Engineering Competitive and Query-Optimal Minimal-Adaptive Randomized Group Testing Strategies

MUHAMMAD AZAM SHEIKH

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Göteborg, Sweden 2011

Thesis for the degree of Licentiate of Engineering

Engineering Competitive and Query-Optimal Minimal-Adaptive Randomized Group Testing Strategies

Muhammad Azam Sheikh

CHALMERS | GÖTEBORG UNIVERSITY

Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
Göteborg, Sweden 2011

Engineering Competitive and Query-Optimal Minimal-Adaptive Randomized Group Testing Strategies

Muhammad Azam Sheikh

This thesis has been prepared using LaTeX.

Technical Report No. 81L
ISSN 1652-876X

Algorithms Research Group
Computing Science Division
Department of Computer Science and Enginnering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Göteborg, Sweden

Phone: +46 (0)31 772 1031
Fax: +46 (0)31 772 3663
E-mail: azams@chalmers.se

Printed by Chalmers Reproservice
Göteborg, Sweden, May 2011

Never have I dealt with anything more difficult than my own soul, which sometimes helps me and sometimes opposes me.

–Imam Al-Ghazali

Engineering Competitive and Query-Optimal Minimal-Adaptive Randomized Group Testing Strategies

Muhammad Azam Sheikh
Department of Computer Science and Engineering
Chalmers University Of Technology

Abstract

Suppose that given is a collection of n elements where d of them are *defective*. We can query an arbitrarily chosen subset of elements which returns Yes if the subset contains at least one defective and No if the subset is free of defectives. The problem of group testing is to identify the defectives with a minimum number of such queries. By the information-theoretic lower bound at least $\log_2 \binom{n}{d} \approx d \log_2(\frac{n}{d}) \approx d \log_2 n$ queries are needed. Using adaptive group testing, i.e., asking one query at a time, the lower bound can be easily achieved. However, strategies are preferred that work in a fixed small number of stages, where queries in a stage are asked in parallel. A group testing strategy is called *competitive* if it works for completely unknown d and requires only $O(d \log_2 n)$ queries. Usually competitive group testing is based on sequential queries. We have shown that actually competitive group testing with expected $O(d \log_2 n)$ queries is possible in only 2 or 3 stages. Then we have focused on minimizing the hidden constant factor in the query number and proposed a systematic approach for this purpose.

Another main result is related to the design of query-optimal and minimal-adaptive strategies. We have shown that a 2-stage randomized strategy with prescribed success probability can asymptotically achieve the information-theoretic lower bound for $d \ll n$ and growing much slower than n. Similarly, we can approach the entropy lower bound in 4 stages when $d = o(n)$.

Acknowlegments

Nobody is born learned, but has the abilities to acquire knowledge through various sources. When it comes to difficulties in seeking knowledge, it is a must for the beginner to get benefit from those who have already acquired knowledge and wisdom in the field.

I started with no clues about the domain of my research work. However, after working with my supervisor Peter Damaschke, I feel affectionate to mention that together with his invaluable guidance and collaborative work, I had a great opportunity to learn from him and successfully conducted some good piece of research work. I am thankful to him for encouraging and providing all sorts of professional support.

I would like to thank to the members of the follow up committee, all my colleagues, and others who provided help in one way or the other for my research. I would also like to acknowledge Mr. Zulfiqar Hasan Khan, a fellow PhD student in the Signal Processing Group, for his help to resolve many issues regarding formatting of this thesis.

My wife and kids contributed with their smiles, love and support to achieve this goal. Finally, with all my heart I would like to extend sincere thanks and gratitude for my parents and family, specially the late Father, who has always been like a blessed cloud in the warm open sky. God bless you my great father!

Contents

Part I
Preamble

Chapter 1

Introduction

We will start with some informal talk over a little bit of background and origin of the problem addressed in this research. Then, in the next section we start with a broader definition of the problem and discuss its basic models. In section 3, we discuss different methods and their results to solve the problem.

Moving towards the end of this chapter, we first describe the scope of this thesis and summarize the main research contributions.

1.1 Background

Digging into the literature we find that the group testing (GT) problem, which has widely gained popularity in diverse disciplines, does not have a history spanned over centuries. It dates back to World War II, around 1943, when it was originally proposed. Mostly it is credited to Robert Dorfman due to his seminal report [8]. However, there are evidences that his colleagues, particularly David Rosenblatt, also contributed somehow at least in the initial brainstorming discussions where the idea was first presented. Group testing methods, their applications and other related history can be found in [11] by Du and Hwang.

The motivation behind development of the first group testing strategy was to reduce the number of blood tests required for syphilis screening of draftees for the U.S. Army [8]. Chemical analysis of a blood sample was used to reveal presence(positive outcome) or absence(negative outcome) of syphilis germs. There were millions of draftees with a possibility of only a few thousand of them actually suffering from the disease. Instead of the wasteful exhaustive approach of carrying out separate blood tests for all the individuals, an economical alternate method was suggested. The method was not successfully applied due to limitations of the testing procedure available at that time [11]. Nevertheless, we continue with this scenario to informally introduce the main theme behind origin of the GT problem.

The idea is based on the simple observation that blood samples drawn from a number of individuals can be analyzed simultaneously with just one test. In order to achieve that, we can pool the blood samples together to form a group blood sample and perform the chemical analysis test. Now, a negative test outcome would mean that entire group is healthy, i.e., nobody in the group is suffering from the syphilis. On the contrary, a positive test outcome would not reveal any potential information. From this we can only conclude that at least one of the group members is diseased. Depending on the group size, either we can go for individual testing or further dividing the group into subgroups. Each alternative has its own pros and cons. Negative outcome suggests larger group size but at the same time it increases the chances of positive outcome, thus favors for smaller group size. However, knowing that infected individuals are rare, we can hope that blood tests on groups of carefully chosen size will more often result into a negative outcome. Positive outcomes are minimized such that individual testing remains linear in the number of infected individuals only. Thus, we expect to reveal status of the entire individuals with less number of tests compared to their total count.

1.2 Preliminaries

Above introductory discussion is merely to serve as a quick overview about the basic idea of GT. It assumes that we have a prior information about the probability of each individual having the disease. Another situation could be that we know the actual value or an upper bound on the number of victims but not their identities. Being more realistic, in the natural setting we only know that "some" unknown number of individuals have the disease. Similarly there are variations depending on the construction of groups and the testing plan, i.e., performing tests sequentially one group at a time versus multiple groups in parallel.

Obviously before we extend our discussion to describe any group testing algorithm, we first need to formalize different versions of the problem statement. We start with some basic definitions and related jargon.

Let X be a set of size n. Usually n is very large and elements of X represent the collection of objects on which we want to perform testing. Each element has a binary status, i.e., it is either *defective* or not. Defective elements are also called *positive* while nondefective elements are called *negative*. We call the defective elements *defective set* or simply *defectives* for short.

Group: A group is any subset of X, also called a *pool*. A pool is said to be a *positive pool* if it contains at least one defective, and otherwise a *negative pool*.

Group Test: A *query* on a pool is called a group test and reveals whether the pool is positive or negative. A negative outcome declares that all elements are nondefective. On the other hand, a positive outcome just confirms presence of one or more defectives without revealing their identity information. In the basic setting, it is assumed that the cost of a query is fixed regardless of the group size.

Throughout we use words query or test synonymously to always refer to a group test. We refer to a group testing algorithm as a *group testing strategy*, or just a *strategy*. Next, we discuss statistical and combinatorial models for the GT problem.

1.2.1 Statistical Model

The probabilistic view requires a probability distribution over the defectives and named as *probabilistic group testing* (PGT) problem. The objective is to minimize the expected number of pools required to reveal the status of the

entire collection from the binary test outcomes under the assumed probability model. PGT is most often studied under the binomial probability distribution, called *binomial group testing*, i.e., each element is independently defective with fixed probability r. Binomial distribution is called the standard assumption for PGT [7].

The entropy lower bound for PGT is that on average for each element we need $r\log(\frac{1}{r}) + (1-r)\log(\frac{1}{1-r})$ queries, or $\log(\frac{1}{r}) + (1-r)\log\frac{(\frac{1}{1-r})}{r}$ queries per defective. Here and in the following, logarithms are always base 2, if not said otherwise. For small r this simplifies to $\log(\frac{1}{r}) + \log e$ expected queries per defective. The *cut-off point* for the statistical model, i.e., the ratio r below which group testing becomes more efficient than the trivial individual testing, is due to Ungar [25]. He proved that the individual testing minimizes the expected queries compared to any PGT if $r \geq \frac{1}{2}(3-\sqrt{5})$.

1.2.2 Combinatorial Model

In the *combinatorial group testing* (CGT) problem first studied by Li [21], besides the set X of elements, the number of defectives $d \leq n$ or an upper bound on d is also known a priori. Usually d is small compared to n, and the task is to find the defectives by asking a minimum number of queries to arbitrarily chosen pools.

Compared to the PGT where the defectives follow a fixed probability model and tries to reduce the expected number of queries. CGT are strategies which strive to minimize the query number for worst case scenario. The only information CGT requires about the defectives is that it is a subset of X consisting of at most d elements. It should also be noted that in case of CGT, d does not necessarily grow at a constant rate for increasing value of n.

CGT problem for exactly d defectives is called the (d,n) *group testing* problem whereas named as *generalized* (d,n) *group testing* [17] when d represents an upper bound on the defectives. It has been proved [18] that the generalized (d,n) problem only requires at most one additional test compared to the minimum number of tests required when exact d is known. Let $t(d,n)$ be the minimum number of queries required for the combinatorial model in the worst case. If there is no defective we only need one test consisting of all n elements. Whereas, in worst case we may have to test all items individually when $d = n$, thus naively we have $1 \leq t(d,n) \leq n$. We omit ceiling brackets in expressions for simplicity. By the entropy lower bound or so called information-theoretic lower bound, at least $\log\binom{n}{d}$ pools are needed if d out of n are defective.

Using Stirling's approximation we can write.
$\log \binom{n}{d} \approx n \log n - d \log d - (n-d) \log(n-d) + \frac{1}{2}(\log n - \log d - \log(n-d))$.
Let $x = \frac{d}{n}$, simplifying the above expression we get:
$\log \binom{n}{d} \approx d(\log(\frac{n}{d}) + f(x)) + c$, where $f(x) = (1 - \frac{1}{x}) \log(1-x)$ and the constant term $c = \frac{1}{2}(\log(\frac{1}{d}) - \log(1-x))$. The additive term $f(x)$ increases as the ratio x decreases and attains its maximum value of 1.44 for very small x. The optimal cut-off point for combinatorial group testing was studied in [16]. They proved that $n-1$ queries are necessary for $x \geq \frac{1}{3}$ if the groups of at most two elements are allowed. Du and Hwang [10] have presented a more general proof for slightly larger value, i.e., $x \geq \frac{8}{21}$. At this cut-off value, we have $f(x) = 1.13$, which is still greater than one. Thus, the worst case lower bound for the CGT problem is $t(d,n) \geq d \log(\frac{n}{d}) + 1.44d$. For known d, the upper bound $t(d,n) \leq \log \binom{n}{d} + d - 1$ is proven by Hwang [17] for his *generalized binary splitting algorithm.*

1.3 GT Methods and Complexity Bounds

Regardless of the model, testing methods for a GT problem can be adaptive, nonadaptive or work in a specified number of stages. These testing methods have been studied extensively and it is well known that optimally achievable complexity bounds differ based on the choice of a testing method. In either case, the complexity of a GT strategy is the largest number of pools that are queried to separate all defectives from the rest.

In the following we are mainly concerned about complexity bounds for the combinatorial model, though the basic definitions are equally applicable to both of the group testing models.

1.3.1 Adaptive Strategies

A strategy is called *adaptive* if group tests are conducted sequentially and every pool can be chosen based on the outcomes of all previous queries. An upper bound for adaptive strategies is $O(d \log n)$. It follows simply from the *halving strategy*, i.e., first query the whole set, if outcome is positive, divide the elements into two pools of equal size and query one of these. If it is positive, continue halving with this pool, otherwise with the second half. In this way, with $O(\log n)$ adaptive queries we can identify one defective and remove it from the set. Repeating it d times we get the upper bound. Since each iteration starts with a group test on the whole set, the process stops when no defectives are left. Therefore, in this adaptive strategy we do not need any prior knowledge of d and essentially $d \log n$ queries are sufficient.

While the lower bound for adaptive strategies is $d\log(\frac{n}{d})$, a strategy with $O(d\log(\frac{n}{d}))$ can be easily devised even for unknown d. This is actually due to Du and Hwang [9] who first studied the group testing problem when nothing is known about d beforehand. Inspired from the study of online algorithms [22], they proposed *competitive group testing*. Let $t_A(d|n)$ denote the minimum number of queries required by an algorithm A if there are d unknown defectives. Then A is called *c-competitive* if there exist constants c and a such that $t_A(d|n) \leq ct(d,n) + a$ holds for $0 \leq d < n$. Beginning with [1] [13] [14], substantial work has been done to minimize the constant factor c, called the *competitive ratio*. To our best knowledge, the currently best competitive ratio for deterministic, adaptive strategies is 1.5 [24].

1.3.2 Nonadaptive Strategies

Nonadaptive strategies are on the other extreme, that is, all the pools should be prepared in advance without knowing the outcome of any test, and then queried simultaneously. Any nonadaptive strategy requires $\Omega(\frac{d^2}{\log d}\log n)$ pools even for known d. However, $O(d^2 \log n)$ pools are sufficient and currently best factor is 4.28; see [3] and the references therein.

Among others, nonadaptive strategies have applications in molecular biology experiments where these strategies are often referred to as *pooling designs*. A complete book has been written on this subject, an interested reader is referred to [12] for details.

1.3.3 Multistage Strategies

In between the above two, as a third option, there are *multistage* strategies or called *s-stage* strategies where the testing plan is divided into a fixed number of s stages. A *stage* refers to one round of simultaneously querying pools. Stages are treated adaptive while all pools within a stage are prepared prior to the stage. Here the main advantage is that queries prepared for the next stage can depend on the outcome of all previous stages thus trying to achieve optimal query bounds with minimal adaptivity. In the context of stages, nonadaptive strategies are 1-stage group testing whereas fully adaptive strategies proceed with one query per stage.

The concept of multistage strategies is not new. The very first method proposed for the group testing problem by Dorfman [8] was actually a 2-stage PGT which was later extended to an *s-stage* strategy for CGT by Li [21]. Recently, multistage strategies have attracted the research community due to its successful application in the computational molecular biology problems such as DNA library screening. In these applications mostly 2-stage strategies

are preferred because the problem size n is huge and a group test may take several hours[20].

A 2-stage strategy is called *trivial* if stage 1 is used to find $O(d)$ candidate elements including all defectives, which are then tested individually in stage 2. These 2-stage strategies require an upper bound d on the number of defectives and guarantee that all the defectives can be identified using $cd\log(\frac{n}{d})$ pools, c being some constant. This was first shown in [7] with a high constant c, more precisely $7.54\,d\log(\frac{n}{d})$. It was later improved to $4\,d\log(\frac{n}{d})$ [15] and currently $c = 1.9$ for all d, and asymptotically to $c = 1.44$ as d grows [3].

1.4 Scope of the Thesis

Usually a GT strategy assumes the ideal situation where group tests are considered *error-free*, i.e., there are no testing errors and whenever we observe a positive pool, we are sure about the presence of at least one defective element. Similarly a negative pool in the absence of testing errors declares all items as nondefective.

In the discussion so far, we have been actually talking about the error-free model of group testing. However, in group testing applications, e.g., when screening DNA libraries, we cannot neglect testing error which presumably increases with the pool size. Similarly there are other situations where classical error-free model cannot be applied. There have been studies [4] where strategies that tolerate testing errors up to certain limits have been devised. We would not go into details as our current work is not in this direction.

Another source of error called the *design error*, can affect the outcome of a group testing strategy. Design error is basically due to the nature of the group testing strategy itself when it fails to identify all defective elements. A strategy can be based on *deterministic* or *randomized* construction of pools. While deterministic strategies do not have design error, randomized strategies sometimes allow for a small design error. Generally both deterministic and randomized strategies aim at minimizing the number of tests, however, in some cases randomization can achieve better complexity bounds, e.g., if the output is allowed to be incorrect with a small prescribed error probability. For example, an $O(d\log n)$ query number cannot be achieved deterministically in one stage [2]. Whereas, there exists a randomized 1-stage strategy which needs asymptotically only $1.45d\log n$ queries to identify up to d defectives for small fixed error probability [3]. It does not say that a randomized strategy with guaranteed output promises can not be constructed.

Besides some results for the deterministic case, our contributions are mainly based on randomized constructions.

1.5 Main Contributions

In the following, we first motivate for the need of work in our line of research. Then, we formulate some basic questions and highlight our main results. Later in the next chapter we will discuss some of them in detail.

1.5.1 Motivation

In the previous section 1.3.1, we studied competitive strategies. The main result for competitive GT is that despite the ignorance of d, we can devise GT strategies with a query number within a constant factor of optimum. A major ingredient of the best known competitive strategies is adaptivity [14] [24]. In many applications of GT, the time consuming nature of adaptive strategies is hardly acceptable. Therefore, competitive strategies that run in small fixed number of stages are desirable.

There are 2-stage strategies which are more favorable in these situations. However, these strategies suffer again from the restriction that the searcher must know d, or some close upper bound on d, in advance. Usually in GT applications d is unknown and some large enough d is assumed as upper bound. Such a 2-stage strategy guarantees an almost optimal query complexity (within constant factor) relative to this assumed d only. However, the strategy fails to find all defectives if the assumed bound was too small. On the other hand, it can be much larger than the true number of defectives in the particular case, leading to unnecessarily many tests. For these strategies, a good bound on d can save many tests.

Turning towards special situations such as, knowing a priori that there is a constant defective rate or d grows smaller than n, one may ask, can we exploit this additional information and what are characteristics of optimal strategies in these cases.

1.5.2 Research Questions and Results Highlights

We can enjoy using $O(d \log n)$ queries without knowing d, but we have to pay the cost of this freedom in terms of adaptivity. A natural demand is to ask for GT strategies which are "competitive" as well as "minimal adaptive". Now, we turn to the main focus of our research and define the following research question:

Can we take the best of two worlds and perform group testing without prior knowledge of d in a few stages, using a number of pools close to the information-theoretic lower bound?

We distinguish between deterministic and randomized strategies and separately state the two versions of the problem as follows:

- *Can we achieve competitive group testing that insists on $O(d \log n)$ pools in a constant number of stages using deterministic strategies?*

  Unfortunately the answer is No. There can not be a deterministic competitive strategy that succeeds in constant number of stages to achieve $O(d \log n)$ queries. In Paper-I it is proved that a strategy with above demands would require $\Omega(\frac{\log d}{\log \log d})$ stages.

- *Does there exist randomized constructions for previously unknown d that work in small fixed stages of parallel queries and closely achieve optimal query bounds?*

  Still the answer is No, because the proof in Paper-I is more general and also extends to randomized constructions with strict demands on stages and query number.

The key focus of our problem is to derive strategies for the case of unknown d where the complexity bounds are not influenced when the number of defectives vary a lot between the problem instances. At the same time, to minimize adaptivity, we can not allow for sequential queries. Then, a natural idea is that a strategy with such set of demands should start by estimating the magnitude of defectives in the given problem instance. Hence, to answer the research question, we can follow the two steps procedure:

- **Step-1**: Use nonadaptive queries and determine d; exactly or an upper bound.

- **Step-2**: Using d from step-1, apply the best known 1- or 2-stage group testing strategy.

While step-2 just involves selecting an appropriate strategy from the already established results, finding the number but not the identities of defective elements has been rarely studied. Therefore, all we need to achieve our goal is to have a group testing strategy that implements step-1. This problem can also be of independent interest in situations where one only wants an estimate of the number of defectives.
Following the above discussion, in order to answer our main question, we proceed further and redefine it as:

Can we estimate a previously unknown d using nonadaptive queries?

Once again our initial result is negative, i.e., determining d exactly would be

as hard as the group testing problem itself. However, we have shown that using $O(\log n)$ randomized nonadaptive queries and allowing the strategy to fail for a small prescribed probability, we can estimate d subject to a constant factor. Hence, according to the above two steps procedure, we can construct a randomized 2- or 3-stage competitive group testing strategy with $O(d \log n)$ pools that succeeds with given fixed probability. Results presented so far are based on Paper-I.

Related to the proposed randomized strategy, next obvious question an be formulated as follows.

Do we really need $O(\log n)$ pools to find an upper bound for d? Moreover, what are optimal constant factors in the query number and the accuracy of d depending on the prescribed error probability?

Main result in this direction is that a lower bound of $\Omega(\log n)$ pools is proved, for our particular but very natural way of choosing randomized pools for the 1-stage estimator. However, we have reasons to conjecture that $\Omega(\log n)$ is also a lower bound for any other randomized pooling design for our problem.

We have also put in quite a lot of efforts to minimize the hidden constant factors. We give practical methods to derive upper bound tradeoffs between the hidden constant in the query number $O(\log n)$ and the ratio of estimated versus true d for given error probability. To interpret the asymptotic behaviour, we devise a special method to generate sequence of randomized pools with a nice invariance property. Paper-II is entirely devoted to this lower bound proof and finding optimal constant factors.

Next, as part of our recent work in Paper-III, we extend our objective towards minimal-adaptive strategies where the constant factor in the leading term $d \log n$ is as close as possible to 1. Firstly, we consider the situations where defectives are rare and define the following question:

For d growing slower than n, can we have group testing strategies that work in a small fixed number of stages and their query complexity asymptotically reaches the information-theoretic lower bound?

We have shown that there exist randomized 2- or 3-stage strategies where the complexity bound converges to the entropy lower bound for d growing slowly with n. Again the construction allows for a small prescribed failure probability. we also considered the case of unknown d and derived similar bounds at the cost of one additional stage devoted to find a randomized estimate for d.
We have also considered a special case of statistical model of group testing, that is, when defectives appear at some constant rate $r = \frac{d}{n}$. Here, as opposed

to the previous case, now we assume that d grows as fast as n. We again put the same question but with the new adaption:

For fixed defective rate r, can we have group testing strategies that work in a small fixed number of stages and their query complexity asymptotically reaches the entropy lower bound?

For this particular case, we have presented a 4-stage randomized strategy that asymptotically achieves entropy lower bound for $r \to 0$. Here again, for unknown r, we can use our nonadaptive defectives estimation strategy.

Chapter 2

Randomized Constructions

The research has been carried out as a joint collaborative work, however, contributions made by an author vary for certain sections of the produced papers. The purpose of this chapter is to provide a brief summary of major results where author of this thesis has contributed substantially. For details of each section in this chapter, obviously one should consult the corresponding sections in the papers.

In the following, we divide the discussion into three main sections. First, we elaborate on the important aspects of the randomized construction that has been developed to estimate the number of defectives for competitive minimal-adaptive group testing strategies. In the next section, we will mainly focus on the particular strategy that we have adopted to find optimal constant factors in our randomized estimate of defectives. These constant factors are important to determine the minimum competitive ratio for the proposed strategies. The first two sections are based on our results in Paper-I and Paper-II.

In the last section, we discuss some results from Paper-III where again we have proposed randomized group testing strategies which require a fixed small number of stages. These strategies asymptotically attain entropy lower bound when defectives grow according to the certain criteria.

2.1 Competitive Group Testing in Only 2 or 3 Stages

Apparently we were the first to study this combination of demands, i.e., competitive and minimal-adaptive group testing strategies. As discussed in previous chapter, the core of this problem lies in deciding on how to tackle with unknown value of d. Once we have an estimate for d, subsequently we can use it in any 2-stage $O(d \log n)$ strategy that works for known d to obtain a randomized 3-stage competitive strategy. If we instead append a probabilistic 1-stage strategy from [3], which requires $O(d \log n)$ queries and succeeds with high probability, we even get a competitive group testing strategy that needs only 2 stages.

Now, we motivate our particular way to figure out a reasonable strategy to deal with unknown d.

2.1.1 Why A Randomized Estimate for Unknown d?

In the ideal case, aiming at minimum adaptivity, one would ask for a strategy that can determine d exactly using nonadaptive queries where the complexity bound of the strategy depends on d only. Unfortunately, this is not possible. We made the counterintuitive observation that determining d exactly would be as hard as the combinatorial group testing itself. Hence, the known lower complexity bounds for group testing carry over to this seemingly "simpler" problem. Thus, it would require $\Omega(\frac{d^2}{\log d} \log n)$ nonadaptive queries. Then, the natural choice is to hope for a method to estimate a close and reliable upper bound for d.

It was also proved that for unknown d, any deterministic strategy can not run in constant number of stages, i.e., deterministic competitive group testing in fixed number of stages is not possible. Therefore, we opted for randomized constructions. Particularly, we have derived a strategy to estimate d within a constant factor in 1 stage. We remark that an estimate of d subject to some constant is necessary for minimal-adaptive competitive group testing.

Our problem with unknown d was also raised in [19]. However, they have studied the problem only experimentally using several batching strategies. To our best knowledge, our work [6, 5], is the first to establish rigorous results.

2.1.2 Outline of the Randomized Strategy

we have developed a randomized strategy using nonadaptive queries to estimate d. The outline of our 1-stage strategy is as follows. To prepare a pool we fix some probability q, and put every element in the pool with probability $1-q$. The pool is negative with probability q^d, since this is the probability that d defectives are outside the pool. We increment $1-q$ in small steps such that we prepare $O(\log n)$ random pools of exponentially growing size and then query them simultaneously. The query results are independent because we put elements independently in each pool.

Now, pools of sizes smaller than $\frac{n}{d}$ will most probably be negative. Similarly those having sizes larger than $\frac{n}{d}$ will most probably be positive. Thus, such pools convey very little information about the magnitude of d. However, the cut-off point between negative and positive pool sizes can be used to estimate d.

We have actually shown that a "conservative bound" $\hat{d}$ on the number d of defectives in a set of n elements can be determined by $O(\log n)$ randomized nonadaptive group tests, such that:

- On the one hand, d is underestimated (that is, $\hat{d} < d$) with probability at most ϵ.
- On the other hand, the expected ratio $\frac{\hat{d}}{d}$, in the good case $\hat{d} \geq d$, is bounded by some constant expected factor c, independently of d.

Thus, $\hat{d}$ can be further used in any group testing strategy that needs an upper bound on d: such a strategy fails only with a small probability due to $\hat{d} < d$, but it is also unlikely to waste too many tests due to a large $\frac{\hat{d}}{d}$.

2.2 Engineering Optimal Competitive Ratio

Let g be some constant and suppose that we use $L = g \log n$ nonadaptive queries to get the estimate $\hat{d}$ with expected factor $c = \frac{\hat{d}}{d}$ for a given error probability ϵ. As discussed in section 1.3.3, there are 2-stage group testing strategies where $c'd \log n$ queries are sufficient for known d. Just to remind, for the best result in this direction, constant factor $c' = 1.9$ [3].

Now, for unknown d we can use our randomized estimation to get $\hat{d} = cd$ in stage 1 and appending the best known 2-stage strategy afterwards, we get a 3-stage competitive group testing strategy with $g \log n + c'cd \log n$ or $(\frac{g}{d} + c'c)d \log n$ expected queries in total. In GT strategies aiming at unknown d, the constant factor in the query number is usually called the competitive

ratio. In our case as discussed above, the competitive ratio is determined by the factor $\frac{g}{d} + c'c$. For $d = 1$, the competitive ratio becomes $g + c'c$ while asymptotically for growing d we have $c'c$. Since in any case the factor c has direct impact on the total expected query number, an optimal value for c can save many pools.

Apparently, at the cost of a large g, we can make c arbitrarily close to 1. However, the worst-case competitive ratio $g + c'c$ depends on both. Therefore, to minimize the competitive ratio, an optimal strategy should balance the number $g \log n$ of pools and the expected ratio c. Driven by this, our next goal was to achieve optimal values of the constants c and g for given error probability ϵ. Getting an optimal tradeoff turns out to be a highly nontrivial problem in itself. We first formalize it as an independent problem and then present a systematic approach to solve it.

Formalizing the Problem: Now we formally describe the problem of estimating d from the test outcomes with the objective to achieve optimal factors. Let us represent a positive test outcome with 1 and a negative test outcome with 0. As discussed above in the section 2.1, we fix some q and select each element independently with the same probability $1-q$. In this way, we characterize each randomized pool by only one number: the probability q_k *not* to put an element in the kth pool.

For the given problem size n, we fix some value for g, e.g., $1, 1.5, 2$, *etc.*, and prepare $L = g \log n$ pools using corresponding q_k probabilities where $k = 0, 1, ..., L-1$. Let $s = s_0 \dots s_{L-1}$ be the binary string representing nonadaptive query outcomes for our randomized estimator, the problem can be defined as:

For already fixed $L = g \log n$ and the probabilities q_k, predict the unknown number $d \in [1, n]$ from the string s of test outcomes such that the expected accuracy $\frac{\hat{d}}{d}$ is minimized, but at the same time $\hat{d} < d$ with probability at most ϵ.

The essence of this problem depends on two things. First, how we define the sequence of probabilities q_k. Second and most important, once the pool sizes are fixed, how we use the information contained in the result string s to make efficient guess. Remember, test outcomes where pool sizes are fairly smaller(larger) than $\frac{n}{d}$ will most probably be negative(positive). Therefore, a searcher can ignore extreme ends of the binary string s and estimate d from the pool sizes where test results toggle between 0 and 1. The situation is depicted in Figure 2.1.

However, the searcher may utilize the entire binary string s. Although a pool size far from $\frac{n}{d}$ has a low probability to produce an unexpected outcome, in case it happens, the searcher needs to quantify its influence on the

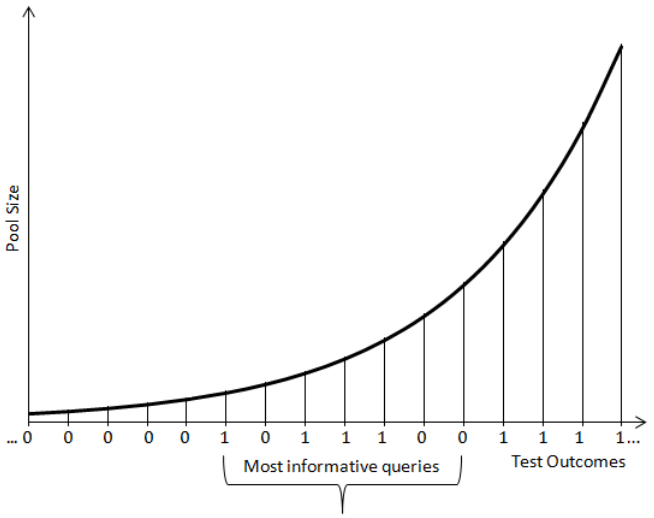

Figure 2.1: A typical Pools versus Test Outcomes Scenario

estimated value of d.

we will discuss a practical method to achieve optimal competitive ratio for our randomized competitive minimal-adaptive group testing strategies. We again emphasis that worst case competitive ratio for our 3-stage competitive GT strategy is determined by the factor $g+c'c$. Here c' is the constant factor due to the 2-stage strategy whereas g and c are the factors which we get as a result of our randomized estimate for d in stage 1. Now, to derive asymptotic upper bound, we actually need to find the best possible tradeoff between the pool number $L = g \log n$ and the expected accuracy $\frac{\hat{d}}{d} \leq c$. We will also discuss results for fixed n.

2.2.1 Ad hoc Rules

As first attempts we tried out some *ad hoc* rules. First, we give an equivalent definition of our problem, based on which we will construct our particular rules to solve the problem.

Suppose that q_{kd} denotes the conditional probability that the kth pool is negative when there are d defectives. Since a pool is negative if and only if none of the defectives is selected, and elements are selected independently, this simply yields $q_{kd} = q_k^d$ and we have $d = \frac{\log q_k}{\log q_{kd}}$. Then the task of finding $\hat{d}$ can be formulated as follows:

Based on the result string s, the objective is to estimate $\hat{q}_{kd}$ for any fixed k, i.e., the probability with which kth pool produces a negative outcome.

Since q_k is known, now using our estimate $\hat{q}_{kd}$ instead of the actual q_{kd}, we can predict the corresponding $\hat{d}$. Note that a good estimate of $\hat{q}_{kd}$ is crucial. If it is lesser than the actual, then the predicted $\hat{d}$ will be larger and we will observe a large factor c. On the other hand an overestimated $\hat{q}_{kd}$ fails to predict $\hat{d} \geq d$.

We started with a basic approach and the idea is as follows. We prepare a number of pools for fixed $g = \frac{L}{\log n}$ using the already defined probabilities q_k. Let us index the pools according to increasing size and perform parallel queries to get the result string s. Let i be the largest index of a negative pool. We call it the *main index* and let q_i be the probability with which this pool was prepared. Now, assuming $\hat{q}_{jd} = \frac{1}{2}$, where $j := i - m$ for some previously fixed m that depends on the desired failure probability bound, we can compute our estimate $\hat{d}_j = \frac{1}{\log \frac{1}{q_j}}$. Initially we considered $m = 0, 1, 2, \ldots, i$. However, we realized that this approach is not reliable because the position of the main index can easily be an outlier. To overcome this, we considered m moving along the negative pool indices only, jumping over the positive pools.

Similar to the main index approach, we can base our estimate on the index of the smallest positive pool. Let l denotes this index. Test outcomes between pool numbers l and i form the window that captures the most informative queries. In order to get more robust estimates, we have tried different rules that combine the test outcomes in this window and estimated d accordingly. We implemented all rules and compared their results empirically. Variant of main index approach where we loop over negative pool indices only, and some averaging rule gave the best results. We will discuss the numerical results in a separate section.

2.2.2 A Linear Programming Formulation

In the previous discussion we mentioned index $j := i - m$ which represents the pool on which we based our estimation. Choosing most relevant j actually plays an important role in minimizing the factor c and also observing the given failure probability constraint. There, for some fixed ϵ, we empirically predicted appropriate m based on several runs of the simulation. From this we learned that we can formulate it as a linear program that returns probabilities of choosing j for a given error bound and fixed number of pools and their sizes. For practical purposes also, it is nice to know which c we can accomplish for a given problem size n, $L = g \log n$ pools of predefined sizes and error probability ϵ.

This again led us towards an experimental work. We formulated the

problem of minimizing c as a linear programming (LP) optimization problem. For given ϵ and predetermined pools, our LP minimizes the expected upper bound on c by assigning probabilities of choosing every $d \in [1, n]$ against each possible string s of query outcomes (2^L in total). To explore c vs. g tradeoff, for the same ϵ we can increment g at fixed steps to get corresponding c values. For exact LP formulation and related discussion, we refer to section 5 in Paper-II.

In the next section we discuss construction of randomized pools which we used in our LP followed by the numerical results.

2.2.3 Translation Invariant Pooling Design

Before we go into further details to derive the asymptotic behaviour, we first elaborate a nice way of choosing q_k probabilities. These probabilities determine the pool sizes for our strategy and provide us an important structure required to interpret optimal results from the test outcomes.

We can extend over the fact used in ad hoc rules. That is, choosing $q_{kd} = \frac{1}{2}$, we can compute corresponding value d_k of the defective number using $d_k = \frac{1}{\log \frac{1}{q_k}}$ for all k. In other words, for every q_k there exists a corresponding d_k such that the pool would answer 0 or 1 with probability $\frac{1}{2}$. We call such d_k a *query point* which is actually a function of the corresponding probability q_k.

Our goal is to estimate d for every possible $1 \leq d \leq n$ within some constant factor c. In the $\Omega(\log n)$ lower bound proof, it is actually shown that remote query points (on the logarithmic axis of defective numbers) being almost surely positive or negative, give too little information about the exact position of d. It means that we should carefully choose probabilities q_k such that the resulting sequence of query points is within a constant factor of every d. Intuitively, the ratio c we can achieve is related to the largest multiplicative gap $\frac{d_{k+1}}{d_k}$ between neighbored query points. Hence, it was also suggested as part of the lower bound proof that in an optimal pooling design for our randomized estimation problem, the query points should divide the logarithmic axis between 1 and n defectives equidistantly.

From the above discussion, it is evident that choice of the probability values q_k is crucial. So far in our discussion of ad hoc rules and LP formulation, we have not mentioned how we choose these values. There, we assumed that these probabilities are already defined. Now, here we motivate a specific way of choosing them. We take some starting value $q_0 < 1$. Then, for some fixed ratio $b > 1$, we define corresponding probabilities q_k such that every number is the bth power of the previous number and bth root of the next number.

Table 2.1: Optimal values c for $1 \leq d \leq 32$.

g	ϵ 0.01	ϵ 0.02	ϵ 0.03	ϵ 0.04	ϵ 0.05
0.5	10.67	9.29	8.41	7.77	7.20
1	5.24	4.46	4.03	3.74	3.55
1.5	4.09	3.58	3.21	2.99	2.82
2	3.69	3.20	2.92	2.78	2.63
2.5	3.61	3.08	2.83	2.59	2.50
3	3.42	3.01	2.81	2.63	2.42

Thus, we have $q_{k-1} = q_k^b$ for all k. This gives us *translation invariance*, i.e., query points are placed equidistant on the logarithm axis of the defective numbers.

Now, we are ready to present some numerical results in the next section.

2.2.4 Numerical Results

For a given problem size n, we actually prepare $L = \frac{\log n}{\log b}$ pools, and we have $g = \frac{1}{\log b}$. For the ad hoc rules, extensive simulations done in Matlab with independent random choices of d and very large n, suggested that for $0.01 \leq \epsilon \leq 0.05$ asymptotically we have $2.5 \leq c \leq 5$ for $1.7 \leq g \leq 2.7$. Structure of this simple approach does not allow us to observe a single value c, against fixed ϵ. However, we observed that value of g when b is somewhere between 1.3 and 1.5 reveals the best c. For detailed discussion on the results and comparative performance of different rules that we used to estimate d, we refer to section 7 of Paper-I.

For the LP implementation, we used GLPK and run it for different parameter combinations. The results suggest that always some g around 2 gives optimal results. For some of the LP results we refer to the Table 2.1. GLPK could not handle higher number of variables $n2^L$ arising when $n > 32$ and stopped us to go to the limits. But some experiments with another optimization formulation of the problem confirms that LP results are already nearly optimal.

2.2.5 Asymptotic Competitive Ratio

Our estimate c is independent of d and only depends on ϵ and g. However, if we fix both ϵ and g, we observe that c also increases with n. Our LP formulation works for a given number n, and we could test up to $n = 32$. We were unable to find the upper bound for c when $n \to \infty$ using the

LP approach because of the exponentially increasing number of variables for increasing n. However, in our randomized approach, the specific way of choosing pool elements together with some hints that we learned from the lower bound proof of $\Omega(\log n)$ for this problem, guided us to make progress towards obtaining asymptotic value for c.

To calculate the actual limit, we formulated the problem of finding optimal competitive ratio as another nonlinear constraint optimization problem. The new problem is actually an "infinite extension" of our 1-stage randomized estimator, but uses an alternative tool to solve. Still it follows the same idea of translation invariant query points. According to the lower bound proof, we actually have $L = \frac{\ln n}{\ln c}$, which means $\frac{\ln n}{L}$ must be a constant, say u. We call this constant u the query *density*, i.e., the number of queries per length unit on the logarithmic axis. We can have translation invariant query points as: fix u, place queries at points $t = ju + v$, where j being integer $-\infty < j < \infty$, v a random shift such that $0 \leq v < u$. We use infinitely many query points just to obtain upper bounds on c for $n \to \infty$. Nevertheless, their total influence is bounded and the result is an upper bound for our randomized estimate. Of course above is a high-level description and the details can be found in Paper-III.

Now, to find optimal value of c, let $g = \frac{\ln 2}{u}$. Using $g \log n$ queries, we solve the optimization problem of minimizing c using Matlab. For instance, fixing $\epsilon = 0.01$, for very large n, asymptotically we achieved $c = 3.69$ when $g = 1.5$ and similarly $c = 2.99$ when $g = 2$. Combining our estimate with the best result of $1.9d \log n$ for the 2-stage strategy [3] working for known d, the resultant 3-stage competitive group testing has competitive ratio 7.68 for $d = 1$, while it tends to 5.68 asymptotically for growing d. Although the constant factor for the 2-stage strategy is 1.9, we remind that instead of the actual defective number, here in the query number $1.9d \log n$, d actually refers to the assumed upper bound. On the contrary, when we combine it with our estimate for d, now in our result, e.g., $5.68d \log n$, d refers to the actual number of defectives.

2.3 Asymptotically Query-Optimal Strategies

In Paper-III we have used randomization and present different strategies which asymptotically get closer to the entropy lower bound when the defectives grow within certain limits with respect to the problem size. We discuss two cases and explain them for known d or defective rate, but we can easily extend them for the unknown case using our randomized pooling design for estimation prior to these strategies.

2.3.1 Defectives Growing Slowly with the Problem Size

According to the best known deterministic 2-stage strategy using $O(d \log n)$ queries for known d, we know that the hidden constant factor is 1.9 for all d and tends to 1.44 for growing d [3]. Recently in a lower bound proof [23] for 2-stage strategies which insist on $O(d \log n)$ queries, it is shown that when $d = n^{\delta}$, $\delta < 1$, then the hidden constant factor in the query number is strictly greater than 1. They have actually shown it for the statistical model of GT with independent random defectives, but asymptotically it extends to the combinatorial model as well.

Above mentioned lower bound says that we can not have a query-optimal strategy for the GT problem in 2 stages. However, we have shown that when defectives grow at a very slow rate than the problem size (e.g., polylogarithmic), we can have a 2-stage strategy that succeeds with probability $1 - \epsilon$ for known d with a constant factor of 1 in the leading term $d \log n$ subject to some minor terms which depend on d and ϵ only. Thus, asymptotically our results achieve the information-theoretic lower bound. We refer to Theorems 3 and 6 and related Lemmas discussed in section 2 of Paper-III for the exact statements and technical details. Here, we describe some important aspects of our strategy.

Outline of the Strategy: Given n and d, we use only $O(d \log d)+O(d \log(\frac{1}{\epsilon}))$ queries in stage 1 to divide the elements with probability $1 - \epsilon$ into disjoint subsets called *cells* such that each cell contains at most one defective. Like pools, we discriminate between positive and negative cells. Now, a pooling design constructed over these cells (instead of the individual elements) identifies up to d positive cells. Then, in stage 2, we can find the individual defective elements by asking $\log n$ nonadaptive queries to each of the positive cells in parallel. In stage 1, we make q cells and choose q large enough so that the separation of defectives into the cells works with the required probability. Thus, in total our strategy requires $d \log n$ queries plus the query complexity for stage 1.

We discuss stage 1 in detail. To make q cells, we select elements for every cell uniformly at random and independent of each other. This implies that a defective element is put in any cell with probability $\frac{1}{q}$. We define *collision* be the event C_{ij} such that an unordered pair of defectives elements i and j, $(i \neq j)$ is present in the same cell. Since elements are assigned to cells equally likely and the choice of a cell for a defective element is independent of whether the cell already has a defective or not, the collision probability $Pr(C_{ij})$ is simply $\frac{1}{q}$. Because we have d defectives, there are $\binom{d}{2}$ possible distinct pairs of defectives. We want to choose minimum number q such

that any collision occurs with probability at most ϵ_1. Collision events are not independent. Even so, we can apply union bound to bound the collision probability as follows. $Pr[\bigcup_{i\neq j} C_{ij}] \leq \binom{d}{2}\frac{1}{q} \leq \frac{d^2}{2q} \leq \epsilon_1$. Choosing $q = \frac{d^2}{2\epsilon_1}$ we get $\log q = 2\log d + \log(\frac{1}{\epsilon_1}) - 1$.

As described in Theorem 10 of [3], there is a probabilistic pooling design with success probability at least $1 - \epsilon_2$, that correctly identifies up to d defectives from n elements using $O(d(\log n + \log \frac{1}{\epsilon_2}))$ queries. We can use their result as a black box with parameters d and q, such that each pool in their design is actually constructed over the cells rather than individual elements. Overall we get an incorrect result with probability at most $\epsilon := \epsilon_1 + \epsilon_2$. Now, using standard calculations we find that $\log(1/\epsilon_1) + \log(1/\epsilon_2)$ under the constraint $\epsilon = \epsilon_1 + \epsilon_2$ is minimum when $\epsilon_1 = \epsilon_2 = \epsilon/2$. This concludes the stage 1 of our strategy and we use at most $O(d(\log d + \log \frac{1}{\epsilon}))$ queries.

2.3.2 Defectives Growing at a Constant Rate

Now we consider a strategy for the statistical model of group testing and try to get an expected query number as close as possible to the entropy lower bound. The previous result for 2-stage strategy holds asymptotically when d grows much slower than n whereas here we are interested in studying the case when defectives grow at a constant rate. We denote the defective rate with r. As per the entropy lower bound discussed in previous chapter under the section Statistical Model, for small r we need $\log(\frac{1}{r}) + \log e$ expected queries per defective. In Paper-III, we have shown that we can solve the group testing problem in 4 stages using $(1 + o(1))\log(1/r)$ queries per defective. The term $o(1)$ vanishes for $r \to 0$. Actually, we have first reviewed results for adaptive strategies for the fixed rate r, and already observed that the $o(1)$ term cannot be avoided.

For details and other technicalities, we refer to the section 4 of Paper-III. Here we just want to emphasis that result do not follow from the 2-stage strategy discuss in the previous section. Remember the lower order terms depending only on d, that we get as the query number of stage 1 there. In that setting these terms although monotone in d, do not affect the asymptotic behaviour of the strategy because we have only considered the situation when there are a few defectives while problem size is huge and the defectives grow at a much smaller rate compared to the problem size. On the contrary, now considering a fixed defective rate r, defectives will grow with the problem size which means we cannot simply ignore terms which grow unbounded with the defective number. Therefore, we need more stages to avoid them.

Chapter 3

Future Plans

In this chapter, we will briefly outline need and road map of possible direction for future work.

3.1 Need

From the previous discussions, we realize that optimally attainable bounds differ based on the testing method and construction of the pools which may be deterministic or randomized. Results also differ when a strategy assumes that an upper bound on number d is known while others aim for unknown d. We have also discussed scenarios when, e.g., allowing for a small prescribed failure probability ϵ, randomized strategies guarantee constant number of queries and stages which otherwise cannot be achieved using deterministic constructions. There are also differences due to the growth rate of defectives. For example, in the previous section we observed that a 2-stage randomized strategy with prescribed success probability can asymptotically achieve the entropy lower bound for $d \ll n$ and growing much slower than n, whereas we can approach the entropy lower bound in 4 stages when $d = o(n)$.

Thus in practice, it becomes difficult to decide which group testing results from the literature are suitable for the particular needs of a given group testing problem instance. To address this problem, we first need to characterize the available results with respect to:

- **Design Differences**: We differentiate between deterministic and randomized strategies. A strategy can be adaptive, nonadaptive or multistage. In multistage, we further distinguish between strategies which work in a fixed number s of stages versus strategies with expected number of stages. Design difference also incorporates the case of known versus unknown number of defectives (or rate of defectives).

- **Reliability Demands**: We consider whether query complexity of a strategy refers to guaranteed or expected upper bound. We need not only the asymptotic results, but here we also require the exact expressions for the query number including the minor terms which are normally neglected during asymptotic analysis. In this way, we will be able to compare them based on exact query number arising from these expressions for specific input parameters. With respect to output reliability, we study whether the output is always correct, or probably correct according to a fixed probability of success. A probably correct strategy may verify its result and report its correctness, obviously at the cost of more queries. In case of failure, one may choose to run it until a correct and verified output is achieved. Similarly, reliability issues also arise due to testing errors. Therefore, complexity bounds are also different when tests err within certain limits.

From the study of established results for the group testing problem, our

objective is to sort them out in the light of above points and organize the results in the form of an easily accessible knowledge base. Later, one can access this knowledge base and figure out which group testing strategy can efficiently solve a given problem instance. Our special concern is to focus at the problem size, i.e., to find out optimal strategies for fixed values of n and d. Most of the known results primarily present asymptotic query complexity. They do not say much about the exact query number for fixed problem size, especially for small n and d and we may need to develop new strategies for these particular situations.

3.2 Road Map

As first steps, to capture the variations in terms of design differences and reliability constraints, we have proposed a classification of search strategies in Paper-I. There we have specified problem constraints in terms of *queries*, *stages* and *output reliability*. At the moment we have not considered the noisy group testing, i.e., when tests are not reliable and err with some fixed probability. Later on, we plan to develop appropriate classification for this case also.

For a group testing problem instance at hand, we can proceed as follows. At first, we determine input parameters. Generally input refers to a set X of n elements. Here it is sufficient to fix an exact value of n. Next and very important input parameter is the number of defectives d or rate r of defectives. We will discuss different models depending on the available information about d or r. Naturally, one model will be for the case of known d or r. Usually d is unknown and we may only have expected upper bound for d. Similarly we always do not know exact r. Thus another model could be for the cases when we have a promised bound for d or r. Here a promised bound means that in reality the actual number may be more than what is initially given. For this situation, we will discuss whether a GT strategy can actually recognize this fact and still find all defectives or not. Once, input characteristics are specified, we can further move towards the problem specific design and reliability constraints and narrow down a strategy with the best matching demands.

Bibliography

[1] A. Bar-Noy. A new competitive algorithm for group testing. *Discrete Applied Mathematics*, 52(1):29–38, 1994.

[2] H. Chen and F.K. Hwang. Exploring the missing link among d-separable, $\bar{d}$-separable and d-disjunct matrices. *Discrete Applied Mathematics*, 155(5):662–664, 2007.

[3] Y. Cheng and D.Z. Du. New constructions of one- and two-stage pooling designs. *Journal of Computational Biology*, 15(2):195–205, 2008.

[4] M. Cheraghchi. Noise-resilient group testing: Limitations and constructions. In *Proceedings of the seventeenth International Symposium on Fundamentals of Computation Theory FCT'09*, volume 5699 of *LNCS*, pages 62–73. Springer Berlin / Heidelberg, 2009.

[5] P. Damaschke and A. Sheikh Muhammad. Bounds for nonadaptive group tests to estimate the amount of defectives. In *Proceedings of the fourth International Conference on Combinatorial Optimization and Applications COCOA'10*, volume 6509 of *LNCS*, pages 117–130. Springer Berlin / Heidelberg, 2010.

[6] P. Damaschke and A. Sheikh Muhammad. Competitive Group Testing and Learning Hidden Vertex Covers with Minimum Adaptivity. *Discrete Mathematics, Algorithms and Applications*, 2(3):291–311, 2010.

[7] A. De Bonis, L. Gasieniec, and U. Vaccaro. Optimal Two-Stage Algorithms for Group Testing Problems. *SIAM Journal on Computing*, 34(5):1253–1270, 2005.

[8] R. Dorfman. The Detection of Defective Members of Large Populations. *The Annals of Mathematical Statistics*, 14(4):436–440, 1943.

[9] D.Z. Du. Competitive group testing. *Discrete Applied Mathematics*, 45(3):221–232, 1993.

[10] D.Z. Du and F.K. Hwang. Minimizing a combinatorial function. *SIAM Journal on Algebraic and Discrete Methods*, 3(4):523–528, 1982.

[11] D.Z. Du and F.K. Hwang. *Combinatorial group testing and its applications*. World Scientific Pub. Co. Inc., 2000.

[12] D.Z. Du and F.K. Hwang. *Pooling Designs and Nonadaptive Group Testing: Important Tools for DNA Sequencing*, volume 18. World Scientific Pub. Co. Inc., 2006.

[13] D.Z. Du and H. Park. On competitive group testing. *SIAM Journal on Computing*, 23(5):1019–1025, 1994.

[14] D.Z. Du, G.L. Xue, S.Z. Sun, and S.W. Cheng. Modifications of Competitive Group Testing. *SIAM Journal on Computing*, 23(1):82–96, 1994.

[15] D. Eppstein, M.T. Goodrich, and D.S. Hirschberg. Improved Combinatorial Group Testing Algorithms for Real-World Problem Sizes. *SIAM Journal on Computing*, 36(5):1360–1375, 2007.

[16] P. Fischer. On the cut-off point for combinatorial group testing. *Discrete Applied Mathematics*, 91(1-3):83–92, 1999.

[17] F.K. Hwang. A method for detecting all defective members in a population by group testing. *Journal of the American Statistical Association*, 67(339):605–608, 1972.

[18] F.K. Hwang, T.T. Song, and D.Z. Du. Hypergeometric and Generalized Hypergeometric Group Testing. *SIAM Journal on Algebraic and Discrete Methods*, 2(4):426–428, 1981.

[19] A.B. Kahng and S. Reda. New and improved BIST diagnosis methods from combinatorial Group testing theory. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(3):533–543, 2006.

[20] E. Knill. Lower bounds for identifying subset members with subset queries. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms SODA '95*, pages 369–377. Society for Industrial and Applied Mathematics, 1995.

[21] C.H. Li. A sequential method for screening experimental variables. *Journal of the American Statistical Association*, 57(298):455–477, 1962.

[22] M. Manasse, L. McGeoch, and D. Sleator. Competitive algorithms for on-line problems. In *Proceedings of the twentieth annual ACM symposium on Theory of computing STOC '88*, pages 322–333. ACM, 1988.

[23] M. Mézard and C. Toninelli. Group Testing With Random Pools: Optimal Two-Stage Algorithms. *IEEE Transactions on Information Theory*, 57(3):1736–1745, 2011.

[24] J. Schlaghoff and E. Triesch. Improved Results for Competitive Group Testing. *Combinatorics, Probability and Computing*, 14(1):191–202, 2005.

[25] P. Ungar. The cutoff point for group testing. *Communications on Pure and Applied Mathematics*, 13(1):49–54, 1960.

Part II
Publications

Paper-I

Competitive Group Testing and Learning Hidden Vertex Covers with Minimum Adaptivity[1]

Peter Damaschke and Azam Sheikh Muhammad

In *Discrete Mathematics, Algorithms and Applications*
Vol. 2, No. 3 (2010) 291–311

[1]This is an extended version of a paper presented at the *17th International Symposium on Fundamentals of Computation Theory FCT 2009*, Wroclaw, *Lecture Notes in Computer Science* (Springer) 5699, pages 84-95.

Paper-II

Bounds for Nonadaptive Group Tests to Estimate the Amount of Defectives[1]

Peter Damaschke and Azam Sheikh Muhammad

Submitted to
Discrete Mathematics, Algorithms and Applications

[1]This is an extended version of a paper that appeared in preliminary form in the Proceedings of the *4th International Conference on Combinatorial Optimization and Applications COCOA 2010*, Big Island, Hawaii, *Lecture Notes in Computer Science* (Springer) 6509, pages 117-130.

Paper-III

Randomized Group Testing Both Query-Optimal And Minimal Adaptive

Peter Damaschke and Azam Sheikh Muhammad

Submitted