the above statement can be found in Appendix B. The idea is to show that $\gamma_{t,i}$ is an increasing function on $\mathbf{U}$ (assuming a partial ordering); using this observation, the result follows as an application of the FKG inequality. Thus, we have

$$\begin{aligned}\mathcal{E}(G) &\geq \sum_{i=1}^{N}(1-p_i)\prod_{t=1}^{T}\mathbb{E}_{\mathbf{U}\setminus\{i\}}\gamma_{t,i}\\ &= \sum_{i=1}^{N}(1-p_i)\prod_{t=1}^{T}\left(1 - G_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-G_{tj}p_j)\right)\\ &= \mathcal{E}_{LB}(G)\end{aligned}$$

□

In all numerical evaluations we performed, $\mathcal{E}(G)$ (as estimated via Monte-Carlo simulations) and the lower bound $\mathcal{E}_{LB}(G)$ were close – we provide example scatter plots in Figure 1 – which indicates that minimizing $\mathcal{E}_{LB}(G)$ is a viable alternative to minimizing $\mathcal{E}(G)$. Recall that computing $\mathcal{E}(G)$ is computationally difficult, which motivates the use of $\mathcal{E}_{LB}(G)$ as its proxy.

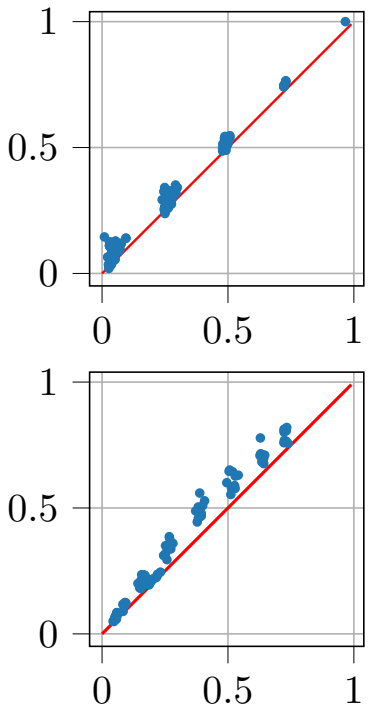


Fig. 1: Scatter plot of $\mathcal{E}(G)$ (on y-axis) vs. $\mathcal{E}_{LB}(G)$ (on x-axis) normalized by the blocklength N. $\mathcal{E}(G)$ is estimated via Monte-Carlo simulations while $\mathcal{E}_{LB}(G)$ is computed exactly. For a fixed prior distribution, we pick a variety of G matrices and plot the two metrics – the left figure plots for every $G \in \{0,1\}^{2\times 4}$ while the right figure plots for 1000 choices of G sampled from $\{0,1\}^{300\times 500}$.

B. A continuous optimization formulation

Given the above discussion, we now propose using $\mathcal{E}_{LB}(G)$ as a proxy for $\mathcal{E}(G)$ – more precisely we propose to solve the following optimization problem:

$$\arg\min_{G\in\{0,1\}^{T\times N}} \mathcal{E}_{LB}(G). \tag{8}$$

We next use Lemma 1 to argue that a continuous relaxation of (8) is equivalent to (8). Before stating the main result, we give a definition: we say that a random matrix $\mathbf{G} \sim Ber(Q)$ (read as "$\mathbf{G}$ is parameterized by the Bernoulli matrix Q") if each $G_{ti} \sim \text{Ber}(Q_{ti})\ \forall\ t, i$ and the G_{ti} variables are independent of each other.

Corollary 1. *Suppose $U_i \sim Ber(p_i)\ \forall\ i$. In order to solve the optimization problem*

$$\arg\min_{G\in\{0,1\}^{T\times N}} \mathcal{E}_{LB}(G), \tag{9}$$

it is sufficient to solve

$$\arg\min_{Q\in[0,1]^{T\times N}} \mathbb{E}_{G\sim Ber(Q)}\mathcal{E}_{LB}(G). \tag{10}$$

This is a direct corollary of Lemma 1, where the objective function is $\mathcal{E}_{LB}(G)$ and we associate a parameter Q_{ti} corresponding to each G_{ti} (see Appendix D for a complete proof).

Thus, we now have the following approximate formulation for which the objective function (and its gradient) can be computed in $O(N^2)$ time complexity (see Section IV-C). The hope is that solving (11) gives sufficiently good choices of $G \sim Ber(Q^*)$; our experimental results in Section VI indicate that this is indeed the case.

Approximate formulation: Solve for

$$Q^* = \underset{Q \in [0,1]^{T \times N}}{\arg\min} \; f(Q), \tag{11}$$

where $f(Q) \triangleq \mathbb{E}_{G \sim Ber(Q)} \mathcal{E}_{LB}(G)$.

Given the above formulation, we can now use techniques such as gradient descent (GD) to select the testing matrix G. In essence, we are searching over the continuous space of distribution matrices Q. If the gradient of $f(Q)$ can be efficiently computed, one could use GD to converge to a local minima Q^* and pick a $G \sim Ber(Q^*)$.

C. Expression for $f(Q)$

We now give a closed-form expression for $f(Q)$ and briefly discuss the computational complexity of computing $f(Q)$ and its gradient; the details are deferred to Appendix C, Appendix E and Appendix F. We have,

$$\begin{aligned} f(Q) &\triangleq \mathbb{E}_{G \sim Ber(Q)} \mathcal{E}_{LB}(G) \\ &= \mathbb{E}_{G \sim Ber(Q)} \sum_{i=1}^{N} (1-p_i) \prod_{t=1}^{T} \left(1 - G_{ti} \prod_{\substack{j=1: \\ j \neq i}}^{N} (1 - G_{tj} p_j) \right) \\ &\overset{(a)}{=} \sum_{i=1}^{N} (1-p_i) \prod_{t=1}^{T} \mathbb{E}_{G \sim Ber(Q)} \left(1 - G_{ti} \prod_{\substack{j=1: \\ j \neq i}}^{N} (1 - G_{tj} p_j) \right) \\ &= \sum_{i=1}^{N} (1-p_i) \prod_{t=1}^{T} \left(1 - Q_{ti} \prod_{\substack{j=1: \\ j \neq i}}^{N} (1 - Q_{tj} p_j) \right), \end{aligned} \tag{12}$$

where in (a) the expectation is pushed inside the product terms as $\mathcal{E}_{LB}(G)$ is linear when viewed as a function of a single G_{ti}.

In Appendix C we discuss an $O(N^2)$ algorithm that simplifies the computation of $f(Q)$ above, under the assumption of $T = O(N)$. Given (12), one could derive an expression for the gradient $\nabla f(Q)$ by calculating each partial derivative $\frac{\partial f(Q)}{\partial Q_{lm}}$. The details of the derivation can be found in Appendix E. Moreover, in Appendix F, we discuss the computation of $\nabla f(Q)$ in $O(N^2)$ runtime.

V. Algorithms

Leveraging the approximate formulation in (11), we here explore a GD approach to find good choices of G. Our proposed approach uses informed initialization with information provided by traditional group test designs. Thus, it can be viewed as a way to refine and improve existing designs via local search. Moreover, we propose a variation of GD that numerically seems to converge to good choices of G in many situations even without informed initialization.

A. Baseline test designs

We use the following two group test design algorithms as baselines for comparison:

- **Constant column weight (CCW) design** (see [17], [32]). This design was introduced in the context of group testing for identical priors[6], but we adapt it to be applicable for non-identical priors as well, in addition to identical priors. Here we construct a randomized G assuming that all individuals have the same prior probability of infection p_{mean} (this assumption is trivially true if the priors are identical), where p_{mean} is defined as the mean prior probability of infection $\frac{1}{N}\sum_{i=1}^{N} p_i$. The testing matrix G is constructed column-by-column by placing each individual in a fixed number ($\frac{0.69T}{Np_{mean}}$) of tests, uniformly at random.
- **Coupon Collector Algorithm (CCA)** from [24]. The CCA algorithm was introduced in [24] for the case of non-identical, independent priors. In short, the CCA algorithm constructs a random non-adaptive test design G by sampling each row independently from a distribution (we refer the reader to [24] for the exact description of this distribution). The idea is to place objects which are less likely to be infected in more number of tests and vice-versa.

B. Test designs based on gradient descent

We are now ready to describe the gradient descent (GD) approaches to search for G. The high-level idea for our algorithms is as follows:

- We consider the approximate formulation in (11). Pick an initial point $Q^{(0)}$.
- At each gradient iteration l, update $Q^{(l)} \leftarrow Q^{(l-1)} - \epsilon \nabla_Q f(Q)$, where ϵ is the step size. Project $Q^{(l)}$ onto $[0,1]^{T \times N}$ by resetting negative entries to 0 and entries greater than 1 to 1.
- Stop based on some stopping criteria (e.g. limit number of gradient steps or check for convergence).
- Let Q^* be the resulting output. Sample a matrix G^* where $G^* \sim Ber(Q^*)$ and return it.

As it turns out, in our experiments, the choice of initialization plays a significant role in finding good choices of G. We propose the following initializations.

- **GD + CCW init.** We first sample a testing matrix according to the CCW testing matrix and set $Q^{(0)}$ as this matrix. The GD proceeds with this initialization.
- **GD + CCA init.** We first sample a testing matrix according to the CCA testing matrix and set $Q^{(0)}$ as this matrix. The GD proceeds with this initialization.

[6]Most of these were proposed in the context of combinatorial priors. However, Theorem 1.7 and Theorem 1.8 from [10] imply that any algorithm that attains a vanishing probability of error on the combinatorial priors, also attains a vanishing probability of error on the corresponding i.i.d probabilistic priors.

Notably, any other state-of-the-art test design could have been used as initialization. In principle, the above approach can be perceived as a way to refine existing test designs via local search. Alternatively, we also propose a modification to the GD approach called **GD + sampling** that helps avoid getting stuck in a local minima by encouraging GD to explore multiple neighborhoods. The idea is use stochastic re-initialization of the solution state every few gradient iterations, while ensuring that the value of the objective function is approximately preserved. First note that the objective value $f(Q)$ is the mean of $f(G)$ with $G \sim Ber(Q)$. Therefore, it is reasonable to expect that typical realizations of G will be such that $f(G)$ is close to $f(Q)$. Given this idea, we propose the following: start from the all 0 initialization. However, every few gradient iterations, we replace the current solution state $Q^{(l)}$ by G^s where G^s is sampled from the distributed matrix $Q^{(l)}$, i.e., $G^s \sim Ber(Q^{(l)})$. This encourages GD to explore different neighborhoods while (approximately) preserving the monotonicity of GD.

VI. Numerical results

In this section, we show simulation results to demonstrate the improvement our GD based approaches provides.

Test designs compared: We compare the testing matrices G obtained via each of the following methods: CCW, CCA, GD + CCW init., GD + CCA init., GD + sampling. For completeness, we consider also the trivial all 0-initialization for GD (which we call **GD + 0 init**), where the initial point $Q^{(0)}$ is set to all zeros.

Set-up: We first fix the prior probabilities of infection $(p_1, p_2, ..., p_N)$ – each p_i is sampled from an exponential distribution with mean 0.05; if $p_i > 1$, we set it to 1. We repeat for 10 such prior distributions. For each design, we estimate $\mathcal{E}(G)$ via Monte-Carlo simulations.

Metrics: We use the false positive (FP) rate (defined as the fraction of uninfected individuals incorrectly determined to be infected) to measure the performance w.r.t the DND decoder. Recall that the DND decoder results in 0 false negatives (FN) by construction.

Transferability to other decoders: As our GD methods aim for optimal designs with the DND decoder, a natural follow-up question is how they perform with other decoders. We compare the performance of each of the test designs w.r.t the Definite Defective (DD) decoder. One could also consider other decoders, such as ones based on belief propagation, but these result in both FP and FN, and consequently the comparison between different methods is not trivial; it requires weighing FP against FN, which can be application specific. We refer the reader to Section 2.4 in [10] for a precise description of DD decoder. Consequently, DD has 0 FP by construction. In this case, we use as performance measure the false negative (FN) rate.

Observations: In Figure 2a, we plot the FP rate for each test design w.r.t DND decoder, as a function of T. We observe that the GD based methods significantly outperform CCW and CCA[7]. Notably, the improvement of our enhanced GD with informed initialization or sampling seems inversely proportional to T, which is of practical importance.

Next, we plot the FN rate of each test design w.r.t the DD decoder, as a function of T in Figure 2b. The performance trend here is similar to what was observed with the DND decoder, which further supports the usefulness of our GD based approach and its transferability to other decoders.

Figures 3a, 3b, 4a and 4a depict our results for two other cases, where the priors are sampled from a discrete bimodal distribution. The performance trend for the GD based approach is similar, while the performance gap from the well-established CCW and CCA algorithms increases. This indicates the usefulness of our method in more realistic scenarios, where priors are expected to be drawn from few distinct values.

VII. Conclusions and open questions

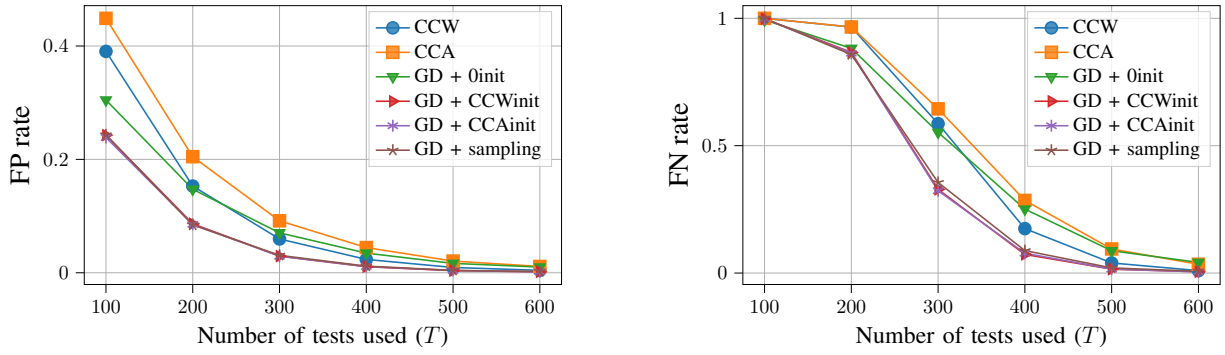
In this work, we formulated the search for good group-test designs, under the assumption of a DND decoder, as a non-convex optimization problem, and we proposed a solution via enhanced gradient descent. Our solution is approximate in the sense that it minimizes a lower bound on the expected number of identification errors (as opposed to the exact expectation). But, our numerical evaluation, over various infection scenarios, demonstrated that our approach can significantly outperform state-of-the-art designs (up to 58% in the best case). Moreover, our designs performed well with the DD decoder, which allows us to claim that test designs are transferable to other decoders. Finally, our results indicate that there is still space for improvement in the traditional nonadaptive group-testing setting and new test designs (similar to the ones obtained through optimization) as well as their properties may be worth exploring in the future.

This is a first paper proposing this approach, and we believe that, given the promising first results, there are a number of open questions worth exploring. These include: 1. How do we use the continuous optimization formulation with other decoders? For decoders such as the definite defective decoder or belief propagation based ones, $\mathcal{E}(G)$ does not admit a nice form and we currently do not have an approach to calculate a non-trivial lower bound; this remains a challenging open problem. 2. Can we use this approach with noisy group testing? 3. Can we use this approach with other (non-binary output) group test models? 4. One could also ask if the continuous optimization formulation can be used to design a separate decoder, by formulation the decoding problem as a combinatorial optimization problem.

VIII. Acknowledgements

This work was supported in part by NSF grants #2007714, #2139304, #2146838 and #2146828. We also thank Katerina Argyraki for her ongoing support and the valuable discussions we have had about this project.

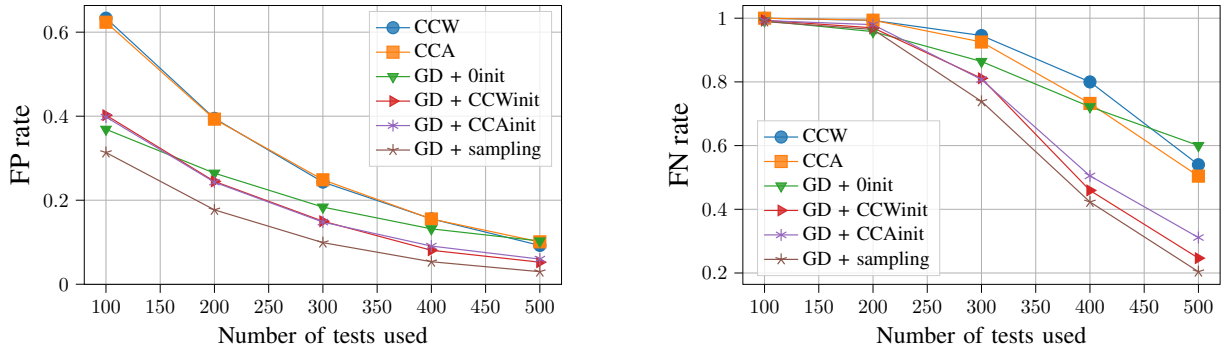
[7]Interestingly, CCW outperforms CCA here, despite using less information about the priors. We refer the reader to Figures 3 and 4 for cases where CCA outperforms CCW.



(a) DND decoder.

(b) DD decoder.

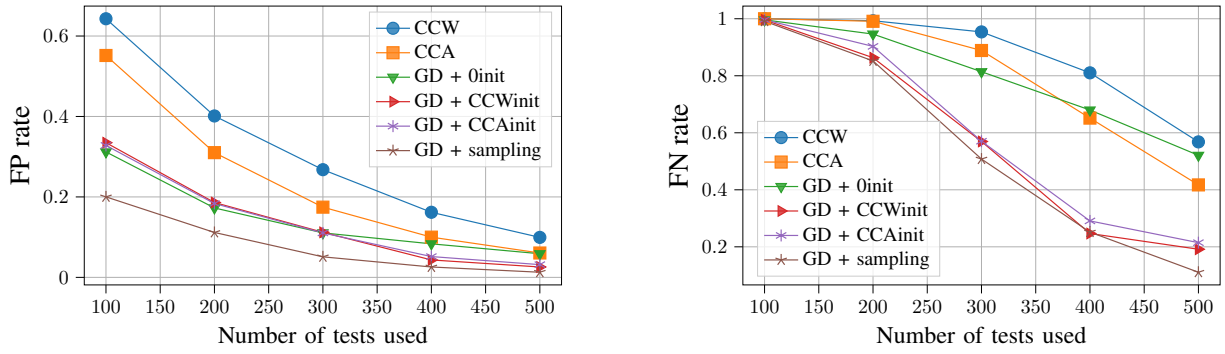
Fig. 2: Priors sampled from an exponential distribution with mean 0.05, $N = 1000$. We average over 10 such instances.



(a) DND decoder.

(b) DD decoder.

Fig. 3: Priors sampled from a discrete bimodal distribution (priors take value 0.02 or 0.3) with mean 0.1, $N = 1000$. We average over 10 such instances.



(a) DND decoder.

(b) DD decoder.

Fig. 4: Priors sampled from a discrete bimodal distribution (priors take value 0.02 or 0.5) with mean 0.1, $N = 1000$. We average over 10 such instances.

APPENDIX

A. Proof of Lemma 1

Proof. We first note that for any $\mathbf{q} \in [0,1]^n$ we have,

$$\mathbb{E}_{\mathbf{X}\sim Ber(\mathbf{q})} g(\mathbf{X}) \geq \min_{\mathbf{x}\in\{0,1\}^n} g(\mathbf{x}),$$

since the expectation of a random variable is at least as large as its minimum value over its support. Since the above holds for any $\mathbf{q}$, as a result we have

$$\min_{\mathbf{q}\in[0,1]^n} \mathbb{E}_{\mathbf{X}\sim Ber(\mathbf{q})} g(\mathbf{X}) \geq \min_{\mathbf{x}\in\{0,1\}^n} g(\mathbf{x}). \tag{13}$$

Let $\mathbf{x}^*$ be a minimizer of $g(\mathbf{x})$ in (6). The choice of $\mathbf{q}^* = \mathbf{x}^*$ (i.e. $\mathbf{X} = \mathbf{x}^*$ with probability 1) gives

$$f(\mathbf{q}^*) = \mathbb{E}_{\mathbf{X}\sim Ber(\mathbf{q}^*)} g(\mathbf{X}) = g(\mathbf{x}^*) = \min_{\mathbf{x}\in\{0,1\}^n} g(\mathbf{x}). \tag{14}$$

From (13) and (14) we conclude that

$$\min_{\mathbf{q}\in[0,1]^n} \mathbb{E}_{\mathbf{X}\sim Ber(\mathbf{q})} g(\mathbf{X}) = \min_{\mathbf{x}\in\{0,1\}^n} g(\mathbf{x}).$$

In order to obtain a solution to (6), we obtain a solution $\mathbf{q}^*$ of (7) and simply sample from $\mathbf{X} \sim Ber(\mathbf{q}^*)$ (sample $X_i \sim \text{Ber}(q_i)$). Any such value taken by $\mathbf{X}$ is guaranteed to be a solution of (6). To see this, first note that from the definition, $f(\mathbf{q}^*)$ can be written as $f(\mathbf{q}^*) = \sum_{\mathbf{x}:\mathbf{x}\in Supp(\mathbf{X}\sim Ber(\mathbf{q}^*))} \Pr(\mathbf{X}=\mathbf{x}) g(\mathbf{x})$. Basically, $f(\mathbf{q}^*)$ is a convex combination of some $g(\mathbf{x})$ terms. But from (14), we know that $f(\mathbf{q}^*) = g(\mathbf{x}^*)$, the minimum value taken by $g(\mathbf{x})$. For this to hold true, we need $g(\mathbf{x}) = g(\mathbf{x}^*)$ for every $\mathbf{x} \in Supp(\mathbf{X} \sim Ber(\mathbf{q}^*))$, otherwise $f(\mathbf{q}^*) > g(\mathbf{x}^*)$. □

B. Theorem 1 proof: filling in the gaps

In the proof of Theorem 1 we claimed the following:

$$\mathbb{E}_{\mathbf{U}\setminus\{i\}} \prod_{t=1}^{T} \gamma_{t,i} \geq \prod_{t=1}^{T} \mathbb{E}_{\mathbf{U}\setminus\{i\}} \gamma_{t,i}.$$

where $\gamma_{t,i} \triangleq \left(1 - G_{ti} \prod_{\substack{j=1:\\ j\neq i}}^{N} (1 - G_{tj}U_j)\right)$. We prove this using the Fortuin–Kasteleyn–Ginibre (FKG) inequality (see [39]–[41] or proof of Lemma 4 in [18]), restated here for convenience.

Lemma 2 (FKG inequality)**.** *Consider a finite distributive lattice Γ with partial ordering $\prec$ and meet ($\wedge$) and join operators ($\vee$). Consider a probability measure μ on Γ that is log-supermodular, i.e.,*

$$\mu(a)\mu(b) \leq \mu(a \wedge b)\mu(a \vee b) \ \forall \ a, b \in \Gamma.$$

Then, any two functions f and g which are non-decreasing on Γ are positively correlated, i.e.,

$$\mathbb{E}_\mu(fg) \geq \mathbb{E}_\mu(f)\mathbb{E}_\mu(g).$$

Remark: Consider $\Gamma = \{0,1\}^N$ with partial ordering $\prec$, where $a \prec b$ if every coordinate of b is at least as large as a. When the meet and join operators coincide with logical AND and logical OR respectively, this is a distributive lattice. It can be verified that any product measure μ on Γ is log-supermodular. As a result, any two functions f and g which are non-decreasing on Γ are positively correlated, i.e., $\mathbb{E}_\mu(fg) \geq \mathbb{E}_\mu(f)\mathbb{E}_\mu(g)$. Consequently, given any M non-negative, non-decreasing functions $f_1, f_2, ..., f_M$ one could inductively apply FKG inequality to obtain

$$\mathbb{E}_\mu(\prod_{i=1}^{M} f_i) \geq \prod_{i=1}^{M} \mathbb{E}_\mu f_i. \tag{15}$$

Given (15) what remains to be shown is that each $\gamma_{t,i}(\mathbf{U})$ is non-negative and non-decreasing as a function of $\mathbf{U} \in \{0,1\}^N$. To see that it is non-negative is straight-forward – we have $G_{tj}U_j \geq 0$ and hence $(1 - G_{tj}U_j) \leq 1$. Therefore, the product $\prod_{\substack{j=1:\\ j\neq i}}^{N} (1 - G_{tj}U_j) \leq 1$ and the non-negativity follows. To see that $\gamma_{t,i}(\mathbf{U})$ is non-decreasing, we first consider $\mathbf{U} \prec \mathbf{U}'$, i.e., $U_j \leq U'_j \ \forall \ j$. Then we have $(1 - G_{tj}U_j) \geq (1 - G_{tj}U'_j) \ \forall \ t, j$ and $\prod_{\substack{j=1:\\ j\neq i}}^{N} (1 - G_{tj}U_j) \geq \prod_{\substack{j=1:\\ j\neq i}}^{N} (1 - G_{tj}U'_j) \ \forall \ t$. Thus, $\gamma_{t,i}(\mathbf{U}) \leq \gamma_{t,i}(\mathbf{U}')$ and $\gamma_{t,i}$ is non-decreasing. Applying (15), we have

$$\mathbb{E}_{\mathbf{U}\setminus\{i\}} \prod_{t=1}^{T} \gamma_{t,i} \geq \prod_{t=1}^{T} \mathbb{E}_{\mathbf{U}\setminus\{i\}} \gamma_{t,i}.$$

C. Computing the objective function $f(Q)$

Here we give a $O(N^2)$ algorithm to compute the objective function $f(Q)$ in (11). We assume $T \leq N$ so $T = O(N)$ throughout. We first restate the expression for $f(Q)$ in (12):

$$f(Q) = \sum_{i=1}^{N} (1 - p_i) \prod_{t=1}^{T} (1 - Q_{ti} \prod_{j=1, j\neq i}^{N} (1 - Q_{tj}p_j)).$$

Note that this can be rewritten as:

$$f(Q) = \sum_{i=1}^{N} (1 - p_i) F[i],$$

where the intermediate terms are defined as

$$F[i] \triangleq \prod_{t=1}^{T} (1 - Q_{ti} G[t,i])$$

and

$$G[t,i] \triangleq \prod_{j=1, j\neq i}^{N} (1 - Q_{tj}p_j).$$

Thus, we first compute and store $G[t,i] \ \forall \ t,i$, which is then used to compute $F[i] \ \forall \ i$ in $O(N^2)$ time (assuming $T = O(N)$). Subsequently, $f(Q)$ can be computed from $F[i]$ in $O(N)$. Computing $G[t,i]$ takes $O(N^2)$ as one can first compute $H[t] \triangleq \prod_{j=1}^{N} (1 - Q_{tj}p_j)) \ \forall \ t$ in $O(N^2)$ time and obtain $G[t,i] = H[t]/(1 - Q_{ti}p_i)$ in $O(N^2)$. The overall time complexity of computing $f(Q)$ is $O(N^2)$.

D. Proof of Corollary 1

The proof is similar to Lemma 1. We first note that for any Q we have,

$$\mathbb{E}_{G\sim Ber(Q)}\mathcal{E}(G) \geq \min_{\{G\,:\,G_{ti}\in\{0,1\},\forall t,i\}} \mathcal{E}(G)\ \forall\ Q,$$

since the expectation of a random variable is at least as large as its minimum value over the support. As a result,

$$\min_{\{Q\,:\,Q_{ti}\in[0,1],\forall t,i\}} \mathbb{E}_{G\sim Ber(Q)}\mathcal{E}(G) \geq \min_{\{G\,:\,G_{ti}\in\{0,1\},\forall t,i\}} \mathcal{E}(G).$$

Moreover, the choice of $Q = G_{opt}$ gives

$$\mathbb{E}_{G\sim Ber(Q)}\mathcal{E}(G) = \mathcal{E}(G_{opt}) = \min_{\{G\,:\,G_{ti}\in\{0,1\},\forall t,i\}} \mathcal{E}(G),$$

and as a result we have

$$\min_{\{Q\,:\,Q_{ti}\in[0,1],\forall t,i\}} \mathbb{E}_{G\sim Ber(Q)}\mathcal{E}(G) = \min_{\{G\,:\,G_{ti}\in\{0,1\},\forall t,i\}} \mathcal{E}(G).$$

Thus it is sufficient to solve (10), and given an optimal solution Q_{opt}, any $G \sim Ber(Q_{opt})$ is a solution of (5).

E. Expression for each partial derivative in $\nabla_Q f(Q)$

Here, we give an expression for the gradient $\nabla f(Q)$ by calculating each partial derivative $\frac{\partial f(Q)}{\partial Q_{lm}}$.

$$\begin{aligned}
\frac{\partial f(Q)}{\partial Q_{lm}} &= \frac{\partial}{\partial Q_{lm}} \sum_{i=1}^{N}(1-p_i)\prod_{t=1}^{T}\left(1-Q_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-Q_{tj}p_j)\right)\\
&= \frac{\partial}{\partial Q_{lm}} \sum_{i=1:i\neq m}^{N}(1-p_i)\prod_{t=1}^{T}\left(1-Q_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-Q_{tj}p_j)\right)\\
&\quad + (1-p_m)\frac{\partial}{\partial Q_{lm}}\prod_{t=1}^{T}\left(1-Q_{tm}\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{tj}p_j)\right)\\
&\overset{(a)}{=} \sum_{i=1:i\neq m}^{N}(1-p_i)\frac{\partial}{\partial Q_{lm}}\left(1-Q_{li}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-Q_{lj}p_j)\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}\left(1-Q_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-Q_{tj}p_j)\right)\\
&\quad + (1-p_m)\frac{\partial}{\partial Q_{lm}}\left(1-Q_{lm}\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{lj}p_j)\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}\left(1-Q_{tm}\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{tj}p_j)\right)\\
&\overset{(b)}{=} \sum_{i=1:i\neq m}^{N}(1-p_i)\left(Q_{li}p_m\prod_{\substack{j=1:\\ j\neq i,j\neq m}}^{N}(1-Q_{lj}p_j)\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}\left(1-Q_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-Q_{tj}p_j)\right)\\
&\quad + (1-p_m)\left(-\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{lj}p_j)\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}\left(1-Q_{tm}\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{tj}p_j)\right), \qquad (16)
\end{aligned}$$

where in (a) we separate out the term corresponding to $t = l$ from the product term $\prod_{t=1}^{T}$ and apply the derivative in (b).

F. Computing $\nabla_Q f(Q)$

The computation of gradient follows a similar approach as the computation of the objective function $f(Q)$. We assume $T \leq N$ so $T = O(N)$ throughout. We first restate the expression for the gradient in (16):

$$\begin{aligned}
\nabla_{Q_{lm}} f(Q) &= \sum_{i=1:i\neq m}^{N}(1-p_i)\left(Q_{li}p_m\prod_{\substack{j=1:\\ j\neq i,j\neq m}}^{N}(1-Q_{lj}p_j)\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}\left(1-Q_{ti}\prod_{\substack{j=1:\\ j\neq i}}^{N}(1-Q_{tj}p_j)\right)\\
&\quad + (1-p_m)\left(-\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{lj}p_j)\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}\left(1-Q_{tm}\prod_{\substack{j=1:\\ j\neq m}}^{N}(1-Q_{tj}p_j)\right).
\end{aligned}$$

As we did in the case of objective function computation, we first simplify and rewrite this in terms of intermediate terms:

$$\begin{aligned}
\nabla_{Q_{lm}} f(Q) &= \sum_{i=1:i\neq m}^{N}(1-p_i)\left(Q_{li}\frac{p_m}{1-Q_{lm}p_m}G[l,i]\right)\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}(1-Q_{ti}G[t,i])\\
&\quad + (1-p_m)\,(-G[l,m])\\
&\quad\cdot\prod_{t=1:t\neq l}^{T}(1-Q_{tm}G[t,m])\\
&= \frac{p_m}{1-Q_{lm}p_m}\sum_{i=1:i\neq m}^{N}(1-p_i)\,(Q_{li}G[l,i])\,F[l,i]\\
&\quad + (1-p_m)\,(-G[l,m])\,F[l,m]\\
&= \frac{p_m}{1-Q_{lm}p_m}\left(\sum_{i=1}^{N}(1-p_i)Q_{li}G[l,i]F[l,i]\right.
\end{aligned}$$

$$
\begin{aligned}
&\quad - (1-p_m)Q_{lm}G[l,m]F[l,m]\Big) \\
&\quad + (1-p_m)\left(-G[l,m]\right)F[l,m] \\
&= \frac{p_m}{1-Q_{lm}p_m}\sum_{i=1}^{N}(1-p_i)Q_{li}G[l,i]F[l,i] \\
&\quad - (1-p_m)G[l,m]F[l,m]\left(\frac{1}{1-Q_{lm}p_m}\right),
\end{aligned}
$$

where the intermediate terms are

$$F[l,i] \triangleq \prod_{t=1:t\neq l}^{T}(1-Q_{ti}G[t,i])$$

and

$$G[t,i] = \prod_{j=1,j\neq i}^{N}(1-Q_{tj}p_j).$$

As we showed earlier, computing $G[t,i]\ \forall\ t,i$ can be done in $O(N^2)$ runtime complexity, and $F[l,i]$ can be obtained as $\frac{H[i]}{1-Q_{li}G[l,i]}$ where $H[i] \triangleq \prod_{t=1}^{T}(1-Q_{ti}G[t,i])$. Clearly, $H[i]\ \forall\ i$ can be obtained once in $O(N^2)$ and reused to compute $F[l,i]\ \forall\ l,i$ in $O(N^2)$. Having computed F and G terms, one could again use a similar trick to pre-compute $J[l] \triangleq \sum_{i=1}(1-p_i)Q_{li}G[l,i]F[l,i]\ \forall\ l$ in $O(N^2)$. With this, one could now compute each gradient term $\nabla_{Q_{lm}}$ in $O(1)$ thus giving an overall time complexity $O(N^2)$.

REFERENCES

[1] C. Gollier and O. Gossner, "Group testing against covid-19," April 2020. [Online]. Available: https://www.tse-fr.eu/publications/group-testing-against-covid-19

[2] M. Broadfoot, "Coronavirus test shortages trigger a new strategy: Group screening," May 2020. [Online]. Available: https://www.scientificamerican.com/article/coronavirus-test-shortages-trigger-a-new-strategy-group-screening2/

[3] J. Ellenberg, "Five people. one test. this is how you get there." *NYtimes*, May 2020.

[4] C. Verdun *et al.*, "Group testing for sars-cov-2 allows up to 10-fold efficiency increase across realistic scenarios and testing strategies," *medRxiv*, 2020.

[5] S. Ghosh *et al.*, "Tapestry: A single-round smart pooling technique for covid-19 testing," *medRxiv*, 2020.

[6] L. M. Kucirka, S. A. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, "Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based sars-cov-2 tests by time since exposure," *Annals of Internal Medicine*, vol. 173, pp. 262–267, Aug. 2020.

[7] S. Mallapaty. (2020) The mathematical strategy that could transform coronavirus testing. [Online]. Available: https://www.nature.com/articles/d41586-020-02053-6

[8] FDA. (2020) Pooled sample testing and screening testing for covid-19. [Online]. Available: https://www.fda.gov/medical-devices/coronavirus-covid-19-and-medical-devices/pooled-sample-testing-and-screening-testing-covid-19

[9] R. Dorfman, "The detection of defective members of large population," *The Annals of Mathematical Statistics*, vol. 14, pp. 436–440, 1943.

[10] M. Aldridge, O. Johnson, and J. Scarlett, "Group testing: an information theory perspective," *CoRR*, vol. abs/1902.06002, 2019.

[11] D.-Z. Du and F. Hwang, *Combinatorial Group Testing and Its Applications*. Series on Applied Mathematics, 1993.

[12] Y. Malinovsky and P. Albert, "Revisiting nested group testing procedures: New results, comparisons, and robustness," *The American Statistician*, vol. 73, no. 2, pp. 117–125, 2019.

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

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

[15] C. L. Chan, S. Jaggi, V. Saligrama, and S. Agnihotri, "Non-adaptive group testing: Explicit bounds and novel algorithms," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, p. 3019–3035, 2014.

[16] J. Scarlett and V. Cevher, "Phase transitions in group testing," in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. SIAM, 2016, pp. 40–53.

[17] O. Johnson, M. Aldridge, and J. Scarlett, "Performance of group testing algorithms with near-constant tests per item," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 707–723, 2019.

[18] M. Aldridge, "Individual testing is optimal for nonadaptive group testing in the linear regime," *IEEE Trans. Inf. Theory*, vol. 65, no. 4, 2019.

[19] W. H. Bay, J. Scarlett, and E. Price, "Optimal non-adaptive probabilistic group testing in general sparsity regimes," *Information and Inference: A Journal of the IMA*, vol. 11, no. 3, pp. 1037–1053, 02 2022. [Online]. Available: https://doi.org/10.1093/imaiai/iaab020

[20] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Optimal group testing," ser. Proceedings of Machine Learning Research, J. Abernethy and S. Agarwal, Eds., vol. 125, Jul. 2020, pp. 1374–1388.

[21] I. Armendáriz, P. A. Ferrari, D. Fraiman, J. M. Martínez, and S. P. Dawson, "Group testing with nested pools," *arXiv preprint arXiv:2005.13650*, 2020.

[22] E. Price and J. Scarlett, "A fast binary splitting approach to non-adaptive group testing," *arXiv preprint arXiv:2006.10268*, 2020.

[23] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick, "Information-theoretic and algorithmic thresholds for group testing," *IEEE Trans. Inf. Theory*, 2020.

[24] T. Li, C. L. Chan, W. Huang, T. Kaced, and S. Jaggi, "Group testing with prior statistics," in *2014 IEEE International Symposium on Information Theory*, 2014, pp. 2346–2350.

[25] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. Diggavi, "Community-aware group testing," *IEEE Transactions on Information Theory*, vol. 69, no. 7, pp. 4361–4383, 2023.

[26] ——, "Group testing for connected communities," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, vol. 130. PMLR, 2021, pp. 2341–2349.

[27] ——, "Group testing for overlapping communities," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–7.

[28] J. Zhu, K. Rivera, and D. Baron, "Noisy pooled pcr for virus testing," *arXiv preprint arXiv:2004.02689*, 2020.

[29] R. Goenka, S.-J. Cao, C.-W. Wong, A. Rajwade, and D. Baron, "Contact tracing enhances the efficiency of covid-19 group testing," *arXiv preprint arXiv:2011.14186*, 2020.

[30] S. Ahn, W.-N. Chen, and A. Ozgur, "Adaptive group testing on networks with community structure," *arXiv preprint arXiv:2101.02405*, 2021.

[31] P. Bertolotti and A. Jadbabaie, "Network group testing," *arXiv preprint arXiv:2012.02847*, 2020.

[32] M. Aldridge, O. Johnson, and J. Scarlett, "Improved group testing rates with constant column weight designs," in *2016 IEEE International Symposium on Information Theory (ISIT)*. Ieee, 2016, pp. 1381–1385.

[33] I. Kiss, J. Miller, and P. Simon, *Mathematics of Epidemics on Networks*, 01 2017, vol. 46.

[34] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi, "Dynamic group testing to control and monitor disease progression in a population," *arXiv preprint arXiv:2106.10765*, 2021.

[35] ——, "An entropy reduction approach to continual testing," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 611–616.

[36] S. R. Srinivasavaradhan, M. Du, S. N. Diggavi, and C. Fragouli, "Algorithms for reconstruction over single and multiple deletion channels," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3389–3410, 2020.

[37] S. R. Srinivasavaradhan, S. Diggavi, and C. Fragouli, "Equivalence of ml decoding to a continuous optimization problem," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 343–348.

[38] L. Trevisan, "Combinatorial optimization: exact and approximate algorithms," *Stanford University*, 2011.

[39] C. M. Fortuin, P. W. Kasteleyn, and J. Ginibre, "Correlation inequalities on some partially ordered sets," *Communications in Mathematical Physics*, vol. 22, no. 2, pp. 89–103, 1971.

[40] J. Kemperman, "On the fkg-inequality for measures on a partially ordered space," in *Indagationes Mathematicae (Proceedings)*, vol. 80, no. 4. North-Holland, 1977, pp. 313–331.

[41] P. Fishburn, *Encyclopedia of Mathematics*. European Mathematical Society. [Online]. Available: http://encyclopediaofmath.org/index.php?title=FKG_inequality&oldid=14368