ABSTRACT

Title of dissertation: GROUP TESTING IN STRUCTURED AND DYNAMIC NETWORKS

Batuhan Arasli, Doctor of Philosophy, 2023

Dissertation directed by: Professor Sennur Ulukus
Department of Electrical and Computer Engineering

We consider efficient infection identification algorithms based on group testing under the structured disease spread network and dynamically evolving disease spread network assumptions. Group testing is an efficient infection identification approach based on the idea of pooling the test samples. Group testing has been widely studied in various areas, such as screening and biology, communications, networks, data science, and information theory. In this dissertation, we study group testing applications over structured and dynamic networks, such as random graph-governed correlated connections of nodes and dynamically evolving network topologies under discrete time.

First, we propose a novel infection spread model based on a random graph representing connections between n individuals. The infection spreads via connections between individuals, resulting in a probabilistic cluster formation structure as well as non-i.i.d. (correlated) infection statuses for individuals. We propose a class of *two-step sampled group testing algorithms* where we exploit the known probabilistic

infection spread model. We investigate the metrics associated with two-step sampled group testing algorithms. To demonstrate our results for analytically tractable *exponentially split cluster formation trees*, we calculate the required number of tests and the expected number of false classifications in terms of the system parameters and identify the trade-off between them. For exponentially split cluster formation trees, for zero-error construction, we prove that the required number of tests is $O(\log_2 n)$. Thus, for such cluster formation trees, our algorithm outperforms any zero-error non-adaptive group test, binary splitting algorithm, and Hwang's generalized binary splitting algorithm. Our results imply that, by exploiting probabilistic information on the connections of individuals, group testing can be used to reduce the number of required tests significantly even when the infection rate is high, contrasting the prevalent belief that group testing is useful only when the infection rate is low.

Next, we study a dynamic infection spread model inspired by the discrete time SIR (susceptible-infected-recovered) model, where infections are spread via non-isolated infected individuals; while infection keeps spreading over time, limited capacity testing is performed at each time instant as well. In contrast to the classical, static group testing problem, the objective in our setup is not to find the minimum number of required tests to identify the infection status of every individual in the population but to *control* the infection spread by detecting and isolating the infections over time by using the given, limited number of tests. To analyze the performance of the proposed algorithms, we focus on the average-case analysis of the number of individuals that remain non-infected throughout the process of

controlling the infection. We propose two dynamic algorithms that both use a given limited number of tests to identify and isolate the infections over time while the infection spreads. The first algorithm is a dynamic randomized individual testing algorithm; in the second algorithm, we employ the group testing approach similar to the original work of Dorfman. By considering weak versions of our algorithms, we obtain lower bounds for the performance of our algorithms. Finally, we implement our algorithms and run simulations to gather numerical results and compare our algorithms and theoretical approximation results under different sets of system parameters.

Finally, we consider the dynamic infection spread model based on the discrete SIR model, which assumes the disease to be spread over time via infected and non-isolated individuals. In our system, the main objective is not to minimize the number of required tests to identify every infection but instead to utilize the available, given testing capacity T at each time instant to efficiently control the infection spread. We introduce and study a novel performance metric, which we coin as ϵ-disease control time. This metric can be used to measure how fast a given algorithm can control the spread of a disease. We characterize the performance of the dynamic individual testing algorithm and introduce a novel dynamic SAFFRON-based group testing algorithm. We present theoretical results and implement the proposed algorithms to compare their performances.

GROUP TESTING IN STRUCTURED AND DYNAMIC NETWORKS

by

Batuhan Arasli

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor Sennur Ulukus, Chair/Advisor
Professor Behtash Babadi
Professor Richard La
Professor Adrian Papamarcou
Professor Thomas Goldstein

Dedication

This dissertation is dedicated to my loving wife Özde and my parents Özlem and Yılmaz for always believing in me.

Acknowledgments

I am deeply grateful to my advisor, Professor Sennur Ulukus, for her guidance, support, and encouragement throughout my Ph.D. journey. Her unwavering support has been a constant source of motivation. I am honored to have had the opportunity to work with such a brilliant and dedicated mentor. I would also like to thank her for her patience and understanding. This dissertation would not have been possible without her support and guidance.

I would like to extend my heartfelt gratitude to Professors Behtash Babadi, Richard La, Adrian Papamarcou, and Thomas Goldstein for being on my dissertation committee and for their time, support, and valuable feedback. I am extremely grateful for the time, expertise, and insights that each of them has provided. I am also thankful to every professor I have interacted with throughout my Ph.D. journey at the University of Maryland. In particular, I would like to thank Professor Nuno Martins and Adrian Papamarcou since I have learned much while working with them as their teaching assistant. I would also like to thank Professor Alexander Barg. His teachings in the three classes I took from him have been instrumental in my academic and professional development.

I am thankful to Tolga Mete Duman of Bilkent University. In the last year of my undergraduate studies at Bilkent University, I had the great opportunity to do research under his supervision which sparked my interest in information theory, which eventually led me to do information theory research throughout my Ph.D., which resulted in this dissertation.

I am lucky to have shared the lab space with brilliant people who made the lab a friendly place to work. Many thanks to my lab mates Sajani Vithana, Zhusheng Wang, Priyanka Kaswan, Matin Mortaheb, Cemil Vahapoglu, Purbesh Mitra, Mustafa Doger, Subhankar Banerjee, Shreya Meel, Alptug Aytekin, Sahan Liyanaarachchi, Mohamed Nomeir, Arunabh Srivastava, Brian Kim, Baturalp Buyukates, Melih Bastopcu, Yi-Peng Wei, Karim Banawan, Ajaykrishnan Nageswaran, and Sagnik Bhattacharya. I am also thankful to Karim Banawan and Yi-Peng Wei for their guidance and our collaborations in the first year of my Ph.D.; I feel lucky to collaborate with them in my very first year at the University of Maryland.

I would like to sincerely thank my dear friends Baturalp Buyukates, Melih Bastopcu, Semih Kara, and Ece Yegane. I am so thankful that I have shared my time in my Ph.D. journey with them. They have been like a family to me throughout this journey. I will be missing all of our memories together (including the delicious and extra cheesy meals that we cooked together with my dear roommates Baturalp and Melih, and the moment when Semih, Ece and I realized the absurdity of doing research in a basement). I am also grateful to Faizan Tariq for being a great friend throughout this journey.

I am thankful to other friends at the University of Maryland for making these years enjoyable. I also thank Dogan Kutay Pekcan and Can Ozgurel for their friendship and for always being there for me, even when we are thousands of miles away.

Last but not least, I am thankful and grateful to my family. I still remember the first day at school as a first grader, when I officially started this long journey that led me to this day. My father, Yilmaz Arasli, and my mother Ozlem Arasli have

never stopped believing in me since that day which was almost twenty-one years ago. I am thankful to them and also to my brother, Dogukan Arasli, for always making me laugh whenever we are together. I am most grateful to my loving, caring, and beautiful wife, my soulmate Ozde Ozkaya Arasli. As someone who has recently completed a dissertation that involves a considerable amount of probability theory, I know how arbitrarily small the probability of meeting the one and only soulmate in my life is, but you make me believe in miracles more and more every passing day. I am lucky to have you in my life.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Efficient testing and infection identification are crucial in slowing down and eventually preventing disease spread, especially in the cases of novel pandemics where fully effective vaccines and cures have not been developed. In cases where the disease of interest is highly contagious, such as covid-19, pandemics can affect the lives of billions of people, even under strict measures to slow down the spread of the disease. While controlling the spread of disease, identification of the infection status of the individuals plays a crucial role, which would be helpful to further implement additional measures such as quarantining, which will eventually help to slow down and control the pandemic. Even for diseases that do not result in global pandemics, fast and efficient infection identification can be crucial and life-saving. Motivated by the need for fast and efficient detection of the prevalence of syphilis among the WW2 draftees, Robert Dorfman proposed the novel group testing approach in his seminal paper [1].

The novel idea behind the group testing approach is mixing the test samples of groups of individuals and testing the mixed samples rather than testing each

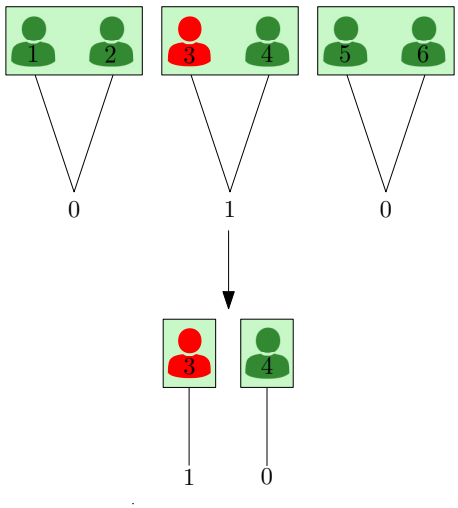


Figure 1.1: Infection status identification for a group of 6 people via a 2-stage group testing algorithm.

of the samples individually. When the practical constraints of the testing method and the infection of interest allow, group testing results in a problem where the infection statuses of the individuals are combined by binary-OR operation, and the main objective becomes designing groups of individuals (group tests) and decoding the group test results to identify the infection statuses. For instance, in the seminal work of Dorfman in [1], the proposed algorithm assigns disjoint groups of individuals uniformly randomly and mixes the samples within groups. After testing the mixed samples, if a mixed sample is negative, then every sample mixed in that mixed sample is negative. On the other hand, if a mixed sample is positive, it implies that there is at least one positive sample among the samples mixed in that mixed sample. In the second round, every individual in the positive groups is tested individually to identify positive groups. Especially when the prevalence rate of the infection is low among the population, this group testing procedure results in a significant reduction in the number of performed tests to identify the infection statuses of the

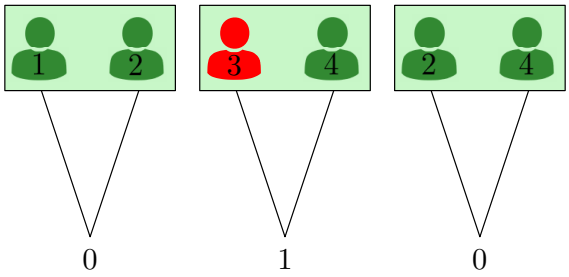


Figure 1.2: Infection status identification for a group of 4 people with a single stage group testing algorithm.

individuals [1].

The groups for the testing can be disjoint, or a sample from an individual can be mixed into multiple mixed samples. Furthermore, testing can be performed in a single stage (non-adaptive group testing), or multiple stages of testing can be performed (adaptive group testing) where tests can be designed by using the results of prior stages. In Fig. 1.1, we present a toy example where the infection statuses of 6 individuals are identified via the group testing algorithm proposed in the seminal paper of Dorfman. Individual 3 (marked with red) is the only positive among these six individuals. Notice that it is a two-stage group testing algorithm, where three tests are performed in the first stage, and two tests are performed in the second stage. At the end of the first stage, the infection statuses of four individuals (individuals 1,2, 5, and 6) are identified. At the end of the second stage, the infection statuses of the remaining two individuals are identified. In Fig. 1.2, we present the identification of the infection statuses of four individuals by performing three group tests in a single stage. Notice that the samples of individuals 2 and 4 are mixed in two different mixed samples. Negative tests directly identify the negative infection status of the samples that are mixed in them, while the positive test implies that

individual 3 is positive since individual 4 is identified as negative already. In both of these examples, the total numbers of tests required to identify the infection statuses of the individuals are strictly less than individual testing: five and three rather than six and four, respectively. The overall benefit of the group testing can be observed better with an increasing total number of individuals, as long as the prevalence rate is low [2].

In the following, we briefly review the developing literature on group testing; a detailed survey can be found in [2]. Following the seminal work of Dorfman, adaptive algorithms [3–9] and non-adaptive algorithms [10–22] have been proposed and performance guarantees have been characterized. The capacity of the group testing problem has been studied in [14, 23–30] under various system models. The resemblance of the group testing problem with the multiaccess communication first stated by [31] and with the compressed sensing problem first studied in [32]. The standard system models have been challenged, and a variety of models have been studied: References [16, 32–39] study noisy group testing problem where the test results are noisy, references [40–44] focus on limited adaptive testing stages for group testing algorithm designs, references [20, 21, 33, 45–53] investigate practical decoding times as well as explicit and graph constrained algorithm designs. Group testing has found various applications in distinct fields, such as communications literature [54–63], network literature with fault detection in networks with specific focuses on sensor networks [64–69], data science literature with applications for learning and searching [70–75], data storage and compression [76, 77], cyber-security [78, 79], databases [80], theoretical computer science [81–85], electronics [86, 87], and so on.

Common to the majority of the standard group testing works is the observation that the group testing is beneficial only when the prevalence rate of the infection is low among the population [1, 2, 7–9, 11, 12, 14, 23, 24, 30, 32]. However, this limitation is characterized under standard system models, i.e., standard combinatorial and probabilistic settings. In the combinatorial setting, a fixed number of infections (e.g., d infections) are assumed to be prevalent in the system, and the infected set of individuals are uniformly randomly realized out of all possible d-sized subsets of individuals. On the other hand, in the standard probabilistic setting, each individual is assumed to be independently infected with a given infection probability p. In practice, even though these standard models are proven to be useful for some applications, for many applications, especially in contagious diseases, infection statuses of the individuals are rarely independent and identical as in the probabilistic setting, or the true number of infections in the system is rarely known, and the true infected set is rarely uniformly distributed. In a more recent line of works, these standard models have been challenged: in references [88–92] non-i.i.d. probabilistic models are investigated, in [93–97], community structure based disease spread models are studied, in references [98–100] benefits of structured side information in group testing is the main focus, in references [101–105] dynamically evolving network-based group testing and disease spread controlling are investigated.

In this dissertation, our goal is to analyze group testing under novel structured and dynamic networks. Motivated by the fact that further practically available side information can be utilized while designing group testing algorithms to reduce the required number of tests further to identify the infection statuses of groups of

individuals, we propose and analyze structured side information aided systems that correspond to community networks of the tested individuals. Another goal of this dissertation is to investigate dynamic system models based on dynamically evolving networks of individuals with non-static infection statuses, motivated by the goal of helping group testing to be efficiently employed to help control the next pandemic.

In Chapter 2, we introduce a novel, random graph-based community-structured infection spread model, where nodes represent individuals and possible infection transmissions between individuals are represented by random edges between the nodes. The probability distribution of edge realizations is assumed to be known, and a realization of the random graph results in clusters in the population. A random patient zero introduces the infection to the population by infecting everyone in their cluster. Utilizing the probability distribution of the connections and cluster formations in the random graph, we propose a novel family of algorithms: *two-step sampled group testing algorithms*, which consists of sampling a subset of individuals and performing non-adaptive tests to identify the selected individuals with zero-error. For this second step of two-step sampled group testing algorithms, we introduce $\mathcal{F}$-separable zero-error non-adaptive test matrix designs. We characterize the optimal design of two-step sampled group testing algorithms and derive explicit results for the *exponentially split cluster formation tree* structures that we introduce. We characterize the trade-off between the expected number of false classifications and the required number of tests for different choices of possible sampling cluster formations for the two-step sampled group testing algorithms. We analyze the computational complexity of two-step sampled group testing algorithms. For

zero-error construction, we prove that the required number of tests for the identification is less than $4(\log_2 n + 1)/3)$ and is $O(\log_2 n)$ in a system that consists of at most n equal-sized clusters. Even when we ignore the cluster size gain, we show that our algorithm outperforms the optimal adaptive algorithms that assume the known number of infections, such as Hwang's generalized binary splitting algorithm, including the regimes where the infection rate is high. With this, we show that, with additional side information such as random graph-governed community structures, group testing can be used efficiently even when the infection rate is high and significant improvements over individual testing can be observed by utilizing available side information.

In Chapter 3, we study a discrete time SIR model-based dynamic infection spread model. We consider a population of n individuals divided into three disjoint subsets: susceptible individuals, non-isolated infections, and isolated/recovered individuals. In the beginning, $t = 0$, the infection is introduced to the system, where each individual gets infected with probability p independently. At $t = 0$, the infection model is identical to the standard i.i.d. probabilistic model. At each discrete time instant after that, $t \geq 1$, we consider a cycle of infection spread, testing, and isolation of the detected infections. At each time instant, the infection is spread from non-isolated infections to the susceptible individuals, independently with probability q for each susceptible-infection pair. Then, group testing follows, where only a given, limited number of T tests are performed. Depending on the test results, detected infections are isolated and cannot spread the infection to susceptible individuals at the times that follow their isolation. Eventually, they recover, and we assume they

do not get infected throughout the rest of the process. Here, similar to the real-life scenarios, we consider a dynamically changing system and limited testing capacity at each time instant rather than minimizing the total number of required tests to identify everyone as in the static group testing problem. The performance metrics of dynamic testing algorithms in such a system are the time when the infection spread is brought under control, i.e., when all the infections are detected and isolated, and the number of susceptible individuals when the infection is brought under control. In this work, we analyze the average-case performance of the system. We derive probabilistic results for the random processes of the number of susceptible individuals, non-isolated infections, and isolated/recovered individuals for *symmetric and converging algorithms*. We propose two dynamic algorithms: dynamic individual testing and dynamic Dorfman-type group testing algorithm. We consider the weak versions of these algorithms and use our general results to derive performance lower bounds. We obtain simulation results to compare our theoretical approximation results with the numerical results for our proposed algorithms in different parameter regimes.

In Chapter 4, we further expand the dynamic disease spread model introduced in Chapter 3. We introduce two novel performance metrics: disease control time, $\bar{t}$, and ϵ-disease control time $\bar{t}_\epsilon$. These performance metrics add a novel dimension to the discrete-time SIR-based dynamic system model that we introduce in Chapter 3: to assess the performance of proposed dynamic algorithms in terms of how fast the disease spread is controlled, one can use these novel performance metrics. Moreover, we propose a novel dynamic group testing algorithm: dynamic SAFFRON-based

group testing algorithm. We analyze the performance of the dynamic individual testing algorithm and dynamic SAFFRON-based group testing algorithm in terms of the novel performance metrics we introduce. We obtain simulation results to analyze proposed dynamic group testing algorithms numerically.

In Chapter 5, we present the conclusions of this dissertation.

CHAPTER 2

Group Testing with a Graph Infection Spread Model

2.1 Introduction

In this chapter, we propose a novel infection spread model, where individuals are connected via a random connection graph, whose connection probabilities are known[1]. A realization of the random connection graph results in different connected components, i.e., clusters, and partitions the set of all individuals. The infection starts with a patient zero who is uniformly randomly chosen among n individuals. Then, any individual who is connected to at least one infected individual is also infected. For this system model, we propose a novel family of algorithms which we coin *two-step sampled group testing algorithms*. The algorithm consists of a sampling step, where a set of individuals are chosen to be tested, and a zero-error non-adaptive test step, where selected individuals are tested according to a zero-error non-adaptive group test matrix. In order to select individuals to test in the first step, one of the possible cluster formations that can be formed in the random connection graph is selected. Then, according to the selected cluster formation, we select exactly one individual

[1]For instance, location data obtained from cell phones can be used to estimate connection probabilities.

from every cluster. After identifying the infection status of the selected individuals with zero error, we assign the same infection status to the other individuals in the same cluster as identified individuals. Note that, the actual cluster formation is not known prior to the test design, and because of that, the selected cluster formation can be different from the actual cluster formation. Thus, this process is not necessarily a zero-error group testing procedure.

Our main contributions consist of proposing a novel infection spread model with a random connection graph, proposing a two-step sampled group testing algorithm which is based on novel $\mathcal{F}$-separable zero-error non-adaptive test matrices, characterizing the optimal design of two-step sampled group testing algorithms, and presenting explicit results on analytically tractable *exponentially split cluster formation trees*. For the considered two-step sampled group testing algorithms, we identify the optimal sampling function selection, calculate the required number of tests and the expected number of false classifications in terms of the system parameters, and identify the trade-off between them. Our $\mathcal{F}$-separable zero-error non-adaptive test matrix construction is based on taking advantage of the known probability distribution of cluster formations. In order to present an analytically tractable case study for our proposed two-step sampled group testing algorithm, we consider exponentially split cluster formation trees as a special case, in which we explicitly calculate the required number of tests and the expected number of false classifications. For zero-error construction, we prove that the required number of tests is less than $4(\log_2 n + 1)/3$ and is of $O(\log_2 n)$, when there are at most n equal-sized clusters in the system, each having δ individuals. For the sake of fairness, in our compar-

isons, we take δ to be 1, ignoring further reductions of the number of tests due to δ. We show that, even when we ignore the gain by cluster size δ, our non-adaptive algorithm, in the zero-error setting, outperforms any zero-error non-adaptive group test and Hwang's generalized binary splitting algorithm [7], which is known to be the optimal zero-error adaptive group test [2]. Since the number of infections scale as $\frac{n}{\log_2 n}\delta$ in exponentially split cluster formation trees with $n\delta$ individuals, our results show that, we can use group testing to reduce the required number of tests significantly in our system model even when the infection rate is high by using our two-step sampled group testing algorithm.

2.2 System Model

We consider a group of n individuals. The *random infection vector* $U = (U_1, U_2, \ldots, U_n)$ represents the infection status of the individuals. Here U_i is a Bernoulli random variable with parameter p_i. If individual i is infected then $U_i = 1$, otherwise $U_i = 0$. Random variables U_i need not be independent. A *patient zero random variable* Z is uniformly distributed over the set of individuals, i.e., $Z = i$ with probability $p_Z(i) = \frac{1}{n}$ for $i = 1, \ldots, n$. Patient zero is the first person to be infected. So far, the infection model is identical to the traditional combinatorial model with $k = 1$ infected among n individuals.

Next, we define a *random connection graph* $\mathscr{C}$, a random graph where vertices represent the individuals and edges represent the connections between the individuals. Let $p_{\mathscr{C}}$ denote the probability distribution of the random graph $\mathscr{C}$ over the

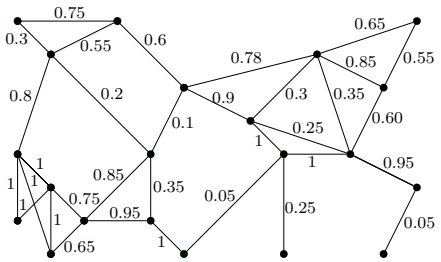
Table 2.1: Nomenclature.

System	
n	number of individuals in the system
U	infection status vector of size n
Z	patient zero random variable
$p_Z(i)$	probability of individual i is the patient zero
$\mathscr{C}$	random connection graph
$E_{\mathscr{C}}$	edge set of $\mathscr{C}$
$V_{\mathscr{C}}$	vertex set of $\mathscr{C}$, also equal to $[n]$
$\boldsymbol{C}$	random connection matrix
F	cluster formation random variable
$\mathcal{F}$	set of all possible cluster formations, i.e., $\{F_i\}$
$p_F(F_i)$	probability of true cluster formation is F_i
f	number of possible cluster formations, i.e., $\lvert\mathcal{F}\rvert$
σ_i	number of clusters in the cluster formation F_i
S_j^i	jth cluster in F_i
λ_j	number of unique clusters in $\mathcal{F}$ at and above the level F_j
$\lambda_{S_i^j}$	number of unique ancestor nodes of S_i^j in $\mathcal{F}$
δ	size of the bottom level clusters in an exponentially split $\mathcal{F}$
Algorithm	
F_m	sampling cluster formation chosen from $\mathcal{F}$
M	sampling function that selects individuals to be tested
$U^{(M)}$	infection status vector of the selected individuals by M
$S^\alpha(M_i)$	the cluster in F_α that contains ith selected individual by M
K_M	set of infections among the selected individuals by M
$\mathcal{P}(K_M)$	set of all possible infected sets that K_M can be
T	number of tests to be performed
$\boldsymbol{X}$	$T \times \sigma_m$ test matrix
$\boldsymbol{X}^{(i)}$	ith column of $\boldsymbol{X}$
y	test result vector of size T
$\hat{U}$	estimated infection status of n individuals after test results
$E_{f,\alpha}$	expected number of false classifications given $F = F_\alpha$
E_f	expected number of false classifications

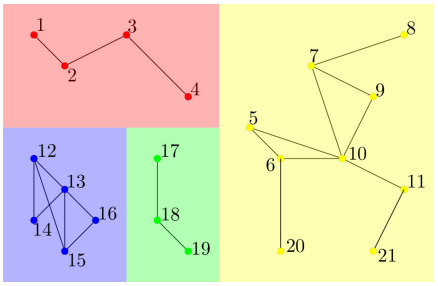
support set of all possible edge realizations. For the special class of random connection graphs where the edges are realized independently, we fully characterize the statistics of the random connection graph by the random connection matrix $\boldsymbol{C}$, which is a symmetric $n \times n$ matrix where the (i,j)th entry C_{ij} is the probability that there is an edge between vertices i and j for $i \neq j$, and $C_{ij} = 0$ for $i = j$ by definition.

A random connection graph $\mathscr{C}$ is an undirected random graph with vertex set $V_{\mathscr{C}} = [n]$, with each vertex representing a unique individual and a random edge set $E_{\mathscr{C}} = \{e_{ij}\}$ which represents connections between individuals, that satisfies the following: 1) If $e_{ij} \in E_{\mathscr{C}}$, then there is an edge between vertices i and j; 2) For an arbitrary edge set $E^*_{\mathscr{C}}$, probability of $E_{\mathscr{C}} = E^*_{\mathscr{C}}$ is equal to $p_{\mathscr{C}}(E^*_{\mathscr{C}}, V_{\mathscr{C}})$. In the case when all $\mathbb{1}_{\{e_{ij} \in E_{\mathscr{C}}\}}$ are independent, where $\mathbb{1}_A$ denotes the indicator function of the event A, the random connection matrix $\boldsymbol{C}$ fully characterizes the statistics of edge realizations. There is a path between vertices i and j if there exists a set of vertices $\{i_1, i_2, \dots i_k\}$ in $[n]$ such that $\{e_{ii_1}, e_{i_1 i_2}, e_{i_2 i_3}, \dots e_{i_k j}\} \subset E_{\mathscr{C}}$, i.e., two vertices are connected if there exists a path between them.

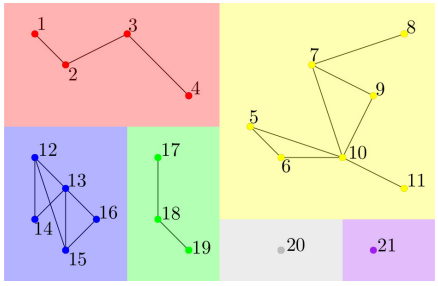
In our system model, if there is a path in $\mathscr{C}$ between two individuals, then their infection statuses are equal. In other words, the infection spreads from patient zero Z to everyone connected to patient zero. Thus, $U_k = U_l$ if there exists a path between k and l in $\mathscr{C}$. Here, we note that a realization of the random graph $\mathscr{C}$ consists of clusters of individuals, where a cluster is a subset of vertices in $\mathscr{C}$ such that all elements in a cluster are connected with each other, and none of them is connected to any vertex that is not in the cluster. More rigorously, a subset



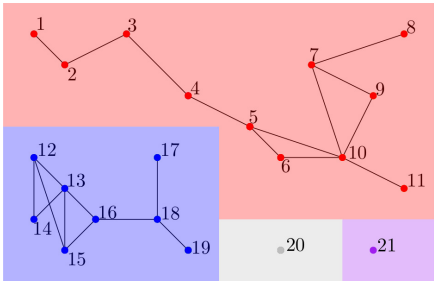
(a) Probabilities of the edges.



(b) In this realization of $\mathscr{C}$, there are 4 clusters.



(c) In this realization of $\mathscr{C}$, there are 6 clusters.



(d) In this realization of $\mathscr{C}$, there are 4 clusters.

Figure 2.1: Random connection graph $\mathscr{C}$ and three possible realizations and cluster formations. We show each cluster with a different color.

$S = \{i_1, i_2, \dots i_k\}$ of $V_{\mathscr{C}}$ is a cluster if, i_l and i_m are connected for all $i_l \neq i_m \in S$, but i_a and i_b are not connected for any $i_a \in S$ and all $i_b \in V_{\mathscr{C}} \backslash S$.

Note that the set of all clusters in a realization of the random graph $\mathscr{C}$ is a partition of $[n]$. In a random connection graph structure, the formation of clusters in $\mathscr{C}$ along with patient zero Z determines the status of the infection vector. Therefore, instead of focusing on the specific structure of the graph $\mathscr{C}$, we focus on the cluster formations in $\mathscr{C}$. For a given $p_{\mathscr{C}}$, we can calculate the probabilities of possible cluster formations in $\mathscr{C}$.

To solidify ideas, we give an example in Figure 2.1. For a random connection graph where the edges are realized independently, we give probabilities of the exis-

tence of edges (zero probabilities are not shown) in Figure 2.1(a) and three different realizations of a random connection graph $\mathscr{C}$, where all three realizations result in different cluster formations in Figure 2.1(b)-(d). In Figure 2.1, we consider a random connection graph $\mathscr{C}$ that has $n = 21$ vertices, which represent the individuals in our group testing model. Since in this example we assume that the edges are realized independently, every edge between vertices i and j exists with probability C_{ij} independently. As we defined, if there is a path between two vertices (i.e., they are in the same cluster), then we say that their infection statuses are the same. One way of interpreting this is, there is a patient zero Z, which is uniformly randomly chosen among n individuals, and patient zero spreads the infection to everyone in its cluster. Therefore, working on the cluster formation structures, rather than the random connection graph itself, is equally informative for the sake of designing group tests. For instance, in the realization that we give in Figure 2.1(b), if the edge between vertices 5 and 10 did not exist, that would be a different realization for the random connection graph $\mathscr{C}$. However, the cluster formations would still be the same. As all infections are determined by the cluster formations and the realization of patient zero, cluster formations are sufficient statistics. Before we rigorously argue this point, we first focus on constructing a basis for random cluster formations.

The *random cluster formation* variable F is distributed over $\mathcal{F}$ as $\mathbb{P}(F = F_i) = p_F(F_i)$, for all $F_i \in \mathcal{F}$, where $\mathcal{F}$ is a subset of the set of all partitions of the set $\{1, 2, \ldots, n\}$. In our model, we know the set $\mathcal{F}$ (i.e., the set of cluster formations that can occur) and the probability distribution p_F, since we know $p_{\mathscr{C}}$. Let us denote

$|\mathcal{F}|$ by f. For a cluster formation F_i, individuals that are in the same cluster have the same infection status. Let $|F_i| = \sigma_i$, i.e., there are σ_i subsets in the partition F_i of $\{1, 2, \ldots, n\}$. Without loss of generality, for $i < j$, we have $\sigma_i \leq \sigma_j$, i.e., cluster formations in $\mathcal{F}$ are ordered in increasing sizes. Let S_j^i be the jth subset of the partition F_i where $i \in [f]$ and $j \in [\sigma_i]$. Then, for fixed i and j, $U_k = U_l$ for all $k, l \in S_j^i$, for all $i \in [f]$ and $j \in [\sigma_i]$.

To clarify the definitions, we give a simple running example which we will refer to throughout this section. Consider a population with $n = 3$ individuals who are connected according to the random connection matrix $\boldsymbol{C}$ and assume that the edges are realized independently,

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0.3 & 0.5 \\ 0.3 & 0 & 0 \\ 0.5 & 0 & 0 \end{bmatrix} \tag{2.1}$$

By definition, the main diagonal of the random connection matrix is zero, since we define edges between distinct vertices only. In this example, $\mathcal{F}$ consists of 4 possible cluster formations, and thus, we have $f = |\mathcal{F}| = 4$. The random cluster formation variable F can take those 4 possible cluster formations with the following

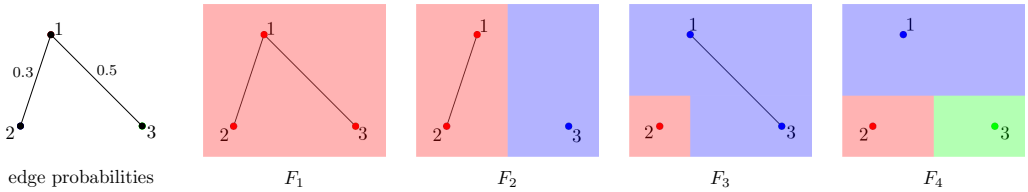


Figure 2.2: Edge probabilities of $\mathscr{C}$ and elements of $\mathcal{F}$ in example $\boldsymbol{C}$ given in (2.1) with clusters shown in different colors.

probabilities,

$$F = \begin{cases} F_1 = \{\{1,2,3\}\}, & w.\ p.\ \ 0.15 \\ F_2 = \{\{1,2\},\{3\}\}, & w.\ p.\ \ 0.15 \\ F_3 = \{\{1,3\},\{2\}\}, & w.\ p.\ \ 0.35 \\ F_4 = \{\{1\},\{2\},\{3\}\}, & w.\ p.\ \ 0.35 \end{cases} \tag{2.2}$$

This example network and the corresponding cluster formations are shown in Figure 2.2. Here, cluster formation F_1 occurs when the edge between vertices 1 and 2 and the edge between vertices 1 and 3 are realized; F_2 occurs when only the edge between vertices 1 and 2 is realized; and F_3 occurs when only the edge between vertices 1 and 3 is realized. Finally, F_4 occurs when none of the edges in $\mathscr{C}$ is realized. In this example, we have $\sigma_1 = |F_1| = 1$, $\sigma_2 = |F_2| = 2$, $\sigma_3 = |F_3| = 2$, and $\sigma_4 = |F_4| = 3$. Note that $\sigma_1 \leq \sigma_2 \leq \sigma_3 \leq \sigma_4$ is assumed without loss of generality above. Each subset that forms the partition F_i are denoted by S_j^i, for instance, F_3 consists of $S_1^3 = \{1,3\}$ and $S_2^3 = \{2\}$.

Next, we argue formally that cluster formations are sufficient statistics, i.e., they represent an equal amount of information as the realization of the random graph as far as the infection statuses of the individuals are concerned. When Z

and F are realized, the infection statuses of n individuals are also realized, i.e., $H(U|Z,F)=0$. Then,

$$\begin{aligned} I(U;F) &= H(U) - H(U|F) && (2.3)\\ &= H(U) - (H(U,Z|F) - H(Z|U,F)) && (2.4)\\ &= H(U) - (H(Z|F) + H(U|Z,F) - H(Z|U,F)) && (2.5)\\ &= H(U) - (H(Z) - H(Z|U,F)) && (2.6)\\ &\geq H(U) - (H(Z|\mathscr{C}) + H(U|Z,\mathscr{C}) - H(Z|U,\mathscr{C})) && (2.7)\\ &= H(U) - H(U|\mathscr{C}) && (2.8)\\ &= I(U;\mathscr{C}) && (2.9) \end{aligned}$$

where in (2.7) we used the fact that F is a function of $\mathscr{C}$ (not necessarily invertible). In addition, from $U \to \mathscr{C} \to F$, we also have $I(U;F) \leq I(U;\mathscr{C})$, which together with (2.9) imply $I(U;F) = I(U;\mathscr{C})$. Thus, F is a sufficient statistic for $\mathscr{C}$ relative to U. Therefore, from this point on, we focus on the random cluster formation variable F in our analysis.

The graph model and the resulting cluster formations we described so far are general. For tractability, in this chapter, we investigate a specific class of $\mathcal{F}$ which satisfies the following condition: For all i, F_i can only be obtained by partitioning some elements of F_{i-1}. This assumption results in a tree-like structure for cluster formations. Thus, we call $\mathcal{F}$ sets that satisfy this condition *cluster formation trees*. Formally, $\mathcal{F}$ is a cluster formation tree if $F_{i+1} \backslash F_i$ can be obtained by partitioning

the elements of $F_i \backslash F_{i+1}$ for all $i \in [f-1]$. Note that $\mathcal{F}$ in (2.2) is not a cluster formation tree. However, if the probability of the edge between vertices 1 and 3 were 0, then $\mathcal{F}$ would not contain F_1 and F_3, and $\mathcal{F}$ would be a cluster formation tree in this case. Note that cluster formation trees may arise in real-life clustering scenarios, for instance, if individuals belong to a hierarchical structure. An example is: an individual may belong to a professor's lab, then to a department, then to a building, then to a campus.

Next, we define the family of algorithms that we consider, which we coin *two-step sampled group testing algorithms*[2]. Two-step sampled group testing algorithms consist of two steps in both the testing and decoding phases. The following definitions are necessary to characterize the family of algorithms that we consider.

To design a two-step sampled group testing algorithm, we first pick one of the cluster formations in $\mathcal{F}$ to be the *sampling cluster formation*. The selection of F_m is a design choice. For example, recalling the running example in (2.1)-(2.2), one can choose F_2 to be the sampling cluster formation.

Next, we define the *sampling function*, M, to be a function of F_m. The sampling function selects which individuals to be tested by selecting exactly one individual from every subset that forms the partition F_m. Let K_M denote the infected set among the sampled individuals. The output of the sampling function M is the individuals that are sampled and going to be tested. In the second step, a zero-error non-adaptive group test is performed on the sampled individuals. This results in

[2]In the two-step sampled group testing algorithms, two steps do not involve consecutive testing phases: the proposed algorithm family in this chapter consist of non-adaptive constructions, and should not be confused with semi-adaptive algorithms with two testing phases such as two-stage algorithm in [93].

identifying the infection status of the selected $\sigma_m = |F_m|$ individuals with zero-error probability. For example, recalling the running example in (2.1)-(2.2), when the sampling cluster formation is chosen as F_2, we may design M as,

$$M = \{1, 3\} \tag{2.10}$$

Note that, for each selection of F_m, M selects exactly one individual from each S_j^m. As long as it satisfies this property, M can be chosen freely while designing the group testing algorithm.

The *test matrix* $\boldsymbol{X}$ is a non-adaptive test matrix of size $T \times \sigma_m$, where T is the required number of tests. Let $U^{(M)}$ denote the infection status vector of the sampled individuals. Then, we have the following test result vector y,

$$y_i = \bigvee_{j \in [\sigma_m]} X_{ij} U_j^{(M)}, \quad i \in [T] \tag{2.11}$$

In the classical group testing applications, while constructing zero-error non-adaptive test matrices, the aim is to obtain unique result vectors, y, for every unique possible infected set and, for instance, in combinatorial setting, with d infections, *d-separable* matrix construction is proposed [17]. In the classical d-separable matrix construction, we have

$$\bigvee_{i \in S_1} \boldsymbol{X}^{(i)} \neq \bigvee_{i \in S_2} \boldsymbol{X}^{(i)} \tag{2.12}$$

for all subsets S_1 and S_2 of cardinality d. As a more general approach, we do

not restrict the possible infected sets to the subsets of $[n]$ of the same size, but we consider the problem of designing test matrices that satisfy (2.12) for every unique S_1 and S_2 in a given set of possible infected sets. This approach leads to a more general basis for designing zero-error non-adaptive group testing algorithms for various scenarios when the available side information can restrict the set of possible infected sets.

Using the test result vector y, in the first decoding step, the infection statuses of the sampled individuals are identified with zero-error probability. In the second stage of decoding, depending on F_m and the infection status of the sampled individuals, other non-tested individuals are estimated by assigning the same infection status to all individuals that share the same cluster in the cluster formation F_m. In the running example, with M given in (2.10), one must design a zero-error non-adaptive test matrix $\boldsymbol{X}$, which identifies the infection status of individuals 1 and 3.

Let $\hat{U} = (\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_n)$ be the *estimated infection status vector*. By definition, the infection estimates are the same within each cluster, i.e., for sampling cluster formation F_m, $\hat{U}_k = \hat{U}_l$, for all $k, l \in S_j^m$, for all $j \in [\sigma_m]$. Since M samples exactly one individual from every subset that forms the partition F_m, there is exactly one identified individual at the beginning of the second step of the decoding phase. By the aforementioned rule, all n individuals have an estimated infection status at the end of the process. For instance, in the running example, for the sampling cluster formation F_2, we have $M = \{1, 3\}$ as given in (2.10) and $\boldsymbol{X}$ identifies U_1 and U_3 with zero-error. Then, $\hat{U}_2 = U_1$, since individuals 1 and 2 are in the same

cluster in F_2.

Finally, we have two metrics to measure the performance of a group testing algorithm. The first is the required number of tests T, which is the number of rows of $\boldsymbol{X}$ in the two-step sampled group testing algorithm family we defined. Having the minimum number of required tests is one of the aims of the group testing procedure. The second metric is the expected number of false classifications. Due to the second step of decoding, the overall two-step sampled group testing algorithm is not a zero-error algorithm (except for the choice of $m = f$), and the expected number of false classifications is a metric to measure the error performance of the algorithm. We use $E_f = \mathbb{E}[d_H(U \oplus \hat{U})]$ to denote the *expected number of false classifications*, where $d_H(\cdot)$ is the Hamming weight of a binary vector.

Designing a two-step sampled group testing algorithm consists of selecting F_m, then designing the function M, and then designing the non-adaptive test matrix $\boldsymbol{X}$ for the second step of the testing and the first step of the decoding phase for zero-error identification of the infection status of the sampled σ_m individuals. We consider cluster formation trees and uniform patient zero assumptions for our infection spread model, and we consider two-step sampled group testing algorithms for the group test design.

The following section presents a motivating example to demonstrate our key ideas.

2.3 Motivating Example

Consider the following example. There are $n = 10$ individuals and a cluster formation tree with $f = 3$ levels. Full characterization of F is as follows,

$$F = \begin{cases} F_1 = \{\{1,2,3\},\{4,5\},\{6,7,8,9,10\}\}, & w.p.\ 0.4 \\ F_2 = \{\{1,2\},\{3\},\{4,5\},\{6,7,8,9,10\}\}, & w.p.\ 0.2 \\ F_3 = \{\{1,2\},\{3\},\{4,5\},\{6,7\},\{8,9,10\}\}, & w.p.\ 0.4 \end{cases} \tag{2.13}$$

First, we find the optimal sampling functions, M, for all possible selections of F_m. First, note that M selects exactly one individual from each subset that forms F_m by definition. Therefore, the number of sampled individuals is constant for a fixed choice of F_m. Thus, in the optimal sampling function design, the only parameter we consider is the minimum number of expected false classifications E_f. Note that a false classification occurs only when one of the sampled individuals has a different infection status than one of the individuals in its cluster in F_m. For instance, assume that $m = 1$ is chosen. Then, assume that the sampling function M selects individual 1 from the set $S_1^1 = \{1,2,3\}$. Recall that after the second step of the two-step group testing algorithm, by using $\boldsymbol{X}$, the infection status of individual 1 is identified with zero error, and its status is used to estimate the status of individuals 2 and 3 since they are in the same cluster in $F_m = F_1$. However, with positive probability, individuals 1 and 3 can have distinct infection statuses, in which case, a false classification occurs. Note that this scenario occurs only when F_m is at

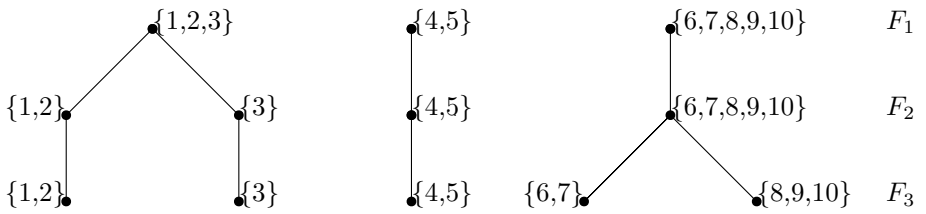


Figure 2.3: Cluster formation tree $\mathcal{F}$.

a higher level than the realized F in the cluster formation tree $\mathcal{F}$, where we refer to F_1 as the top level of the cluster formation tree and F_f as the bottom level.

While finding the optimal sampling function M, one must consider the possible false classifications and minimize E_f, the expected number of false classifications. As shown in Figure 2.3, the cluster $\{4, 5\}$ does not get partitioned, and for all three choices of F_m, M can sample either one of the individuals 4 and 5. This selection does not change the expected number of false classifications since $U_4 = U_5$ in all possible realizations of F. For all sampling cluster formation selections, we have the following analysis:

- *If* $F_m = F_1$: If M samples individual 1 or 2 from the cluster $S_1^1 = \{1, 2, 3\}$, a false classification occurs if $F = F_2$ and the cluster $\{1, 2\}$ is infected. In that case, individual 3 is falsely classified as infected. Similar false classification occurs when $F = F_3$ and the cluster $\{1, 2\}$ is infected. Similarly, in these cases, if individual 3 is infected, individual 3 is falsely classified as non-infected. Thus, for cluster $\{1, 2, 3\}$, when either individual 1 or 2 is sampled, the expected number of false classifications is:

$$(p_F(F_2) + p_F(F_3))(p_Z(1) + p_Z(2) + p_Z(3)) = 0.6 \times 0.3 = 0.18 \tag{2.14}$$

Similarly, when individual 3 is sampled from the cluster $\{1, 2, 3\}$, individuals 1 and 2 are falsely classified when $F = F_2$ or $F = F_3$ and either the cluster $\{1, 2\}$ or individual 3 is infected. Thus, in that case, the expected number of false classifications is:

$$2(p_F(F_2) + p_F(F_3))(p_Z(1) + p_Z(2) + p_Z(3)) = 2 \times 0.6 \times 0.3 = 0.36 \quad (2.15)$$

Thus, (2.14) and (2.15) imply that, for cluster $S_1^1 = \{1, 2, 3\}$, the optimal M should select either individual 1 or 2 for testing. As discussed above, for cluster $S_2^1 = \{4, 5\}$, the selection of sampled individuals is indifferent and results in 0 expected false classification. Finally, for cluster $S_3^1 = \{6, 7, 8, 9, 10\}$, a similar analysis implies that, the optimal M should select one of the individuals in $\{8, 9, 10\}$ for testing.

- *If* $F_m = F_2$: Similar combinatorial arguments follow and we conclude that the selection of sampled individuals from the clusters $S_1^2 = \{1, 2\}$, $S_2^2 = \{3\}$ and $S_3^2 = \{4, 5\}$ are indifferent in terms of the expected number of false classifications. Only possible false classification can happen in cluster $S_4^2 = \{6, 7, 8, 9, 10\}$ when $F = F_3$ and the infected cluster is either $S_4^3 = \{6, 7\}$ or $S_5^3 = \{8, 9, 10\}$. Similar to the case $m = 1$, if the sampled individual is either 6 or 7, then the expected number of false classifications is 0.6 in contrast to 0.4 when the sampled individual is one of 8, 9, and 10. Thus, the optimal M should select one of the individuals 8, 9, and 10 as the sampled individual to minimize the expected number of false classifications.

- *If* $F_m = F_3$: It is not possible to make a false classification since for all clusters in F_3, all individuals that are in the same cluster have the same infection status with probability 1.

Therefore, for this example, the optimal sampling function selects either individual 1 or 2 from the set S_1^1; selects either 4 or 5 from the set S_2^1; and selects either 8, 9 or 10 from the set S_3^1 if $F_m = F_1$ and the same sampling is optimal with the addition of individual 3, if $F_m = F_2$. Let us assume that M selects the individual with the smallest index when the selection is indifferent among a set of individuals. Thus, the optimal sampling function M for this example is: $\{1, 4, 8\}$, $\{1, 3, 4, 8\}$ or $\{1, 3, 4, 6, 8\}$, depending on the selection of F_m being F_1, F_2, or F_3, respectively.

Now, for these possible sets of sampled individuals, we need to design zero-error non-adaptive test matrices.

- *If* $F_m = F_1$ (i.e., $M = \{1, 4, 8\}$): The set of all possible infected sets is $\mathcal{P}(K_M) = \{\{1\}, \{4\}, \{8\}\}$. By a counting argument, we need at least two tests since each of the three possible infected sets must result in a unique result vector y and each one of these sets has one element. We can achieve this lower bound by using the following test matrix:

	1	4	8
Test 1	0	1	1
Test 2	1	0	1

- *If* $F_m = F_2$ (i.e., $M = \{1, 3, 4, 8\}$): In this case, the set of all possible infected sets is now $\mathcal{P}(K_M) = \{\{1\}, \{3\}, \{1, 3\}, \{4\}, \{8\}\}$. In the classical zero-error

construction for the combinatorial group testing model, one can construct *d-separable* matrices, and the rationale behind the construction is to enable the decoding of the infected set when the infected set can be any d-sized subset of $[n]$. However, in our model, the set of all possible infected sets, i.e., $\mathcal{P}(K_M)$, is not a set of all fixed-sized subsets of $[n]$, but instead, consists of varying-sized subsets of $[n]$ that are structured, depending on the given $\mathcal{F}$. As illustrated in Figure 2.3, a given cluster formation tree $\mathcal{F}$ can be represented by a tree structure with nodes[3] representing possible infected sets, i.e., clusters at each level. Then, the aim of constructing a zero-error test matrix is to have unique test result vectors for each unique possible infected set, i.e., unique nodes in the cluster formation tree. In Figure 2.4, we present the subtree of $\mathcal{F}$, which ends at the level F_2, with assigned result vectors to each node. One must assign unique binary vectors to each node, except for the nodes that do not get partitioned while moving from level to level: those nodes represent the same cluster, and thus, the same vector is assigned, as seen in Figure 2.4. Moreover, while merging in upper-level nodes, binary OR of vectors assigned to the descendant nodes must be assigned to their ancestor node. By combinatorial arguments, one can find the minimum vector length such that such vectors can be assigned to the nodes.

In this case, the required number of tests must be at least three, and by assigning result vectors as in Figure 2.4, we can construct the following test

[3]Throughout the chapter, we use the word "node" only for the possible clusters in the cluster formation tree representations, not for the vertices in the connection graphs that represent the individuals.

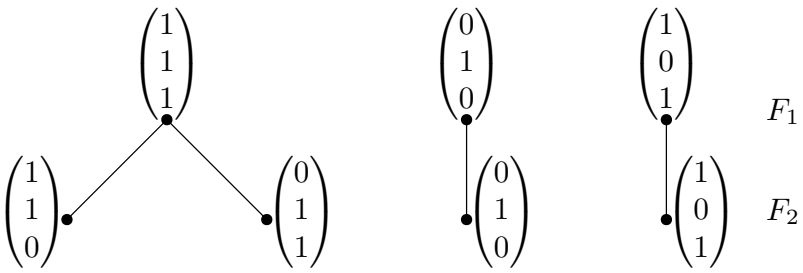


Figure 2.4: Subtree of $\mathcal{F}$ with assigned result vectors for each node.

matrix $\boldsymbol{X}$:

	1	3	4	8
Test 1	1	0	0	1
Test 2	1	1	1	0
Test 3	0	1	0	1

Note that for all elements of $\mathcal{P}(K_M)$, the corresponding result vector is unique and satisfies the tree structure criteria, as shown in Figure 2.4.

- *If* $F_m = F_3$ (i.e., $M = \{1, 3, 4, 6, 8\}$): In this case, the set of all possible infected sets is $\mathcal{P}(K_M) = \{\{1\}, \{3\}, \{1, 3\}, \{4\}, \{6\}, \{8\}, \{6, 8\}\}$. We give a tree structure representation with assigned result vectors of length three that achieves the tree structure criteria discussed above, shown in Figure 2.5 where each unique node is assigned a unique vector except for the nodes that do not get partitioned while moving from level to level. Note that every unique node in the tree representation corresponds to a unique element of $\mathcal{P}(K_M)$. The corresponding test matrix $\boldsymbol{X}$ is the following 3×5 matrix:

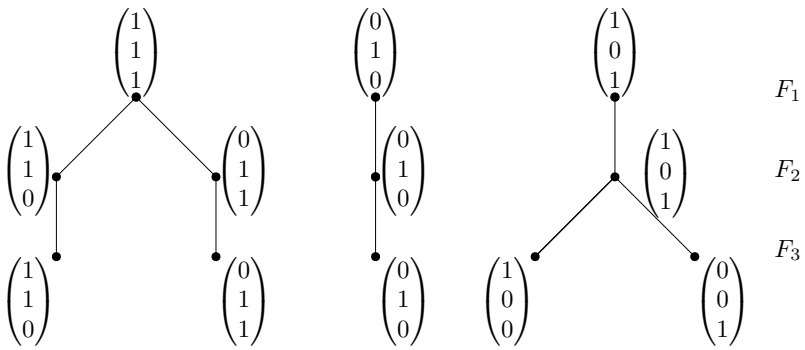


Figure 2.5: $\mathcal{F}$ with assigned result vectors for each node.

	1	3	4	6	8
Test 1	1	0	0	1	0
Test 2	1	1	1	0	0
Test 3	0	1	0	0	1

A more structured and detailed analysis of the selection of the optimal sampling function and the minimum number of required tests is given in the next section.

We finalize our analysis of this example by calculating the expected number of false classifications where $E_{f,\alpha}$ denotes the conditional expected false classifications, given $F = F_\alpha$:

- *If* $F_m = F_1$:

$$\begin{aligned} E_f &= \sum_\alpha p_F(F_\alpha) E_{f,\alpha} \\ &= p_F(F_2) E_{f,2} + p_F(F_3) E_{f,3} \\ &= 0.2(0.3 \times 1) + 0.4(0.3 \times 1 + 0.5 \times 2) = 0.58 \end{aligned} \tag{2.16}$$

- *If* $F_m = F_2$:

$$E_f = p_F(F_3)E_{f,3} = 0.4(0.5 \times 2) = 0.4 \tag{2.17}$$

- *If* $F_m = F_3$, we have $E_f = 0$.

Note that the choice of F_m is a design choice. One can use time sharing[4] between different choices of m, depending on the specifications of the desired group testing algorithm. For instance, if a minimum number of tests is desired, then one can pick $m = 1$, which results in 2 tests, which is the minimum possible, but with expected 0.58 false classifications, which is the maximum possible in this example. On the other hand, if the minimum expected false classification is desired, one can pick $m = 3$, resulting in 0 expected false classifications, which is the minimum possible, but with three tests, which is the maximum possible in this example. Generally, there is a trade-off between the number of tests and the number of false classifications, and we can formulate optimization problems for specific system requirements, such as finding a time-sharing distribution for F_m that minimizes the number of tests for a desired level of false classifications, or vice versa.

In the following section, we describe the details of our proposed group testing algorithm.

[4]Time sharing can be implemented by assigning a probability distribution to F_m over $\mathcal{F}$, instead of picking one cluster formation from $\mathcal{F}$ to be F_m deterministically.

2.4 Proposed Algorithm and Analysis

In our $\mathcal{F}$-*separable matrix* construction, we aim to construct binary matrices with n columns. For each possible infected subset of the selected individuals, there must be a corresponding distinct result vector. A binary matrix $\boldsymbol{X}$ is $\mathcal{F}$-separable if

$$\bigvee_{i \in S_1} \boldsymbol{X}^{(i)} \neq \bigvee_{i \in S_2} \boldsymbol{X}^{(i)} \tag{2.18}$$

is satisfied for all distinct subsets S_1 and S_2 in the set of all possible infected subsets, where $\boldsymbol{X}^{(i)}$ denotes the ith column of $\boldsymbol{X}$. In d-separable matrix construction [17], this condition must hold for all subsets S_1 and S_2 of cardinality d; here, it must hold for all possible feasible infected subsets as defined by $\mathcal{F}$. From this point of view, our $\mathcal{F}$-separable test matrix construction exploits the known structure of $\mathcal{F}$ and thus, it results in an efficient zero-error non-adaptive test design for the second step of our proposed algorithm.

We adopt a combinatorial approach to the design of the non-adaptive test matrix $\boldsymbol{X}$. Note that, for a given M, we have σ_m individuals to be identified with zero-error probability. The key point of our algorithm is that the infected set of individuals among those selected individuals can only be some specific subsets of those σ_m individuals. Without any information about the cluster formation, any one of the 2^{σ_m} subsets of the selected individuals can be the infected set. However, since we are given $\mathcal{F}$, we know that the infected set among the selected individuals, K_M, can be one of the 2^{σ_m} subsets only if there exists at least one set S_i^j that contains

K_M and there is no element in the difference set $M \backslash K_M$ such that it is an element of all sets S_i^j containing K_M. This fact, especially in a cluster formation tree structure, significantly reduces the total number of possible infected subsets that need to be considered. Therefore, we can focus on such subsets and design the test matrix $\boldsymbol{X}$ by requiring that the logical OR operation of the columns corresponding to the possible K_M sets be distinct to decode the test results with zero error. Let $\mathcal{P}(K_M)$ denote the set of possible infected subsets of the selected individuals, i.e., the set of possible sets that K_M can be. Then, matrix $\boldsymbol{X}$ must satisfy (2.18) for all distinct S_1 and S_2 that are elements of $\mathcal{P}(K_M)$. Note that the decoding process is a mapping from the result vectors to the infected sets; thus, we require the distinct result vector property to guarantee zero-error decoding.

Designing the $\boldsymbol{X}$ matrix that satisfies the aforementioned property is the key idea of our algorithm. Before going into the design of $\boldsymbol{X}$, we first derive the expected number of false classifications in a given two-step sampled group testing algorithm. Recall that false classifications occur during the second step of the decoding phase. In particular, in the second step of the decoding phase, depending on the selection of the sampling cluster formation F_m, the infection statuses of the selected individuals M are assigned to the other individuals such that the infection status estimate is the same within each cluster. For fixed sampling cluster formation F_m and the sampling function M, the number of expected false classifications can be calculated as in the following theorem.

Theorem 2.1 *In a two-step sampled group testing algorithm with the given sam-*

pling cluster formation F_m and the sampling function M over a cluster formation tree structure defined by $\mathcal{F}$ and p_F, with uniform patient zero distribution p_Z over $[n]$, the expected number of false classifications given $F = F_\alpha$ is

$$E_{f,\alpha} = \sum_{i \in [\sigma_m]} \left(\frac{|S^\alpha(M_i)|}{n} \cdot |S_i^m \backslash S^\alpha(M_i)| \qquad + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} \frac{|S_j^\alpha|^2}{n} \right) \tag{2.19}$$

and the expected number of false classifications is

$$E_f = \sum_{\alpha > m} p_F(F_\alpha) E_{f,\alpha} \tag{2.20}$$

where $S^\alpha(M_i)$ is the subset in the partition F_α which contains the ith selected individual.

Next, we obtain Theorem 2.2 to characterize the optimal choice of the sampling function M. First, we define $\beta_i(k)$ functions. For $i \in [f]$ and $k \in [n]$,

$$\beta_i(k) \triangleq \sum_{j > i} p_F(F_j) \left(|S^j(k)| \cdot |S^i(k) \backslash S^j(k)| \qquad + \sum_{S_l^j \subseteq S^i(k) \backslash S^j(k)} |S_l^j|^2 \right) \tag{2.21}$$

where $S^i(k)$ is the subset in partition F_i that contains k.

Theorem 2.2 *For sampling cluster formation F_m, the optimal choice of M that minimizes the expected number of false classifications is*

$$M_i = \arg\min_{k \in S_i^m} \beta_m(k) \tag{2.22}$$

where M_i is the ith selected individual. Moreover, the number of required tests is constant and is independent of the choice of M.

We present the proofs of Theorem 2.1 and Theorem 2.2 in the Appendix in Section 2.8.

The optimal M analysis focuses on choosing the sampling function that results in the minimum expected number of false classifications among the set of functions that select exactly one individual from each cluster of a given F_m. For some scenarios, it is possible to choose a sampling function that selects multiple individuals from some clusters of a given F_m that achieves expected false classifications-required number of tests points that cannot be achieved by the optimal M in (2.49). However, in most cases, the sampling functions of interest, i.e., the sampling functions that choose exactly one individual from each F_m, are globally optimal. First, the sampling functions that select multiple individuals from a cluster that never gets partitioned further in the levels below F_m are sub-optimal. These sampling functions select multiple individuals to identify who are guaranteed to have the same infection status. For instance, in zero expected false classifications case, i.e., the bottom level F_f is chosen as the sampling cluster formation, sampling more than one individual from each cluster is sub-optimal. Second, picking the sampling cluster formation F_m and choosing an M such that multiple individuals are chosen from some clusters that further get partitioned in the levels below F_m is equivalent to choosing a sampling cluster formation below F_m and using an M that selects exactly one individual from each cluster of the new sampling cluster formation, except for the scenarios where

there exist partitioning of multiple clusters in two consecutive cluster formations in a given $\mathcal{F}$, and one can consider a sampling function that selects multiple individuals from some clusters of a given F_m that cannot be represented as a sampling function that selects exactly one individual from each cluster of another cluster formation $F_{m'}$. For compactness, we focus on the family of sampling functions M that selects exactly one individual from each cluster of the chosen F_m.

So far, we have presented a method to select individuals to be tested to minimize the expected number of false classifications. Now, we move on to the design of $\boldsymbol{X}$, the zero-error non-adaptive test matrix, which identifies the infection status of the selected individuals M with a minimum number of tests. Recall that since $|\mathcal{F}| = f$, there are f possible choices of F_m, and each choice results in a different test matrix $\boldsymbol{X}$.

Based on the combinatorial viewpoint stated in (2.18), we propose a family of non-adaptive group testing algorithms which satisfy the separability condition for all of the subsets in $\mathcal{P}(K_M)$, which is determined by $\mathcal{F}$. We call such matrices *$\mathcal{F}$-separable matrices* and non-adaptive group tests that use *$\mathcal{F}$-separable matrices* as their test matrix as *$\mathcal{F}$-separable non-adaptive group tests*. In the rest of the section, we present our results on the required number of tests for *$\mathcal{F}$-separable non-adaptive group tests*.

The key idea of designing an $\mathcal{F}$-separable matrix is determining the set $\mathcal{P}(K_M)$ for a given set of selected individuals M and the tree structure of $\mathcal{F}$ so that we can find binary column vectors for each selected individual where all of the corresponding possible result vectors are distinct. Note that, for a given choice of F_m, if we consider

the corresponding subtree of $\mathcal{F}$ which starts from the first level F_1 and ends at the level F_m, the problem of finding an $\mathcal{F}$-separable non-adaptive test matrix is equivalent to finding a set of length T binary column vectors for each node at level F_m that satisfy the following criteria:

- For every node at the levels that are above the level F_m, each node must be assigned a binary column vector that is equal to the OR of all vectors that are assigned to its descendant nodes. This is because each node in the tree corresponds to a possible set of infected individuals among the selected individuals, where each merging of the nodes corresponds to the union of the possible infected sets, which results in taking the OR of the assigned vectors of the merged nodes.

- Each assigned binary vector must be unique for each unique node, i.e., for every node that represents a unique set S_i^j. The assigned vector remains the same for the nodes that do not split between two levels. This is because each unique node (note that when a node does not split between levels, it still represents the same set of individuals) corresponds to a unique possible infected subset of the selected individuals, and they must satisfy (2.18).

In other words, for a cluster formation tree with assigned result vectors to each node, a sufficient condition for the achievability of $\mathcal{F}$-separable matrices is as follows:

Let u be a node with Hamming weight $d_H(u)$. Then, the number of all descendant nodes of u with constant Hamming weights i must be less than $\binom{d_H(u)}{i}$ for all i. This must hold for all nodes u. Furthermore, the number of nodes with

constant Hamming weight i must be less than $\binom{T}{i}$ for all i. In addition, Hamming weights of the nodes must strictly decrease while moving from ancestor nodes to descendant nodes.

This condition is indeed sufficient because it guarantees the existence of a unique set of vectors that can be assigned to each node of the subtree of $\mathcal{F}$ that satisfies the merging/OR structure determined by the subtree.

The problem of designing an $\mathcal{F}$-separable non-adaptive group test can be reduced to finding the minimum number T, for which we can find σ_m binary vectors with length T, such that all vectors that are assigned to the nodes satisfy the above condition. Here the assigned vectors are the result vectors y when the corresponding node is the infected node.

We have the following definitions that we need in Theorem 2.3. For a given $\mathcal{F}$, we define $\lambda_{S_i^j}$ as the number of unique ancestor nodes of the set S_i^j. We also define λ_j as the number of unique sets S_a^b in $\mathcal{F}$ at and above the level F_j. Note that $\sum_{a\leq j}\sigma_a$ is the total number of sets S_a^b in $\mathcal{F}$ at and above the level F_j, and thus we have,

$$\sum_{a\leq j}\sigma_a \geq \lambda_j \tag{2.23}$$

Theorem 2.3 *For given $\mathcal{F}$ and F_m for $m < f$, the number of required tests for an $\mathcal{F}$-separable non-adaptive group test, i.e., the number of rows of the test matrix $\boldsymbol{X}$,*

must satisfy

$$T \geq \max \left\{ \max_{j \in [\sigma_m]} \left(\lambda_{S_j^m} + 1 \right), \ \lceil \log_2(\lambda_m + 1) \rceil \right\} \tag{2.24}$$

with addition of 1's removed in (2.24) for the special case of $m = f$.

We present the proof of Theorem 2.3 in the Appendix in Section 2.8. Note that Theorem 2.3 is a converse argument without a statement about the achievability of the given lower bound. In fact, the given lower bound is not always achievable.

Complexity: The time complexity of the two-step sampled group testing algorithms consists of the complexity of finding the optimal M given F_m and $\mathcal{F}$, the complexity of the construction of the $\mathcal{F}$-separable test matrix given M and $\mathcal{F}$, and the complexity of the decoding of the test results given the test matrix $\boldsymbol{X}$ and the result vector y. In the following lemmas, we analyze the complexity of these processes.

Lemma 2.1 *For a given cluster formation tree* $\mathcal{F}$ *and a sampling cluster formation* F_m*, the complexity of finding the optimal* M *as in Theorem 2.2 is*

$$O(n(f-m)\zeta_m) \tag{2.25}$$

where $\zeta_m = \max_{k \in [n]} |\{S_l^f : S_l^f \subseteq S^m(k) \backslash S^f(k)\}|$.

Proof: To find the optimal M, $\beta_m(k)$ needs to be calculated as in (2.21) for each $k \in [n]$. The complexity of each of these calculations is bounded above by the

number of cluster formations below F_m multiplied by the number of clusters at level f that do not include the individual k and form the cluster $S^m(k)$, i.e., the clusters S_l^f that satisfy $S_l^f \subseteq S^m(k) \backslash S^f(k)$. Note that this upper bound varies for each $k \in [n]$ and the total complexity is the summation of these sizes multiplied by $f - m$, i.e., the number of cluster formations below F_m, for each $k \in [n]$. As an upper bound, we consider the maximum of these sizes, i.e., ζ_m, concluding the proof. ■

In the next lemma, we analyze the complexity of the construction of the $\mathcal{F}$-separable test matrix given M and $\mathcal{F}$.

Lemma 2.2 *For a given cluster formation tree $\mathcal{F}$ and a sampling function M, the complexity of assigning the binary result vectors to the nodes in $\mathcal{F}$, and thus, the construction of the $\mathcal{F}$-separable test matrix is $\Omega(m\sigma_m)$.*

Proof: When the cluster formation tree $\mathcal{F}$ and the sampling function M are given, to assign unique binary result vectors to each node in $\mathcal{F}$ that represents a unique possible infected cluster, we need to consider the subtree of $\mathcal{F}$ that starts with the level F_1 and ends at the level F_m, as in the example in Figure 2.4. Then, we need to traverse from each bottom node in the subtree to the top node to detect every cluster merging. This results in finding the numbers $\lambda_{S_j^m}$ for $j \in [\sigma_m]$ and λ_m and unique binary test result vectors can be assigned to each unique node in $\mathcal{F}$. The traversing on the subtree of $\mathcal{F}$ starting from the bottom level F_m to the top level for each bottom level node has the complexity $\Theta(m\sigma_m)$. This traversing does not immediately result in the explicit construction of unique binary result vectors

to be assigned, but it gives an asymptotic lower bound for the complexity of the construction of the $\mathcal{F}$-separable test matrices. ■

Note that the Lemma 2.2 is an asymptotic lower bound for the complexity of the binary result vector assignment to the unique nodes in $\mathcal{F}$, and thus, for the construction of the $\mathcal{F}$-separable test result matrix $\boldsymbol{X}$. This analysis is a baseline for the proposed model, and proposing explicit $\mathcal{F}$-separable test matrix constructions with the exact number of required tests and complexity is an open problem.

Lemma 2.3 *For a given $\mathcal{F}$-separable test matrix $\boldsymbol{X}$, with corresponding cluster formation tree $\mathcal{F}$ with assigned binary result vectors to each node and the result vector y, the decoding complexity is $O(1)$.*

Proof: While constructing the $\mathcal{F}$-separable test matrix, we consider the assignment of the unique binary result vectors to the nodes in the given cluster formation tree $\mathcal{F}$. For a given test matrix $\boldsymbol{X}$ and the result vector y, the decoding problem is a hash table lookup with the complexity $O(1)$. ■

Since during the proposed process of assignment of unique binary result vectors to each unique node in $\mathcal{F}$, we specifically assign the test result vectors to every unique possible infected set, the decoding process is basically a hash table lookup, resulting in fast decoding with low complexity.

Key Steps of the Proposed Algorithm: The summary of the key steps of the two-step sampled group testing algorithm is given below:

- We start with the assumption that exact connections between the individuals are not known, but the probability distribution of the possible edge realizations

is known.

- The given edge set probability distribution results in a random cluster formation variable, F. Each possible cluster formation is a partition of the set of all individuals.

- Out of all possible cluster formations (which we call this set as $\mathcal{F}$), one cluster formation is selected as the sampling cluster formation, which we call F_m.

- Exactly one individual is selected from each cluster in F_m. These individuals are then tested and identified.

- The selection is made according to the sampling function M. For the given choice of F_m, M selects the individuals from the clusters that minimize the expected number of false classifications, given in Theorem 2.2, and this results in the expected number of false classifications given in Theorem 2.1.

- By using the given set of possible cluster formations, $\mathcal{F}$, an $\mathcal{F}$-separable test matrix is constructed to identify the individuals selected by M. This test matrix is guaranteed to identify the selected individuals since the construction is based on assigning a unique test result vector to every possible infected set among the selected individuals.

- In Theorem 2.3, we present a converse argument by giving a lower bound for the required number of tests in terms of the system parameters.

- After obtaining the test results and identifying the selected individuals with zero error, for each selected individual, their infection status is assigned to

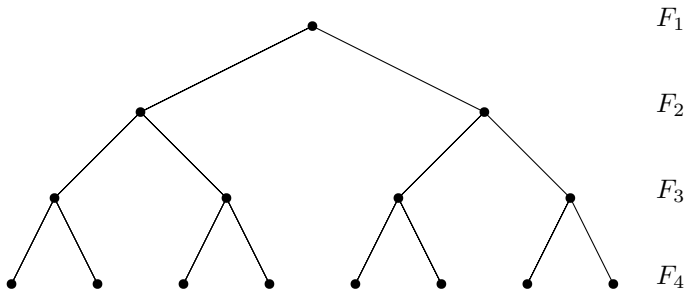


Figure 2.6: A 4-level exponentially split cluster formation tree.

the others in their cluster, in F_m. Note that there is exactly one individual selected and identified from every cluster in F_m. This step introduces possible false classifications.

- Selecting F_m from lower levels from the possible cluster formations tree results in lower expected false classifications while increasing the number of required tests for identification. This results in a trade-off between the number of tests and expected false classifications. By using a randomized selection of F_m, intermediate points can also be achieved for the expected false classifications and required number of tests.

In the next section, we introduce and focus on a family of cluster formation trees, which we call *exponentially split cluster formation trees*. For this analytically tractable family of cluster formation trees, we achieve the lower bound in Theorem 2.3 *order-wise*, and we compare our result with the results in the literature.

2.5 Exponentially Split Cluster Formation Trees

In this section, we consider a family of cluster formation trees and explicitly characterize the selection of optimal sampling function, the resulting expected number of

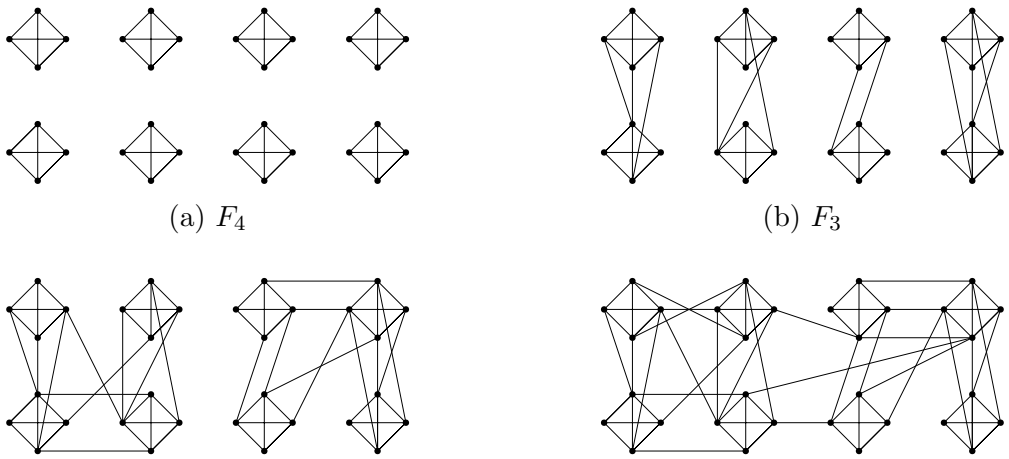
false classifications, and the number of required tests. We also compare our results with Hwang's generalized binary splitting algorithm [7] and zero-error non-adaptive group testing algorithms to show the gain of utilizing the cluster formation structure.

A cluster formation tree $\mathcal{F}$ is an *exponentially split cluster formation tree* if it satisfies the following criteria:

- An exponentially split cluster formation tree that consists of f levels has 2^{i-1} nodes at level F_i, for each $i \in [f]$, i.e., $\sigma_i = 2^{i-1}, i \in [f]$.
- At level F_i, every node has $2^{f-i}\delta$ individuals where δ is a constant positive integer, i.e., $|S_j^i| = 2^{f-i}\delta, i \in [f], j \in [\sigma_i]$.
- Every node has exactly two descendant nodes in one level below in the cluster formation tree, i.e., every node is partitioned into equal-sized two nodes when moving one level down in the cluster formation tree.
- Random cluster formation variable F is uniformly distributed over $\mathcal{F}$, i.e., $p_F(F_i) = 1/f, i \in [f]$.

We analyze the expected number of false classifications and the required number of tests for exponentially split cluster formation trees by using the general results derived in Section 2.4. In Figure 2.6, we give a 4-level exponentially split cluster formation tree example. In that example, there is $2^0 = 1$ node at level F_1, and the number of nodes gets doubled at each level since each node is split into two nodes when moving one level down in the tree. Also, the sizes of the nodes at the same level are the same, with the bottom-level nodes having the size δ.

Being a subset of cluster formation trees, exponentially split cluster formation trees correspond to random connection graphs where edges between individuals are not independently realized in non-trivial cases. For instance, in Figure 2.7, we present 4 different possible realizations of edges of a 4-level exponentially split cluster formation tree system, given in Figure 2.6, where there are $\delta = 4$ individuals in the bottom level clusters. Here, if the edges between individuals are realized independently, there would be possible cluster formations that do not result in exponentially split cluster formation tree structure. The edge realizations are correlated in the sense that if there is at least one edge realized between two bottom-level neighbor clusters, then there must be at least one edge realized between other bottom-level neighbor cluster pairs as well. Similarly, if there is at least one bottom level cluster pair that are not immediate neighbors but get merged in some upper level F_k in $\mathcal{F}$, then other bottom level cluster pairs that get merged in F_k must be connected as well. In Figure 2.7, in F_4 realization, the only edges present are the edges that form bottom-level clusters. In F_3 realization, there is at least one edge realized between each bottom-level neighbor cluster pair, resulting in clusters of 8 individuals. Similarly, there are more distant connections that are realized in F_2 and F_1. From a practical point of view, the 4-level exponential split cluster formation tree example in Figure 2.6 and Figure 2.7 can be used to model real-life scenarios, such as the infection spread in an apartment complex with multiple buildings. In the bottom level, there are households that are guaranteed to be connected, and in the F_3 level the households that are in close contact are connected; in F_2 level, there is a connection building-wise, and in F_1, the whole community is connected. Note



(a) F_4 (b) F_3

(c) F_2 (d) F_1

Figure 2.7: 4 realizations of a random connection graph $\mathcal{C}$ that falls under four different cluster formations in a 4-level exponentially split cluster formation tree with $\delta = 4$.

that the connections given in Figure 2.7 are realization examples that fall under four possible cluster formations and all edge realization scenarios are possible as long as the resulting cluster formation is one of the four given cluster formations. While designing the group testing algorithm, the given information is the probability distribution over the cluster formations, and in practice, one can expect a probability distribution where bottom-level cluster formations, i.e., cluster formations towards F_4, have higher probabilities in a community where there are strict social isolation measures, and high immunity rates for a contagious infection whereas higher probabilities of upper-level cluster formations, i.e., cluster formations toward F_1, can be expected for communities with high contact rate and lower immunity.

Optimal sampling function and expected number of false classifications: Due to the symmetry of the system, for any choice F_m, each element of S_i^m has the same $\beta_m(i)$ value for all $i \in \sigma_m$. Therefore, the sampling function selects

individuals from each set arbitrarily, i.e., the selection of a particular individual does not change the expected number of false classifications. Thus, we can pick any sampling function that selects one element from each S_i^m. By Theorem 2.1, the expected number of false classifications, for given F_m, is

$$E_f = \sum_{\alpha>m} \frac{1}{f} \sum_{i\in[\sigma_m]} \left(\frac{|S^\alpha(M_i)|}{n} \cdot |S_i^m \backslash S^\alpha(M_i)| \quad + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} \frac{|S_j^\alpha|^2}{n} \right) \tag{2.26}$$

$$= \sum_{\alpha>m} \frac{1}{f} \frac{\sigma_m}{\sigma_\alpha} \left(\delta(2^{f-m} - 2^{f-\alpha}) + (2^{\alpha-m} - 1)\delta 2^{f-\alpha} \right) \tag{2.27}$$

$$= \sum_{\alpha>m} \frac{2^{f+1}\delta}{f} \left(2^{-\alpha} - 2^{m-2\alpha} \right) \tag{2.28}$$

$$= \frac{2^{f+1}\delta}{f} \left(\sum_{\alpha>m} 2^{-\alpha} - 2^m \sum_{\alpha>m} 2^{-2\alpha} \right) \tag{2.29}$$

$$= \frac{2^{f+1}\delta}{f} \left((2^{-m} - 2^{-f}) - \frac{2^m}{3}(2^{-2m} - 2^{-2f}) \right) \tag{2.30}$$

$$= \frac{\delta}{3f} \left(2^{f-m+2} + 2^{m-f+1} - 6 \right) \tag{2.31}$$

This expected number of false classifications takes its maximum value when $F_m = F_1$,

$$E_f = \frac{\delta}{3f} \left(2^{f+1} + 2^{2-f} - 6 \right) \tag{2.32}$$

and it takes its minimum value when $F_m = F_f$ as $E_f = 0$. Since the choice of F_m is a design parameter, one can use time sharing between the possible selections of F_m to achieve any desired value for the expected number of false classifications between $E_f = 0$ and E_f in (2.32).

Required number of tests: We first recall that, if we choose the sampling cluster formation level F_m, the required number of tests for selected individuals at that level for whom we design an $\mathcal{F}$-separable test matrix depends on the subtree that is composed of the first m levels of the cluster formation tree $\mathcal{F}$. Note that the first m levels of an exponentially split cluster formation tree are also an exponentially split cluster formation tree with m levels. In Theorem 2.4 below, we focus on the sampling cluster formation choice at the bottom level, $F_m = F_f$, and characterize the *exact* required number of tests to be between f and $\frac{4}{3}f$. This implies that the required number of tests at level F_f is $O(f)$; thus, the required number of tests at level F_m is $O(m)$.

Theorem 2.4 *For an f level exponentially split cluster formation tree, at level f, there exists an $\mathcal{F}$-separable test matrix, $\boldsymbol{X}$, with not more than $\frac{4}{3}f$ rows, i.e., an upper (achievable) bound for the number of required tests is $\frac{4}{3}(\log_2 n + 1)$ for n individuals. Conversely, this is also the capacity order-wise since the number of required tests must be greater than f.*

We present the proof of Theorem 2.4 in the Appendix in Section 2.8.

Expected number of infections: In an exponentially split cluster formation tree structure with f levels, the expected total number of infections is,

$$\sum_{i=1}^{f} \frac{1}{f} 2^{f-i}\delta = \frac{\delta}{f}(2^f - 1) \tag{2.33}$$

since $p_F(F_i) = 1/f$ and if $F = F_i$ then there are $2^{f-i}\delta$ infections. Thus, the expected

number of infections is $O\left(\frac{n}{\log_2 n}\right)$.

Comparison: To compare our results for the exponentially split cluster formation trees with other results in the literature, for fairness, we focus on the zero-error case in our system model, which happens when $F_m = F_f$ is chosen. Resulting sampling function selects in total 2^{f-1} individuals and the resulting number of required tests is between f and $\frac{4}{3}f$, i.e., $O(\log_2 n)$, as proved in Theorem 2.4. Note that, by performing at most $\frac{4}{3}f$ tests to 2^{f-1} individuals, we identify the infection status of $2^{f-1}\delta$ individuals with zero false classifications, which implies that the number of tests scales with the number of nodes at the bottom level, instead of the number of individuals in the system. This results in a gain scaled with δ. However, to fairly compare our results with the results in the literature, we ignore this gain and compare the performance of the second step of our algorithm only, i.e., the identification of the infection statuses of the selected individuals only. To avoid confusion, let $\delta = 1$, i.e., each cluster at the bottom level is an individual; thus, $n = 2^{f-1}$.

From (2.33), the expected number of infections in this system is $\frac{2^{f-1}}{f} = O(\frac{n}{\log_2 n})$. When the infections scale faster than $\sqrt{n}$, as proved in [18] (see also [2]), non-adaptive tests with zero-error criterion cannot perform better than individual testing. Since our algorithm results in $O(f) = O(\log_2 n)$ tests, it outperforms all non-adaptive algorithms in the literature. Furthermore, we compare our results with Hwang's generalized binary splitting algorithm [7], even though it is an adaptive algorithm and assumes prior knowledge of the exact number of infections. Hwang's

algorithm results in a zero-error identification of k infections among the population of n individuals with $k\log_2(n/k) + O(k)$ tests and attains the capacity of adaptive group testing [2, 7, 27]. Since the number of infections takes f values in the set $\{1, 2, 2^2, \ldots, 2^{f-1}\}$ uniformly randomly, the resulting mean value of the required number of tests when Hwang's generalized binary splitting algorithm is used is

$$\mathbb{E}[T_{\text{Hwang}}] = \sum_{i=0}^{f-1} \frac{1}{f}\left(2^i \log_2 2^{f-1-i}\right) + O\left(\frac{n}{\log_2 n}\right) \tag{2.34}$$

$$= \frac{f-1}{f}\sum_{i=0}^{f-1} 2^i - \frac{1}{f}\sum_{i=0}^{f-1} i2^i + O\left(\frac{n}{\log_2 n}\right) \tag{2.35}$$

$$= \frac{2^f - f - 1}{f} + O\left(\frac{n}{\log_2 n}\right) \tag{2.36}$$

$$= O\left(\frac{n}{\log_2 n}\right) \tag{2.37}$$

Thus, the expected number of tests when Hwang's generalized binary splitting algorithm is used scales as $O\left(\frac{n}{\log_2 n}\right)$, which is much faster than our result of $O(\log_2 n)$. We note that Hwang's generalized binary splitting algorithm assumes the prior knowledge of the exact number of infections and is an adaptive algorithm. Further, we have ignored the gain of our algorithm in the first step (i.e., $\delta = 1$). Despite these advantages, our algorithm outperforms Hwang's generalized binary splitting algorithm for exponentially split cluster formation trees.

2.6 Numerical Results

In this section, we present numerical results for the proposed two-step sampled group testing algorithm and compare our results with the existing results in the literature.

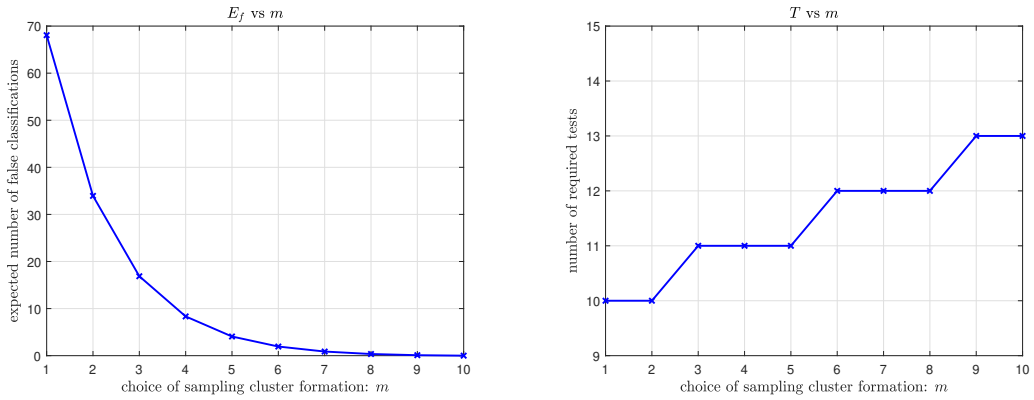
In the first simulation environment, we focus on exponentially split cluster formation trees as presented in Section 2.5. In the second simulation environment, we consider an arbitrary random connection graph, as discussed in Section 2.2, which does not satisfy the cluster formation tree assumption. We verify our analytical results in the first simulation environment by focusing on exponentially split cluster formation trees. We show that our ideas can be applied to arbitrary random connection graph-based networks in the second simulation environment.

2.6.1 Exponentially Split Cluster Formation Tree Based System

In the first simulation environment, we have an exponentially split cluster formation tree with $f = 10$ levels and $\delta = 1$ at the bottom level. For this system of $n = 2^{f-1}\delta = 512$ individuals, for each sampling cluster formation choice F_m (which is a design parameter), from $m = 1$, i.e., the top level of the cluster formation tree, to $m = 10$, i.e., the bottom level of the cluster formation tree, we calculate the expected number of false classifications and the minimum required number of tests. Note that the required number of tests is fixed for a fixed sampling cluster formation F_m, while the number of false classifications depends on the realization of the true cluster formation F_α and patient zero Z. This is because of the fact that when a sampling cluster formation is selected, the test matrix of choice is guaranteed to identify the sampled individuals with zero error, independent of the realized infections. In Figure 2.8(a), we plot the expected number of false classifications which meets the analytical expressions we found in Section 2.5. To plot Figure 2.8,

we run our simulation and realize the infections 1000 times to numerically obtain the average number of false classifications in the system. While calculating the minimum number of required tests, for each choice of F_m, our program finds the minimum T that satisfies the sufficient criteria that we presented in Section 2.4 and in the proof of Theorem 2.4 by searching over possible assignments of binary result vectors to the nodes in the given exponentially split cluster formation tree, starting from the vector length one and increasing the vector length by one if no such assignment is found. When a binary vector assignment to the nodes is found, the resulting test matrix is constructed and used for running the simulation 1000 times to obtain the numerical average of the expected number of false classifications. We plot the minimum required number of tests in Figure 2.8(b). Note that, unlike the number of false classifications, for a fixed F_m, the number of required tests is fixed, and thus, we do not repeat the simulations while calculating the required number of tests. The resulting non-adaptive test matrix $\boldsymbol{X}$ is fixed for a fixed F_m and identifies the infection status of the individuals selected by M, with zero error.

Next for this network setting, we compare our zero-error construction results with the results of a variation of Hwang's generalized binary splitting algorithm [7, 27], presented in [28], which further reduces the number of required tests by reducing the $O(k)$ term in the capacity expression of Hwang's algorithm. As we state in the comparison part of Section 2.5, the required number of tests in our algorithm scales with $O(\log_2 n)$. In our numerical results, we see the required number of tests is 13 at level $m = f = 10$, as seen in Figure 2.8(b). On the other hand,



(a) (b)

Figure 2.8: (a) Expected number of false classifications vs the choice of sampling cluster formation F_m. (b) Required number of tests vs the choice of sampling cluster formation F_m.

the average number of required tests for Hwang's algorithm scales as $O\left(\frac{n}{\log_2 n}\right)$, and is approximately 172 in this case. Further, when we remove the assumption of the known number of infections, we have to use the binary splitting algorithm presented originally in [5], which results in a number of tests that is not lower than individual testing, i.e., $n = 512$ tests in this case. For Hwang's generalized and the original binary splitting algorithm results, we run these algorithms 1000 times by realizing the infection status of the population at each iteration to obtain the numerical average of the number of required tests for both of these algorithms.

2.6.2 Arbitrary Random Connection Graph Based System

In our second simulation environment, we present an arbitrary random connection graph $\mathscr{C}$ with 20 individuals, shown in Figure 2.9(c), where the edges realize independently with probabilities shown on them (zero probability edges are not shown). In this system, since each independent realization of 9 edges that can be either

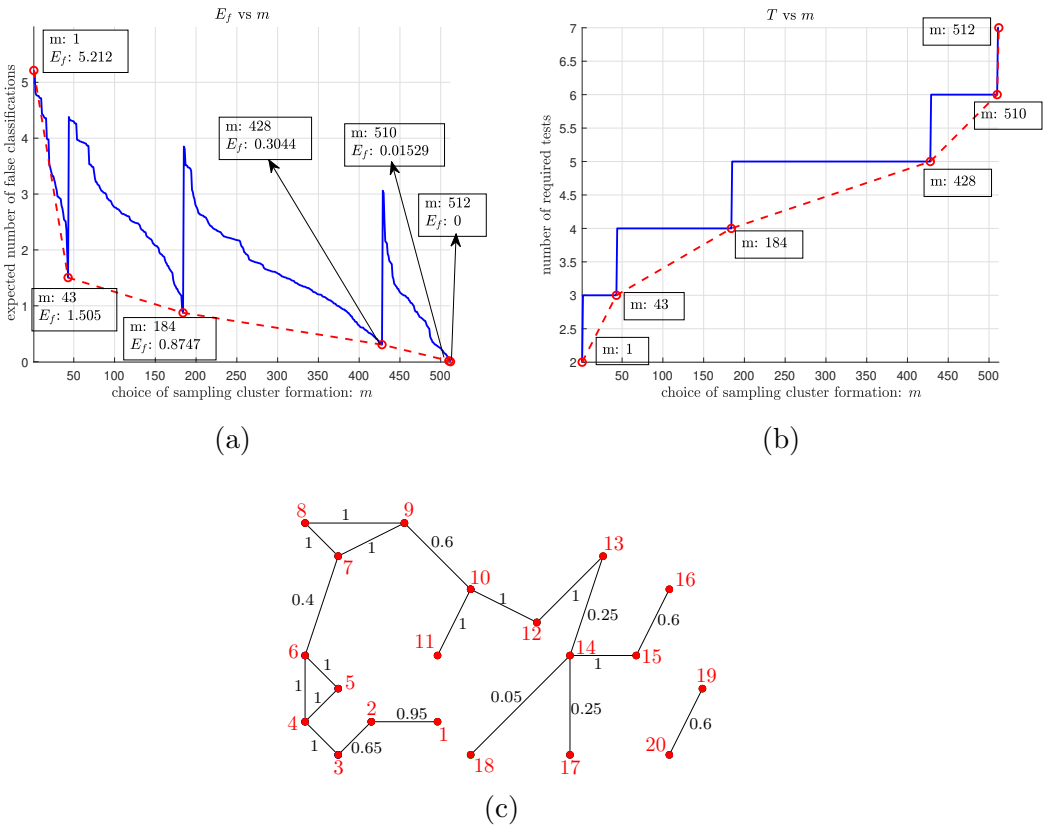


Figure 2.9: (a) Expected number of false classifications vs the choice of sampling cluster formation F_m. (b) Required number of tests vs the choice of sampling cluster formation F_m. (c) Random connection graph.

present or not results in a distinct cluster formation, in total, there are $2^9 = 512$ cluster formations that can be realized with positive probability. Note that this system with the random connection graph $\mathscr{C}$ does not yield a cluster formation tree, yet we still apply our ideas designed for cluster formation trees here. For each one of the 512 possible selections of m, we plot the corresponding expected number of false classifications in Figure 2.9(a) and the required number of tests in Figure 2.9(b) for our two-step sampled group testing algorithm.

In this simulation, for each possible choice of the sampling cluster formation F_m, we calculate the set of all possible infected sets $\mathcal{P}(K_M)$ for all possible choices

of M and calculate the resulting expected number of false classifications by also calculating p_F, the probability distribution of random cluster formations and select the optimal sampling function M. For the required number of tests, we find the minimum number of tests that satisfies the sufficient criteria that we presented in Section 2.4 in order to construct $\mathcal{F}$-separable matrices for this system. In our simulation environment, this procedure is done by brute force, since this system is not a cluster formation tree as in our system model and we cannot use the systematic results that we derived. This simulation demonstrates that the ideas presented can be generalized and applied to arbitrary random connection graph structures.

Since the system here is arbitrary unlike the exponentially split cluster formation tree structure in the first simulation environment in Section 2.6.1, the resulting expected number of false classifications is not monotonically decreasing when we sort the resulting required number of tests in the increasing order for the choices of F_m. In Figure 2.9(a), we mark the choices of sampling cluster formations that result in the minimum number of expected false classifications within each required number of test ranges. By using time-sharing between these choices of the sampling cluster formations, dotted red lines between them can be achieved. The 6 corner points in Figure 2.9(a)-(b) correspond to the following cluster formations,

$$F_1 = \{\{\text{1-18}\}, \{\text{19-20}\}\} \tag{2.38}$$

$$F_{43} = \{\{\text{1-6}\}, \{\text{7-13}\}, \{\text{14-18}\}, \{\text{19-20}\}\} \tag{2.39}$$

$$F_{184} = \{\{\text{1-6}\}, \{\text{7-9}\}, \{\text{10-13}\}, \{\text{14-18}\}, \{\text{19}\}, \{\text{20}\}\} \tag{2.40}$$

$$F_{428} = \{\{1\}, \{2\}, \{3\text{-}6\}, \{7\text{-}9\}, \{10\text{-}13\}, \{14\text{-}17\}, \{18\}, \{19\}, \{20\}\} \tag{2.41}$$

$$F_{510} = \{\{1, 2\}, \{3\text{-}6\}, \{7\text{-}9\}, \{10\text{-}13\}, \{14, 15\}, \{16\}, \{17\}, \{18\}, \{19\}, \{20\}\} \tag{2.42}$$

$$F_{512} = \{\{1\}, \{2\}, \{3\text{-}6\}, \{7\text{-}9\}, \{10\text{-}13\}, \{14, 15\}, \{16\}, \{17\}, \{18\}, \{19\}, \{20\}\} \tag{2.43}$$

For instance, F_{43} in (2.39) is composed of 4 clusters with $S_1^{43} = \{1, 2, 3, 4, 5, 6\}$, $S_2^{43} = \{7, 8, 9, 10, 11, 12, 13\}$, $S_3^{43} = \{14, 15, 16, 17, 18\}$ and $S_4^{43} = \{19, 20\}$. When $F_m = F_{43}$ is chosen as the sampling cluster formation, the resulting expected number of false classifications is $E_f = 1.505$, and the required number of tests is 3, as seen in Figure 2.9(a) and (b). For the sampling cluster formation choices which are not one of the six cluster formations listed above, these six cluster formations can be chosen to minimize the expected number of false classifications while keeping the required number of tests constant. For instance, all choices of m between $m = 2$ and $m = 42$ result in the required number of three tests as $m = 43$ but yield a larger E_f than what $m = 43$ yields.

For this system as well, we calculate the average number of required tests for Hwang's generalized binary splitting algorithm by using the results of [7, 27, 28] as in the first simulation (by implementing and running these algorithms 1000 times where we realize the infection status of the population for each iteration) and find that the average number of required tests is 16.4 in this case. Similar to the first simulation environment, the binary splitting algorithm presented originally in [5] which does not require the exact number of infections, cannot perform better than individual testing.

2.7 Conclusions

In this chapter, we introduced a novel infection spread model that consists of a random patient zero and a random connection graph, which corresponds to a non-identically distributed and correlated (non i.i.d.) infection status for individuals. We proposed a family of group testing algorithms, which we call *two-step sampled group testing algorithms*, and characterized their optimal parameters. We determined the optimal sampling function selection, derived expected false classifications, and proposed $\mathcal{F}$*-separable non-adaptive group tests*, which is a family of zero-error non-adaptive group testing algorithms that exploit a given random cluster formation structure. For a specific family of random cluster formations, which we call *exponentially split cluster formation trees*, we calculated the expected number of false classifications and the required number of tests explicitly, by using our general results, and showed that our two-step sampled group testing algorithm outperforms all non-adaptive tests that do not exploit the cluster formation structure and Hwang's adaptive generalized binary splitting algorithm, even though our algorithm is non-adaptive and we ignore our gain from the first step of our two-step sampled group testing algorithm. Moreover, we characterized the computational complexities of constructing the proposed algorithms. Finally, our work has an important implication: in contrast to the prevalent belief about group testing that it is useful only when the infections are rare, our group testing algorithm shows that a considerable reduction in the number of required tests can be achieved by using the prior probabilistic knowledge about the connections between the individuals, even in scenarios

with a significantly high number of infections.

2.8 Appendix

Theorem 2.1 *In a two-step sampled group testing algorithm with the given sampling cluster formation F_m and the sampling function M over a cluster formation tree structure defined by $\mathcal{F}$ and p_F, with uniform patient zero distribution p_Z over $[n]$, the expected number of false classifications given $F = F_\alpha$ is*

$$E_{f,\alpha} = \sum_{i \in [\sigma_m]} \left(\frac{|S^\alpha(M_i)|}{n} \cdot |S_i^m \backslash S^\alpha(M_i)| + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} \frac{|S_j^\alpha|^2}{n} \right) \tag{2.44}$$

and the expected number of false classifications is

$$E_f = \sum_{\alpha > m} p_F(F_\alpha) E_{f,\alpha} \tag{2.45}$$

where $S^\alpha(M_i)$ is the subset in the partition F_α which contains the ith selected individual.

Proof: For the sake of simplicity, we denote the subset in partition F_α that contains the ith selected individual by $S^\alpha(M_i)$. We start our calculation with the conditional expectation where $F = F_\alpha$ is given. Observe that an error occurs, in the second step of the decoding process, only if F_m is at a higher level of the cluster formation tree than the realization of $F = F_\alpha$ and the true infected cluster $K = S_\gamma^\alpha$ is merged at the level F_m, i.e., $\alpha > m$ and $S_\gamma^\alpha \notin F_m$. Since there is exactly one true infected cluster, which is at level F_α, false classifications only happen in the set S_θ^m that contains S_γ^α.

Now, we know that for the given sampling function M, the θth selected individual is selected from the set S_θ^m and in the second step of the decoding phase, its infection status is assigned to all of the members of the set S_θ^m. Therefore, the members of the difference set $S_\theta^m \backslash S^\alpha(M_\theta)$ are falsely classified if the set $S^\alpha(M_\theta)$ is the true infected set. In that case, all members of S_θ^m would be classified as infected while only the subset of them, which is $S^\alpha(M_\theta)$ were infected. On the other hand, when the cluster of the selected individual at level F_α is not infected, i.e., the infected cluster is a subset of $S_\theta^m \backslash S^\alpha(M_\theta)$, then only the infected cluster is falsely identified since all of the members of S_θ^m are classified as non-infected. Thus, we have the following conditional expected number of false classifications when $F = F_\alpha$ is given, where $p_{S_i^j}$ denotes the probability of the set S_i^j being infected

$$E_{f,\alpha} = \sum_{i \in [\sigma_m]} \left(p_{S^\alpha_{M_i}} |S_i^m \backslash S^\alpha(M_i))| + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} p_{S_j^\alpha} |S_j^\alpha| \right) \tag{2.46}$$

$$= \sum_{i \in [\sigma_m]} \left(\frac{|S^\alpha(M_i)|}{n} \cdot |S_i^m \backslash S^\alpha(M_i)| + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} \frac{|S_j^\alpha|^2}{n} \right) \tag{2.47}$$

where (2.47) follows from the uniform patient zero assumption. Finally, since false classifications occur only when $\alpha > m$, we have the following expression for the expected number of false classifications

$$E_f = \sum_{\alpha > m} p_F(F_\alpha) E_{f,\alpha} \tag{2.48}$$

concluding the proof. ■

Theorem 2.2 *For sampling cluster formation F_m, the optimal choice of M that minimizes the expected number of false classifications is*

$$M_i = \arg\min_{k \in S_i^m} \beta_m(k) \tag{2.49}$$

where M_i is the ith selected individual. Moreover, the number of required tests is constant and is independent of the choice of M.

Proof: We first prove the second part of the theorem, i.e., that the choice of M does not change the required number of tests. In a cluster formation tree structure, when we sample exactly one individual from each subset S_i^m, $\mathcal{P}(K_M)$ contains single element subsets of selected individuals, since when $F = F_m$ we have exactly one infected individual that can be any one of these individuals with positive probability. Now consider the cluster formation F_{m-1}. Since it is a cluster formation tree structure, there must be at least one S_i^{m-1} such that, $S_i^{m-1} = S_j^m \cup S_k^m,\ S_j^m \neq S_k^m$, which means that, $\mathcal{P}(K_M)$ must contain the set of selected individuals from S_k^m and S_j^m as well, because of the fact that in the case of $F = F_{m-1}$, these individuals can be infected simultaneously. Similarly, when moving towards the top node of the cluster formation tree (i.e., F_1), whenever we observe a merging, we must add the corresponding union of the subsets of individuals to $\mathcal{P}(K_M)$, which is the set of all possible infected sets for the selected individuals M. Thus, the structure of distinct sets of possible infected individuals does not depend on the indices of the sampled individuals within each S_i^m, but depends on the given $\mathcal{F}$ and F_m, completing the

proof of the second part of the theorem.

We next prove the first part of the theorem, i.e., we prove that selecting the individual that has the minimum $\beta_m(k)$ value for each S_i^m results in the minimum expected number of false classifications and thus, it is the optimal choice. First, recall that, by definition, M depends on F_m, and thus, we design sampling function M for a given F_m. Now, recall the expected number of false classifications stated in (2.44)-(2.45). Designing a sampling function that minimizes E_f for a given F_m can be done as follows. From (2.44)-(2.45),

$$\min_M E_f = \min_M \left\{ \sum_{\alpha:m<\alpha} p_F(F_\alpha) \sum_{i\in[\sigma_m]} \left(\frac{|S^\alpha(M_i)|}{n} \times |S_i^m \backslash S^\alpha(M_i)| + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} \frac{|S_j^\alpha|^2}{n} \right) \right\} \tag{2.50}$$

$$= \frac{1}{n} \sum_{i\in[\sigma_m]} \min_M \left\{ \sum_{\alpha:m<\alpha} p_F(F_\alpha) \left(|S^\alpha(M_i)| \times |S_i^m \backslash S^\alpha(M_i)| + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(M_i)} |S_j^\alpha|^2 \right) \right\} \tag{2.51}$$

$$= \frac{1}{n} \sum_{i\in[\sigma_m]} \left(\sum_{\alpha:m<\alpha} p_F(F_\alpha) \left(|S^\alpha(k_i^*)| \times |S_i^m \backslash S^\alpha(k_i^*)| + \sum_{S_j^\alpha \subseteq S_i^m \backslash S^\alpha(k_i^*)} |S_j^\alpha|^2 \right) \right) \tag{2.52}$$

where $k_i^* = \arg\min_{k\in S_i^m} \beta_m(k)$, and (2.52) is the minimum value of the expected number of false classifications for given F_m. The sampling function M defined in (2.49) achieves the minimum and thus, it is optimal, completing the proof of the first part of the theorem. ■

Theorem 2.3 *For given $\mathcal{F}$ and F_m for $m < f$, the number of required tests for an*

$\mathcal{F}$-separable non-adaptive group test, i.e., the number of rows of the test matrix $\boldsymbol{X}$, must satisfy

$$T \geq \max\left\{\max_{j\in[\sigma_m]} (\lambda_{S_j^m}+1),\ \lceil\log_2(\lambda_m+1)\rceil\right\} \tag{2.53}$$

with addition of 1's removed in (2.53) for the special case of $m = f$.

Proof: First, we have that each unique node (nodes that represent a unique subset S_i^j) represents a unique possibly infected set K_M where each result vector must be unique as well. Therefore, in total, we must have at least λ_m unique vectors. Furthermore, when $m < f$, it is possible that the infected set among the sampled individuals is the empty set. Thus, we have to reserve the zero vector for this case as well. Therefore, the total number of tests must be at least $\lceil\log_2(\lambda_m+1)\rceil$ in general, with an exception of $m = f$ case, where we can assign the zero vector to one of the nodes and may achieve $\lceil\log_2(\lambda_m)\rceil$.

Second, assume that for any node j at an arbitrary level F_i, $i < m$, the set of indices of the positions of 1's must contain the set of indices of the positions of 1's of the descendants of node j. Moreover, since all nodes that split must be assigned a unique vector, Hamming weights of the vectors must strictly decrease as we move from an ancestor node to a descendant at each level. Considering the fact that the ancestor node at the top level can have Hamming weight at most T and the nodes at the level F_m must be assigned a vector which has Hamming weight at least 1, including the node that has the most unique ancestor nodes, T must be

at least $\max_{j\in[\sigma_m]}(\lambda_{S_j^m}+1)$. Similar to the first case, when $m=f$, we can have zero vector assigned to one of the bottom level nodes, and thus, we can have T at least $\max_{j\in[\sigma_m]}\lambda_{S_j^m}$. ■

Theorem 2.4 *For an f level exponentially split cluster formation tree, at level f, there exists an $\mathcal{F}$-separable test matrix, $\boldsymbol{X}$, with not more than $\frac{4}{3}f$ rows, i.e., an upper (achievable) bound for the number of required tests is $\frac{4}{3}(\log_2 n+1)$ for n individuals. Conversely, this is also the capacity order-wise since the number of required tests must be greater than f.*

Proof: By using the converse in Theorem 2.3, we already know that the required number of tests is at least f from (2.24) since there are $\lambda_f=2^f-1$ unique nodes and also $\lambda_{S_i^f}+1=f$ for every subset S_i^f. This proves the converse part of the theorem.

In order to satisfy the sufficient conditions for the existence of an $\mathcal{F}$-separable matrix, each node in the tree must be represented by a T length vector of sufficient Hamming weight, so that i) every descendant can be represented by a unique vector with positions of 1's being the subsets of the positions of 1's of their ancestor nodes, and ii) OR of vectors that are all descendants of a node must be equal to the vector of the ancestor node. In our proof, we show that, for exponentially split cluster formation trees, it is sufficient to check that we have a sufficient number of rows in $\boldsymbol{X}$ to uniquely assign vectors to the bottom level nodes, i.e., the subsets S_i^f at level F_f.

First, as we stated above, from the converse in Theorem 2.3, an $\mathcal{F}$-separable

test matrix of an exponentially split cluster formation tree with f levels must have at least f rows. However, for exponentially split cluster formation trees, this converse is not achievable: There are 2^{f-1} nodes at level f but $\binom{f}{1}$ binary vectors with Hamming weight 1. Since for $f > 3$, $\binom{f}{1}$ is less than 2^{f-1}, we cannot assign distinct Hamming weight 1 vectors to the bottom level nodes. Thus, we need vectors with lengths longer than f. Now, assume that an achievable $\mathcal{F}$-separable test matrix has $f+k$ rows, where k is a non-negative integer. Our objective in the remainder of the proof is to characterize this k in terms of f.

We argue that if the number of nodes at the bottom level, which is equal to 2^{f-1}, is less than $\sum_{i=1}^{k+1}\binom{f+k}{i}$ then we can find an achievable $\mathcal{F}$-separable test matrix, i.e.,

$$\sum_{i=1}^{k+1}\binom{f+k}{i} \geq 2^{f-1} \tag{2.54}$$

is a sufficient condition for the existence of an achievable $\mathcal{F}$-separable test matrix for a given (f,k) pair. Minimum k that satisfies (2.54) will result in the minimum number of required tests $f+k$. In our construction, we assign each node at level F_i a unique vector with Hamming weight $f+k+1-i$, except for the bottom level F_f. Since each node is assigned a unique vector, when moving from a level to one level down, descendant nodes must be assigned vectors that have Hamming weight of at least 1 less than their ancestor node. At the bottom level, we use the remaining vectors with Hamming weight less than or equal to $k+1$. We choose minimum such k for this construction, resulting in the minimum number of tests.

Before proving the achievability of this above construction, we first analyze the minimum k that satisfies (2.54) in terms of f. We state and prove in Theorem 2.4 in the Appendix in Section 2.8 that $k = f/3$ satisfies (2.54), giving an upper bound for the minimum k, thus finalizing the first part of the achievability proof. This, in turn, shows that we can use all vectors of Hamming weight 1 through $k+1$ in the bottom level to represent all 2^{f-1} nodes at that level.

Next, we show that for the upper levels, our construction is achievable, i.e., we can find sufficiently many vectors of corresponding Hamming weights. By using Theorem 2.5 in the Appendix in Section 2.8, and the fact that for $k \leq f/3$, when $f \geq 13$, we have

$$\binom{f+k}{k+2} \geq 2^{f-2} \tag{2.55}$$

which implies that, we can find unique vectors of Hamming weight $k+2$, to assign to the nodes at level F_{f-1} (one level up from the bottom level). For the remaining levels below $\lceil (f+k)/2 \rceil$, we have $\binom{f+k}{i} > \binom{f+k}{i+1}$ and the number of nodes decreases by half as we move upwards on the tree. Thus, we can find unique vectors to represent the nodes by increasing the Hamming weights by 1 at each level, which is the minimum increase of Hamming weights while moving upwards on the tree. For the remaining nodes, which are above the level $\lceil (f+k)/2 \rceil$, we can use the lower bound for the binomial coefficient,

$$\binom{f+k}{i} \geq \left(\frac{f+k}{i}\right)^i \geq 2^i \tag{2.56}$$

to show that there are unique vectors of required weights at those levels as well.

Thus, there are sufficiently many unique vectors of appropriate Hamming weights at every level. Finally, we have to check whether or not there is a sufficient number of unique vectors for every subtree of descendants of each node. In exponentially split cluster formation trees, due to the symmetry of the tree, any descendant subtrees of each node are again an exponentially split cluster formation tree. If we assume that k, where the number of rows of $\boldsymbol{X}$ is equal to $f+k$, satisfies (2.54) with k being the minimum such number, then every descendant subtree below the top level has parameters $(f-i,k)$ and we show in Theorem 2.4 in the Appendix in Section 2.8 that they also satisfy the condition (2.54). For f values that are below the corresponding threshold in our proof steps (e.g., $f \geq 13$ threshold before (2.55) above), manual calculations yield the desired results. This proves the achievability part of the theorem. ■

Lemma 2.4 *Minimum k that satisfies*

$$\sum_{i=1}^{k+1} \binom{f+k}{i} \geq 2^{f-1} \tag{2.57}$$

is upper bounded by $f/3$.

Proof: We prove the statement of the lemma by showing that the pair $(f,k) = (f, f/3)$ satisfies (2.57). We first consider the left hand side of (2.57) when f is

incremented by 1 for fixed k, and write it as

$$\sum_{i=1}^{k+1} \binom{f+k+1}{i} = 2\sum_{i=1}^{k+1} \binom{f+k}{i} + 1 - \binom{f+k}{k+1} \tag{2.58}$$

which follows by using the identity $\binom{a}{b} = \binom{a-1}{b-1} + \binom{a-1}{b}$.

Second, we prove the following statement for $k \geq 1$,

$$\sum_{i=1}^{k+1} \binom{4k}{i} \geq 2^{3k-1} \tag{2.59}$$

Note that, when $k = f/3$, (2.59) is equivalent to (2.57) for f values that are divisible by 3. For f values that are not divisible by 3, since the pairs $(f-1, k)$ and $(f-2, k)$ satisfy (2.57) when the pair (f, k) satisfies (2.57), by (2.58), it suffices to prove the statement in (2.59).

We prove (2.59) by induction on k. For $k = 1$, the inequality holds. Assume that the inequality holds for a $k \geq 1$, then we show that it also holds for $k+1$. In the lines below, we use the identity $\binom{a}{b} = \binom{a-1}{b-1} + \binom{a-1}{b}$ recursively,

$$\sum_{i=1}^{k+2} \binom{4k+4}{i} = \sum_{i=1}^{k+2} \binom{4k+3}{i} + \sum_{i=1}^{k+2} \binom{4k+3}{i-1} \tag{2.60}$$

$$= \sum_{i=1}^{k+2} \binom{4k+2}{i} + \sum_{i=1}^{k+2} \binom{4k+2}{i-1} + 1 + \sum_{i=1}^{k+1} \binom{4k+2}{i} + \sum_{i=1}^{k+1} \binom{4k+2}{i-1} \tag{2.61}$$

$$\vdots$$

$$=9\sum_{i=1}^{k+1}\binom{4k}{i}-5\binom{4k}{k+1}+\binom{4k}{k+2}+4\binom{4k}{k-1}+5\binom{4k}{k-2}+A \tag{2.62}$$

$$=9\sum_{i=1}^{k+1}\binom{4k}{i}-\frac{2k+11}{k+2}\binom{4k}{k+1}+4\binom{4k}{k-1}+5\binom{4k}{k-2}+A \tag{2.63}$$

$$=8\sum_{i=1}^{k+1}\binom{4k}{i}-\frac{k+9}{k+2}\binom{4k}{k+1}+\binom{4k}{k}+5\binom{4k}{k-1}+6\binom{4k}{k-2}+A' \tag{2.64}$$

$$=8\sum_{i=1}^{k+1}\binom{4k}{i}+3\binom{4k}{k-2}+A'' \tag{2.65}$$

$$\geq 2^{3k+2} \tag{2.66}$$

where A, A', A'' are positive terms that are $o\left(\binom{4k}{k-2}\right)$, and we use the identity $\binom{a}{b}=\frac{a-b+1}{b}\binom{a}{b-1}$ after equation (2.62) to eliminate the negative $\binom{4k}{k+1}$ term. Inequality (2.66) follows from the induction assumption. This proves the statement for $k+1$ and completes the proof. ■

Lemma 2.5 *When $k \leq \frac{2n-8}{5}$, the following inequality holds*

$$\frac{1}{2}\sum_{i=1}^{k}\binom{n}{i}<\binom{n}{k+1} \tag{2.67}$$

Proof: We prove the lemma by induction over k. First, note that the inequality holds when $k=1$,

$$\frac{1}{2}\binom{n}{1}<\binom{n}{2} \tag{2.68}$$

Then, assume that the statement is true for k. Now we check the statement for $k+1$,

$$\frac{1}{2}\sum_{i=1}^{k+1}\binom{n}{i} < \frac{3}{2}\binom{n}{k+1} \tag{2.69}$$

$$\leq \frac{n-k-1}{k+2}\binom{n}{k+1} \tag{2.70}$$

$$= \binom{n}{k+2} \tag{2.71}$$

where (2.69) follows from the induction assumption and (2.70) is because $k \leq \frac{2n-8}{5}$. This proves the statement for $k+1$ and completes the proof. ■

CHAPTER 3

Dynamic Infection Spread Model Based Group Testing

3.1 Introduction

In this chapter, we consider dynamic testing algorithms over discrete time for a dynamic infection spread model with fixed, limited testing capacity at each time instant, where a full identification is not possible. In our system, test results are available immediately, and thus, the disease spread is not due to the delay between applying tests and receiving test results, but rather due to the limited testing capacity at each time instant. We follow a dynamic infection spread model, which is inspired by the well-known SIR model where the individuals are divided into three groups: susceptible individuals (S), non-isolated infections (I), and isolated infections (R), i.e., recovered individuals in the classical SIR model. We do not assume a community structure in our system. We initialize our system by introducing the initial infections, and after that, at each time instant, infection is spread by infected non-isolated individuals to the susceptible individuals. Meanwhile, at each time instant, after the infection spread phase, the testing phase is performed, where a limited number of T tests are performed to detect a number of infections in the

system. In our system, the objective is not to minimize the number of required tests to identify everyone at each time instant, but to control the infection spread either as soon as possible or with a minimum number of people that got infected throughout the process, by using the given, limited, testing capacity T at each time instant.

We analyze the average case performance of our system, i.e., the expected values of the number of susceptible individuals, and non-isolated and isolated infections over time, which are random processes. For symmetric and converging algorithms, we state a general analytical result for the expected number of susceptible individuals in the system when the infection is brought under control, which is the time when there is no non-isolated infection left in the system. We present two dynamic algorithms: dynamic individual testing and dynamic Dorfman-type group testing algorithm. We provide weak versions of these two algorithms and use our general result to obtain a lower bound on the expected number of susceptible individuals when the infection is under control. Finally, we run simulations to get numerical results of our proposed algorithms for different sets of parameters.

3.2 System Model

We consider a population of n individuals whose infection statuses change over time. The time dimension t is discrete in our system, i.e., $t \in \{0, 1, 2, \ldots\}$. Similar to the classical discrete SIR model, the population consists of three distinct subgroups: susceptible individuals who are not infected but can get infected by infected in-

dividuals (S), infected individuals who can infect the susceptible individuals (I), and isolated individuals who were infected, have been detected via performed tests and isolated indefinitely (R) [1]. Let $U_i(t)$ denote the infection status of individual i at time t, where 1 represents being infected, 0 represents not being infected and 2 represents being isolated. At the beginning ($t = 0$), we introduce the initial infections in the system, independently with probability p, where $U_i(0)$ is a Bernoulli random variable with parameter p. Random variables $U_i(0)$ are mutually independent for $i \in [n]$. Let $\alpha(t)$ denote the number of susceptible individuals at time t, $\lambda(t)$ denote the number of non-isolated infected individuals at time t, and $\gamma(t)$ denote the number of isolated individuals at time t. Starting from $t = 1$, each time instant consists of two phases: the infection spread phase and testing phase, in the respective order.

Infection Spread Phase: Infected individuals spread the infection to the susceptible members of the population. At each time instant, starting from $t = 1$, the infection spreads independently across the individuals: Each infected individual can infect each susceptible individual with probability q, independent across both infected individuals and susceptible individuals. Isolated individuals cannot infect others and their infection status cannot change after they are isolated. Thus, the probability of the event that individual i gets infected by another individual j at time $t \geq 1$ is equal to

$$qP\left(U_j(t-1) = 1, U_i(t-1) = 0\right) \quad for \quad i, j \in [n]. \tag{3.1}$$

[1] These are called recovered (R) individuals in the SIR model; we call them isolated individuals. As they are isolated indefinitely, they are recovered eventually.

Testing Phase: At each time instant starting from $t = 1$, T tests can be performed to the individuals. Note that the testing capacity T is a given parameter and thus, in contrast to the classical group testing systems, we do not seek to minimize the number of performed tests for full identification of the infection status of the population but aim to efficiently perform T tests at each time instant to identify and isolate as many infections as possible to control the infection spread. Here, performed tests can be group tests, and we define the $T \times n$ binary test matrices, $\boldsymbol{X}(t)$, which specify the pooling scheme for the tests at each time t. For each time instant $t \geq 1$, we have the test result vectors $y(t)$, which are equal to

$$y_i(t) = \bigvee_{j \in [n]} \boldsymbol{X}_{ij}(t) \mathbb{1}_{\{U_j(t)=1\}}, \quad i \in [T] \tag{3.2}$$

where $y_i(t)$ denotes the ith test result at time t, $\boldsymbol{X}_{ij}(t)$ denotes the ith row, jth column of the test matrix $\boldsymbol{X}(t)$.

Note that, since the previous test matrices and test results are available while designing these test matrices, $\boldsymbol{X}(t)$ can depend on the previous test results $y(t')$ for $t' < t$. We assume that when tests are performed at some time instant t', the test results $y(t')$ will be available before the infection spread phase at time $t' + 1$. Thus, after the test results are available, detected infections are isolated immediately, i.e., if the ith individual is detected to be infected during the testing phase at time t', then $U_i(t') = 2$. Recall that, after an infected individual is isolated at some time t', they cannot infect others at times greater than t' and their infection status cannot change, i.e., $U_i(t) = 2$ for $t \geq t'$.

Testing Policy: A testing policy π is an algorithm that specifies how to allocate the given testing capacity T for each time instant until the infection is under control. We define $\bar{t}$ to be the time when $U_i(\bar{t}) \neq 1$ for all individuals $i \in [n]$ for the first time and we say that the infection is under control at $\bar{t}$. Note that, after $\bar{t}$, the infection statuses of the individuals cannot change and the steady state is achieved: They are either isolated ($U_i(t) = 2$) or non-infected ($U_i(t) = 0$). Since we do not consider re-entries of recoveries to the population, the infection spread is under control when all infections in the system are isolated. Otherwise, the infection may keep spreading to susceptible individuals by non-detected infections.

Performance Metrics: The main objective is to bring the infection spread under control by detecting and isolating each infected individual by performing at most T tests at each time instant. Note that, meanwhile, infection keeps spreading, and thus, detecting the infection status of an individual to be negative does not imply that they are identified for the rest of the process; they can get infected in later time instants. As defined, $\bar{t}$ is the time that the infection is under control, and when the system has reached that state, further testing of the individuals is unnecessary. Therefore, there are two metrics to measure the performance of a testing policy π: The time $\bar{t}$ when the infection is brought under control and the total number of isolated individuals when the infection is under control while comparing the performances of the testing policies, earlier infection control time $\bar{t}$ and less number of total infections at the time of infection control $\gamma(\bar{t})$ are favored. Proposed algorithms may not simultaneously improve both metrics: One policy may bring the infection spread under control fast (i.e., low $\bar{t}$) but may result in a high number of

total infections (i.e., high $\gamma(\bar{t})$) while another policy may bring the infection spread under control slowly but with a lower number of total infections.

3.3 Proposed Algorithms and Analysis

In this section, we propose two algorithms and analyze their performances. The first algorithm does not utilize the group testing approach and it is based on the idea of dynamically and individually testing the population. The second algorithm consists of a group testing approach at each time instant, similar to the original idea of Dorfman [1] in a dynamic setting. Before stating these two algorithms and further analyzing their performances individually, we first state general results.

Symmetric and Converging Dynamic Testing Algorithms: In our analysis, we focus on *symmetric and converging dynamic testing algorithms*, which satisfy the symmetry criterion,

$$P(U_i(t) = k) = P(U_j(t) = k), \quad i, j \in [n], \quad k \in \{0, 1, 2\} \quad t \geq 0 \tag{3.3}$$

and *convergence* criterion,

$$\lim_{t \to \infty} P(U_i(t) = 1) = o(1/n), \quad i \in [n] \tag{3.4}$$

Furthermore, we assume that the probability of an individual not being identified in the tests at time t, denoted by $p'(t)$, only depends on the testing capacity T, $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$. Note that, $\alpha(t) + \lambda(t) + \gamma(t) = n$ for all time instants t.

Infection Spread Probability: We consider $q = o(1/n)$ for the infection spread probability q. This is a practical assumption since q is the probability of the event of infection spread that is realized independently for every element of the set product of the infected individuals and susceptible individuals, at each time instant.

We analyze the long-term behavior of the system in the average case, i.e., we focus on the terms $E[\alpha(t)]$, $E[\lambda(t)]$ and $E[\gamma(t)]$ when t is large enough.

Lemma 3.1 *When a symmetric and converging dynamic testing algorithm is implemented,*

$$\lim_{t\to\infty} E[\lambda(t)] = o(1) \tag{3.5}$$

and thus, the system approaches to steady state, in the average case.

Proof: Note that all three system functions $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$ can be written as the summation of n indicator functions

$$E[\lambda(t)] = E\left[\sum_{i=1}^{n} \mathbb{1}_{\{U_i(t)=1\}}\right] \tag{3.6}$$

$$= \sum_{i=1}^{n} E\left[\mathbb{1}_{\{U_i(t)=1\}}\right] \tag{3.7}$$

$$= \sum_{i=1}^{n} P(U_i(t) = 1) \tag{3.8}$$

$$= nP(U_i(t) = 1) \tag{3.9}$$

which results in $\lim_{t\to\infty} E[\lambda(t)] = no(1/n)$, due to converging algorithm assumption (3.4), which is equal to $o(1)$. ■

Note that, when the system reaches a state where $\lambda(t) = 0$, then there will not

be a further change in the infection statuses of the individuals, i.e., the infection will be under control. The following lemma is useful for the justification of the average case analysis of our system.

Lemma 3.2 *When a symmetric and converging dynamic testing algorithm is implemented, we have* $\lim_{t\to\infty} P(\lambda(t) > \epsilon) = o(1)$ *for arbitrarily small, constant,* $\epsilon \in \mathbb{R}$.

Proof: Since $\lambda(t) \geq 0$ for all $t \geq 0$, we can apply Markov's inequality,

$$\lim_{t\to\infty} P(\lambda(t) > \epsilon) \leq \lim_{t\to\infty} \frac{E[\lambda(t)]}{\epsilon} \tag{3.10}$$

$$= o(1) \tag{3.11}$$

where (3.10) follows from the fact that $P(\lambda(t) > \epsilon) \leq \frac{E[\lambda(t)]}{\epsilon}$ for all $t \geq 0$, and (3.11) follows from the result of Lemma 3.1. ■

The focus of our analysis is to give a lower bound for the number of susceptible individuals (who have never got infected throughout the process) when the infection is brought under control, in the average-case. To analyze the long-term behavior of $E[\alpha(t)]$, we have to analyze the long-term behavior of $P(U_i(t) = 0)$. A direct calculation of this probability is not analytically tractable, however, by conditioning on $\lambda(t-1)$, we give a recursive asymptotic calculation. Before stating the recursive relation, we first prove a lemma that will be useful.

Lemma 3.3 *For* $q = o(1/n)$ *and for all* $t \geq 0$, *we have*

$$cov\left(P(U_i(t) = 0|\lambda(t)), (1-q)^{\lambda(t)}\right) \approx 0 \tag{3.12}$$

Proof: For the proof, we use the covariance inequality, i.e., $|cov(X,Y)| \leq \sqrt{var(X)var(Y)}$ which is a direct application of the Cauchy-Schwarz inequality, applied to the random variables $X - E[X]$ and $Y - E[Y]$. Using the covariance inequality, we have

$$|cov\left(P(U_i(t)=0|\lambda(t)), (1-q)^{\lambda(t)}\right)| \leq \sqrt{var(P(U_i(t)=0|\lambda(t)))var((1-q)^{\lambda(t)})} \tag{3.13}$$

$$\leq \sqrt{var((1-q)^{\lambda(t)})} \tag{3.14}$$

$$= \sqrt{E[(1-q)^{2\lambda(t)}] - (E[(1-q)^{\lambda(t)}])^2} \tag{3.15}$$

$$\approx \sqrt{(1-q)^{E[2\lambda(t)]} - ((1-q)^{E[\lambda(t)]})^2} \tag{3.16}$$

$$= 0 \tag{3.17}$$

where (3.14) follows from the fact that the random variable $P(U_i(t) = 0|\lambda(t))$ is bounded above by 1 and below by 0, and (3.16) follows from the linear approximation of the function $(1-q)^x$ for small $q = o(1/n)$ and $\lambda(t)$ that is bounded above by n.

■

Lemma 3.4 *When a symmetric and converging dynamic testing algorithm is implemented, we have*

$$P(U_i(t)=0) \approx (1-p)(1-q)^{n\sum_{j=0}^{t-1} P(U_1(j)=1)} \tag{3.18}$$

Proof: Conditioned on $\lambda(t-1)$, we have the following recursive relation for $P(U_i(t) =$

0)

$$P(U_i(t)=0) = E[P(U_i(t)=0|\lambda(t-1))] \tag{3.19}$$

$$= E[P(U_i(t-1)=0|\lambda(t-1))(1-q)^{\lambda(t-1)}] \tag{3.20}$$

$$\approx E[P(U_i(t-1)=0|\lambda(t-1))]E[(1-q)^{\lambda(t-1)}] \tag{3.21}$$

$$= P(U_i(t-1)=0)E[(1-q)^{\lambda(t-1)}] \tag{3.22}$$

$$\approx P(U_i(t-1)=0)(1-q)^{E[\lambda(t-1)]} \tag{3.23}$$

$$= P(U_i(t-1)=0)(1-q)^{\sum_{j=1}^{n} P(U_j(t-1)=1)} \tag{3.24}$$

$$= P(U_i(t-1)=0)(1-q)^{nP(U_1(t-1)=1)} \tag{3.25}$$

where (3.21) follows from Lemma 3.3, (3.23) follows from the linear approximation of the function $(1-q)^x \approx 1-qx$, and (3.25) follows from the symmetry criterion of the implemented algorithm. Recursively using the result in (3.25) and the initial value $P(U_i(0)=0) = (1-p)$ yields the desired result. ■

To complete our analysis and give a lower bound for the expected number of susceptible individuals when the infection is under control, we further need to focus on $P(U_i(t)=1)$. Similar to the case of $P(U_i(t)=0)$, a direct calculation is not analytically tractable. However, we have a recursive relation when conditioned on $\lambda(t-1)$, and we obtain the following lemma.

Lemma 3.5 *When a symmetric and converging dynamic testing algorithm is implemented, and* $cov\left(P(U_i(t)=0|\lambda(t)), p'_{\lambda(t)}(t+1)\right)$ *and* $cov\left(P(U_i(t)=1|\lambda(t)), p'_{\lambda(t)}(t+1)\right)$

are arbitrarily small [2] *for all* $t \geq 0$*, we have*

$$P(U_i(t) = 1) \approx p(1 + nq(1-p))^t \prod_{j=1}^{t} p'(j) \tag{3.26}$$

where the conditional probability of an individual not being identified in the tests at time t *given* $\lambda(t-1)$ *is denoted by* $p'_{\lambda(t-1)}(t)$*.*

See the Appendix in Section 3.6 for the proof of Lemma 3.5. Combining the results of Lemmas 3.4 and 3.5, we have the following result.

Theorem 3.1 *When a symmetric and converging dynamic testing algorithm is implemented and vanishing covariance constraints in Lemma 3.5 are satisfied for all* $t \geq 0$*, we have*

$$E[\alpha(t)] \approx n(1-p)(1-q)^{np \sum_{i=0}^{t-1} \left((1+nq(1-p))^i \prod_{j=1}^{i} p'(j) \right)} \tag{3.27}$$

Proof: Expressing $\alpha(t)$ in terms of the corresponding indicator random variables and using the symmetry criterion and results of Lemmas 3.4 and 3.5 yields

$$E[\alpha(t)] = E\left[\sum_{i=1}^{n} \mathbb{1}_{\{U_i(t)=0\}} \right] \tag{3.28}$$

[2] In both of the covariances in the lemma statement, we have probabilities that are conditioned on the number of infected and non-isolated individuals at time t. Note that $P(U_i(t) = 0|\lambda(t))$ and $P(U_i(t) = 1|\lambda(t))$ are the probabilities that the individual i is susceptible, and infected and non-isolated, respectively. Since we only consider symmetric and converging dynamic testing algorithms, these probabilities are symmetric across all individuals for every time instant t. Note that since the standard deviations of each of these random variables can also be arbitrarily small, the arbitrarily small covariance constraints in the lemma statement do not directly imply a weak correlation between these random variables.

$$= \sum_{i=1}^{n} E[\mathbb{1}_{\{U_i(t)=0\}}] \tag{3.29}$$

$$= \sum_{i=1}^{n} P(U_i(t) = 0) \tag{3.30}$$

$$= nP(U_i(t) = 0) \tag{3.31}$$

$$\approx n(1-p)(1-q)^{np \sum_{i=0}^{t-1} \left((1+nq(1-p))^i \prod_{j=1}^{i} p'(j) \right)} \tag{3.32}$$

which is the desired result. ■

Our main result Theorem 3.1 is a general result and holds for the symmetric and converging dynamic testing algorithms as long as they satisfy the vanishing covariance conditions that we state in Lemma 3.5. In the remainder of this section, we propose and describe two dynamic testing algorithms and analyze their performance.

3.3.1 Dynamic Individual Testing Algorithm

In the dynamic individual testing algorithm, we do not utilize the group testing approach, and uniformly randomly select T individuals to individually test at each time instant $t \geq 1$, from the non-isolated individuals.

To analyze the performance of our dynamic individual testing algorithm, we use the general result of Theorem 3.1. First, we show that the dynamic individual testing algorithm satisfies the symmetry and convergence criteria in (3.3) and (3.4).

Since the process of selection of individuals to be tested is repeated at each time instant with uniformly random selections, as well as the infection spread pro-

cess, the dynamic individual testing algorithm is symmetric. We show that the dynamic individual testing algorithm also satisfies the convergence criterion (3.4) in the following lemma. For the range of the testing capacity T, we focus on the case of $T < n$, since when $T \geq n$, at one time instant, everyone can be tested individually and the infection will be under control trivially.

Lemma 3.6 *For constant T and n, the dynamic individual testing algorithm satisfies the convergence criterion*

$$\lim_{t\to\infty} P(U_i(t) = 1) = 0, \quad i \in [n] \tag{3.33}$$

Proof: First, the probability that an infected individual is detected at a time instant t, denoted by $1 - p'(t)$ is

$$1 - p'(t) = E[1 - p'_{\gamma(t-1)}(t)] \tag{3.34}$$

$$= E\left[\frac{T}{n - \gamma(t-1)}\right] \tag{3.35}$$

$$\geq \frac{T}{n} \tag{3.36}$$

where $p'_{\gamma(t-1)}(t)$ denotes the probability of the conditional event that an infected individual is not detected at the time instant t given $\gamma(t-1)$. Now, since the conditional events of detection given that the individual is infected are independent across time due to the uniform random selection of tested individuals at each time

instant, and the fact that

$$\sum_{i=1}^{\infty}(1-p'(i)) \geq \sum_{i=1}^{\infty}\frac{T}{n} \tag{3.37}$$

since the right hand side of (3.37) grows to infinity, from the second Borel-Cantelli lemma, the conditional detection event occurs infinitely often, i.e., let D_t denote the event that the individual i is identified at time t, then

$$P(\limsup_{t\to\infty} D_t) = 1 \tag{3.38}$$

which yields the desired result of $\lim_{t\to\infty} P(U_i(t) = 1) = 0$. ■

Next, we consider a weak version of our algorithm, where at each time instant, during the testing phase, instead of selecting T individuals to test from $n - \gamma(t)$ non-isolated individuals, we select T individuals from n individuals, including the isolated ones, whose test results will be negative. For the weak version of the dynamic individual testing algorithm, we have

$$1 - p'(t) = \frac{T}{n}, \quad t > 0 \tag{3.39}$$

which is the identification probability of an individual at time t. Moreover, since it is an upper bound for the identification probability of an individual for the original dynamic individual testing algorithm, we have

$$\lim_{t\to\infty} E[\alpha_{orig}(t)] \geq \lim_{t\to\infty} E[\alpha_{weak}(t)] \tag{3.40}$$

Since the weak dynamic individual testing algorithm is a symmetric and converging algorithm (note that the result of Lemma 3.6 still holds) and due to the fact that $p'(t)$ is constant in the weak dynamic individual testing algorithm, we can directly use the result of Lemma 3.5, due to the fact that the vanishing covariance criteria are already satisfied. Now, using Theorem 3.1, we have the following result for the weak dynamic individual testing algorithm.

Theorem 3.2 *When weak dynamic individual testing algorithm is used and* $(1 - \frac{T}{n})(1 + nq(1-p)) < 1$*, we have*

$$\lim_{t\to\infty} E[\alpha_{weak}(t)] \approx n(1-p)(1-q)^{\frac{np}{1-(1-\frac{T}{n})(1+nq(1-p))}} \tag{3.41}$$

which is a lower bound for $\lim_{t\to\infty} E[\alpha_{orig}(t)]$*, i.e., the limit of the expected number of susceptible individuals for the dynamic individual testing algorithm.*

Proof: The weak dynamic individual testing algorithm satisfies the constraints for using Theorem 3.1. Thus, we can use Theorem 3.1 directly to derive the long-term behavior of the expected number of susceptible individuals by considering the limit of (3.27) for constant $p'(t) = 1 - T/n$. On the other hand, in the case of $(1 - \frac{T}{n})(1 + nq(1-p)) \geq 1$, we have $\lim_{t\to\infty} E[\alpha_{weak}(t)] \approx 0$. ■

3.3.2 Dynamic Dorfman-Type Group Testing Algorithm

In the dynamic Dorfman-type group testing algorithm, we utilize the group testing idea while designing the test matrices at each time instant $t \geq 1$.

At each time instant, the dynamic Dorfman-type group testing algorithm uniformly randomly partitions the set of all non-isolated individuals to equal-sized $T/2$ disjoint sets (with possibly one unequal-sized set if the total number of non-isolated individuals is not divisible by $T/2$). Then, test samples of the individuals are mixed with others in the same group: $T/2$ group tests are performed, and positive and negative groups are determined. Then, among the positive groups, one group (or multiple groups if the sizes of the groups are less than $T/2$, depending on the system parameters) is uniformly randomly selected to be individually tested. $T/2$ individuals from the selected group are uniformly randomly selected and individually tested; here, depending on the parameters, some individuals from the selected group may not be tested, as well as individuals from multiple positive groups may be selected. Detected infections are isolated, and at the next time instant, the whole process is repeated with uniform random selections.

Since the partition selection and individuals within group selection are uniformly random at each time instant, the dynamic Dorfman-type group testing algorithm is symmetric. Similar to Section 3.3.1, we proceed by showing that the dynamic Dorfman-type group testing algorithm satisfies the convergence criterion in (3.4) as well.

Lemma 3.7 *For constant T and n, the dynamic Dorfman-type group testing algorithm satisfies the convergence criterion*

$$\lim_{t\to\infty} P(U_i(t) = 1) = 0, \quad i \in [n] \tag{3.42}$$

Proof: The probability that an individual is identified at a time instant t, which is $1 - p'(t)$, satisfies the following

$$1 - p'(t) \geq \frac{T}{2n} \tag{3.43}$$

since $T/2$ individuals are individually tested at each time instant and due to the symmetry of the infection status in the system and the fact that the individuals are selected from a positive group (or from multiple positive groups), the probability of detection for the dynamic Dorfman-type group testing algorithm, at each time instant, must be higher than uniformly randomized testing of $T/2$ individuals. Now, since the events of identification of individuals are independent across time due to the uniform random selection of tested individuals at each time instant, and the fact that

$$\sum_{i=1}^{\infty}(1 - p'(t)) \geq \sum_{i=1}^{\infty}\frac{T}{2n} \tag{3.44}$$

grows to infinity, we conclude that $\lim_{t\to\infty} P(U_i(t) = 1) = 0$, from the second Borel-Cantelli lemma as in Lemma 3.6. ■

Similar to the dynamic individual testing case, we focus on a weak version of the dynamic Dorfman-type group testing algorithm to provide a lower bound for the expected number of susceptible individuals in the system at the steady state.

In the weak version of the dynamic Dorfman-type group testing algorithm, the results from the $T/2$ group tests are discarded, and it is basically equivalent to the uniformly random individual testing of $T/2$ individuals. Furthermore, the isolated individuals are also included in the testing procedure: n individuals are divided into

groups and then tested at each time instant, rather than only non-isolated individuals, as in the original dynamic Dorfman type group testing algorithm. The probability of identification at time t for the weak dynamic Dorfman-type group testing algorithm, given by $1 - p'(t)$, is always less than the original dynamic Dorfman-type group testing algorithm, due to the discarded $T/2$ group tests and included isolated individuals to the tests. Note that the weak dynamic Dorfman-type group testing algorithm is also symmetric and satisfies the convergent criterion (3.4) since Lemma 3.7 still holds; the lower bound in (3.44) is the detection probability of the weak algorithm. Moreover, since the weak algorithm has a constant value for $p'_{\lambda(t-1)}(t)$, it satisfies the vanishing covariance constraints given in the statement of Lemma 3.5. Using the general result of Theorem 3.1, we have the following result for the dynamic Dorfman-type group testing algorithm by following similar steps to those in Theorem 3.2.

Theorem 3.3 *When the weak dynamic Dorfman-type group testing algorithm is used and* $(1 - \frac{T}{2n})(1 + nq(1-p)) < 1$*, we have*

$$\lim_{t\to\infty} E[\alpha_{weak}(t)] \approx n(1-p)(1-q)^{\frac{np}{1-(1-\frac{T}{2n})(1+nq(1-p))}} \tag{3.45}$$

which is a lower bound for $\lim_{t\to\infty} E[\alpha_{orig}(t)]$*, i.e., the expected number of susceptible individuals for the dynamic Dorfman-type group testing algorithm.*

Note that this result of the weak dynamic Dorfman-type group testing algorithm is a loose lower bound for the performance of the algorithm, which is only

significant because it shows that the weak dynamic Dorfman-type group testing algorithm performs in a similar manner with the weak dynamic individual testing algorithm, order-wise (T replaced with $T/2$), which is a performance lower bound for the dynamic Dorfman-type group testing algorithm.

3.3.3 Comparison of Dynamic Individual and Dorfman-Type Algorithms

To compare the average number of detected infections at a given time instant for the dynamic individual testing and dynamic Dorfman-type group testing algorithms, we obtain the following results stated in the following lemmas.

Lemma 3.8 *When there are $\tilde{\alpha}(t)$ susceptible and $\tilde{\lambda}(t)$ non-isolated infected individuals in a system after the infection spread phase, and just before the testing phase at time instant t, and the dynamic individual testing algorithm is being used, on average, $\frac{T\tilde{\lambda}(t)}{\tilde{\alpha}(t)+\tilde{\lambda}(t)}$ infections are detected and isolated at time t.*

Proof: When T individuals from $\tilde{\alpha}(t) + \tilde{\lambda}(t)$ individuals are uniformly randomly selected, we have

$$E\left[\sum_{i=1}^{T} \mathbb{1}_{\tilde{U}_i(t)=1}\right] = TP(\tilde{U}_i(t) = 1) \tag{3.46}$$

$$= \frac{T\tilde{\lambda}(t)}{\tilde{\alpha}(t) + \tilde{\lambda}(t)} \tag{3.47}$$

where $\tilde{U}_i(t)$ represents the infection status of the ith selected individual for testing at the time of the testing phase. ■

On the other hand, when the dynamic Dorfman-type group testing algorithm is used, $T/2$ individuals to be individually tested are chosen from a set of individuals of size $\frac{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))}{T}$ that is guaranteed to have at least one infected individual, in the case of $\tilde{\alpha}(t)+\tilde{\lambda}(t) \geq T^2/4$. When $\tilde{\alpha}(t)+\tilde{\lambda}(t) < T^2/4$, $T/2$ individuals to be tested individually are chosen from multiple groups, each having at least one infected individual. The following lemma gives an average number of detected and isolated infections at each time instant for the case of $\tilde{\alpha}(t)+\tilde{\lambda}(t) \geq T^2/4$, which, in general, holds for practical applications with low testing capacity. Moreover, the following result is also a lower bound for the case of $\tilde{\alpha}(t)+\tilde{\lambda}(t) < T^2/4$, where $T/2$ individuals to be individually tested are selected from multiple positive groups.

Lemma 3.9 *When there are $\tilde{\alpha}(t)$ susceptible and $\tilde{\lambda}(t)$ non-isolated infected individuals in a system after the infection spread phase and just before the testing phase at time instant t, with $\tilde{\alpha}(t)+\tilde{\lambda}(t) \geq T^2/4$, and the dynamic Dorfman-type group testing algorithm is being used, if $\tilde{\alpha}(t) \geq 2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T$, on average,*

$$\frac{T\tilde{\lambda}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))}\left(1-\frac{\binom{\tilde{\alpha}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T}}{\binom{\tilde{\alpha}(t)+\tilde{\lambda}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T}}\right)^{-1} \tag{3.48}$$

infections are detected and isolated at time t. If $\tilde{\alpha}(t) < 2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T$, then, on average, $\frac{T\tilde{\lambda}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))}$ infections are detected and isolated at time t. In the case of $\tilde{\alpha}(t)+\tilde{\lambda}(t) < T^2/4$, (3.48) is a lower bound for the average number of detected and isolated individuals at time t.

Proof: When $T/2$ individuals are uniformly randomly selected from a set of indi-

viduals that are guaranteed to have at least one infection, with size $2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T$, we have

$$E\left[\sum_{i=1}^{T/2}\mathbb{1}_{\tilde{U}_i(t)=1}\middle|\sum_{i=1}^{C}\mathbb{1}_{\tilde{U}_i(t)=1}\geq 1\right]=\frac{E\left[\sum_{i=1}^{T/2}\mathbb{1}_{\tilde{U}_i(t)=1}\right]}{P\left(\sum_{i=1}^{C}\mathbb{1}_{\tilde{U}_i(t)=1}\geq 1\right)} \tag{3.49}$$

$$=\frac{T\tilde{\lambda}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))}\left(1-\frac{\binom{\tilde{\alpha}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T}}{\binom{\tilde{\alpha}(t)+\tilde{\lambda}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T}}\right)^{-1} \tag{3.50}$$

where $\tilde{U}_i(t)$ represents the infection status of the ith selected individual for testing at the time of the testing phase and $C=2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T$.

In the case of $\tilde{\alpha}(t)+\tilde{\lambda}(t)<T^2/4$, $T/2$ individuals to be tested individually are chosen from multiple groups where each of them is guaranteed to have at least one infected individual. Therefore, the term in the denominator of the right-hand side of (3.49), i.e., $P\left(\sum_{i=1}^{C}\mathbb{1}_{\tilde{U}_i(t)=1}\geq 1\right)$, is replaced by the probability of the event that multiple subsets of size C having at least one non-isolated infected member, which is a subset of the event that only one subset of individuals of size C having at least one non-isolated infected member and thus, having lower probability. Therefore, (3.49) is also a lower bound for the average number of detected and isolated infections at time instant t, for the case of $\tilde{\alpha}(t)+\tilde{\lambda}(t)<T^2/4$. ■

For a given state of the system at the time of the testing phase, i.e., $\tilde{\alpha}(t)$ and $\tilde{\lambda}(t)$, as we show in Lemmas 3.8 and 3.9, using the dynamic Dorfman-type group testing algorithm becomes advantageous with respect to the dynamic individual

testing algorithm when $\tilde{\alpha}(t) \geq 2(\tilde{\alpha}(t) + \tilde{\lambda}(t))/T$ and

$$1/2 < \frac{\binom{\tilde{\alpha}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T}}{\binom{\tilde{\alpha}(t)+\tilde{\lambda}(t)}{2(\tilde{\alpha}(t)+\tilde{\lambda}(t))/T}} \tag{3.51}$$

$$= \frac{\prod_{i=0}^{C}(\tilde{\alpha}(t) - i)}{\prod_{i=0}^{C}(\tilde{\alpha}(t) + \tilde{\lambda}(t) - i)} \tag{3.52}$$

where $C = 2(\tilde{\alpha}(t) + \tilde{\lambda}(t))/T$.

In the next section, we present the numerical results of our two proposed dynamic algorithms, as well as their weak versions, under various sets of system parameters.

3.4 Numerical Results

In our numerical results, we implement the algorithms that we proposed: the dynamic Dorfman-type group testing algorithm, the dynamic individual testing algorithm, and the weak versions of these algorithms. In all of our simulations, we start with n individuals, with all of them susceptible. Then, at time $t = 0$, we realize the initial infections in the system uniformly randomly with probability p. At each time instant that follows, for the infection spread phase, we simulate the random infection spread from the non-isolated infections to the susceptible individuals. For the testing phase, we simulate the random selection of individuals to be tested and perform the tests. Depending on the test results, we simulate the isolation of the detected infections. We repeat these phases at each time instant until time $t = 500$

and obtain the sample paths of the random processes $\alpha(t)$, $\lambda(t)$, and $\gamma(t)$. We iterate this whole process 1000 times to obtain 1000 sample paths of the random processes, and then we calculate the average of the sample paths to obtain the expected values of $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$, numerically. In Figures 3.1–3.3, we plot these expected values of $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$ for the algorithms that we propose. In our simulations, we also consider the value of the theoretical approximation result that we obtained in Theorem 3.1. For each sample path, at each time instant, we numerically calculate the values of $p'(t)$ for both dynamic individual testing and dynamic Dorfman-type group testing algorithms and then use the expression that we obtained in Theorem 3.1 to calculate the $\alpha(t)$ approximation curve. We calculate and plot the average of the $\alpha(t)$ approximation curve. For the weak versions of the proposed algorithms, we use the results of Theorems 3.2 and 3.3 to directly calculate and plot the steady state approximations of $\alpha(t)$. Pseudo-code of the simulations that we perform is given below in Algorithm 1.

Algorithm 1 Simulations to Obtain Numerical Results

$i \leftarrow 1$ ▷ Simulations are repeated to obtain average sample paths.

while $i \leq i_{max}$ **do**

$t \leftarrow 0$

$U \leftarrow 0_{nx1}$

for individual j in $[n]$ **do**

if random roll in $[0, 1] \leq p$ **then**

$U_j \leftarrow 1$

$t \leftarrow 1$ ▷ Initial infections are realized

while $t \leq t_{max}$ **do**

for individual j in infections **do**

for individual k in susceptibles **do**

if random roll in $[0, 1] \leq q$ **then**

$U_k \leftarrow 1$

▷ End of the infection spread phase

for test τ in $[T]$ **do**

$y_\tau \leftarrow \bigvee_{j \in [n]} \boldsymbol{X}_{\tau j} \mathbb{1}_{\{U_j = 1\}}$

▷ Testing is done

for individual j in [n] **do**

if j has infection and detected **then**

$U_j \leftarrow 2$

▷ Recovery is done

$t \leftarrow t + 1$ ▷ $\alpha(t), \lambda(t), \gamma(t)$ are saved

▷ Averages of $\alpha(t), \lambda(t), \gamma(t)$ are calculated over iterations i

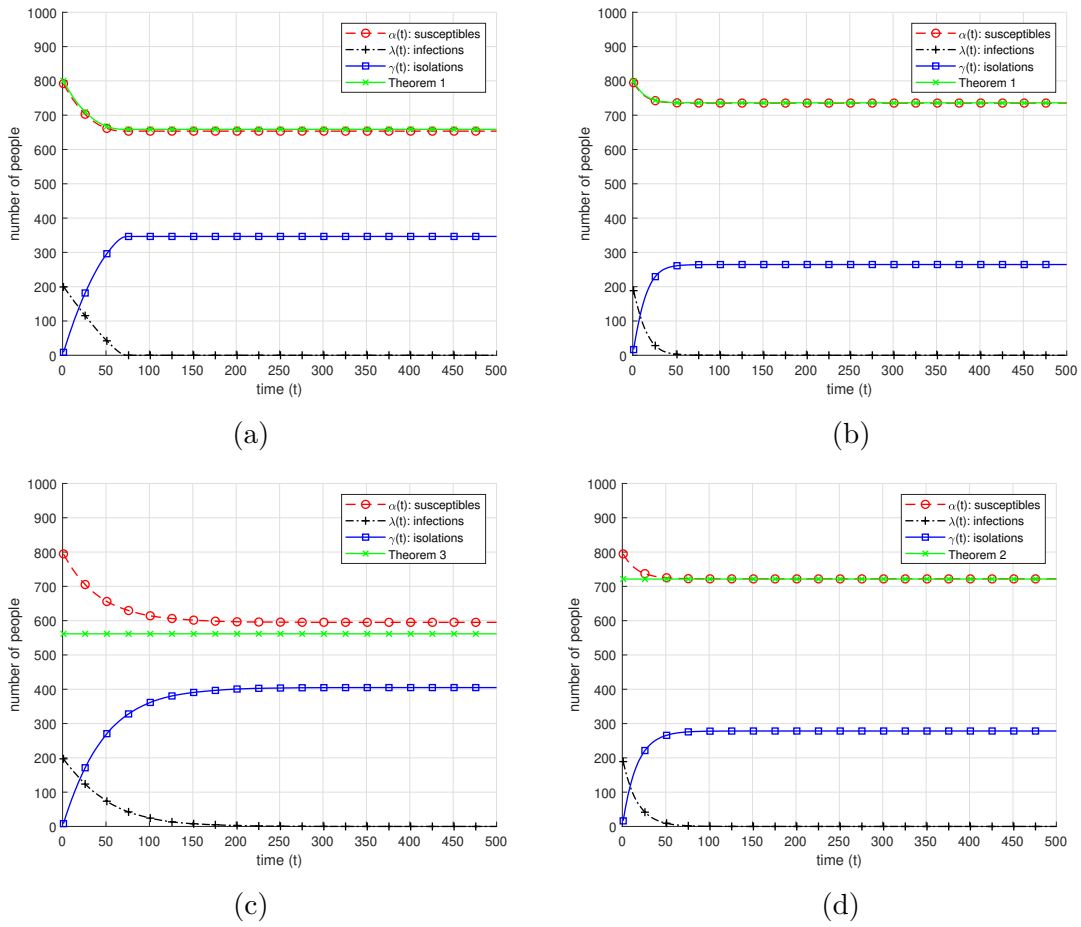


Figure 3.1: Average values of the random processes $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$, with obtained theoretical approximations given in Theorems 3.1–3.3 when $n = 1000$, $T = 80$, $q = 0.00003$, $p = 0.2$, for (**a**) dynamic Dorfman-type group testing algorithm, (**b**) dynamic individual testing algorithm, (**c**) weak dynamic Dorfman-type group testing algorithm, (**d**) weak dynamic individual testing algorithm.

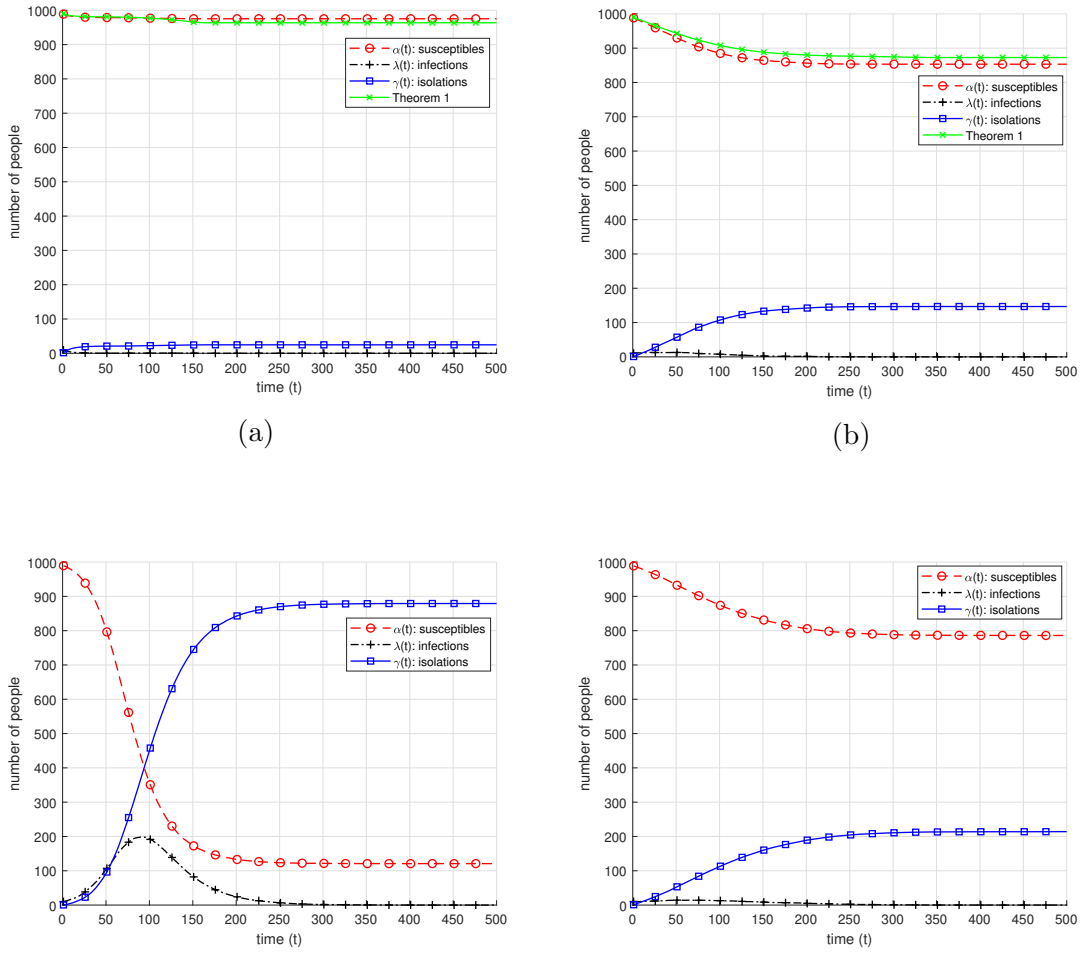


Figure 3.2: Average values of the random processes $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$, with obtained theoretical approximations given in Theorem 3.1 when $n = 1000$, $T = 80$, $q = 0.0001$, $p = 0.01$, for (**a**) dynamic Dorfman-type group testing algorithm, (**b**) dynamic individual testing algorithm, (**c**) weak dynamic Dorfman-type group testing algorithm, (**d**) weak dynamic individual testing algorithm.

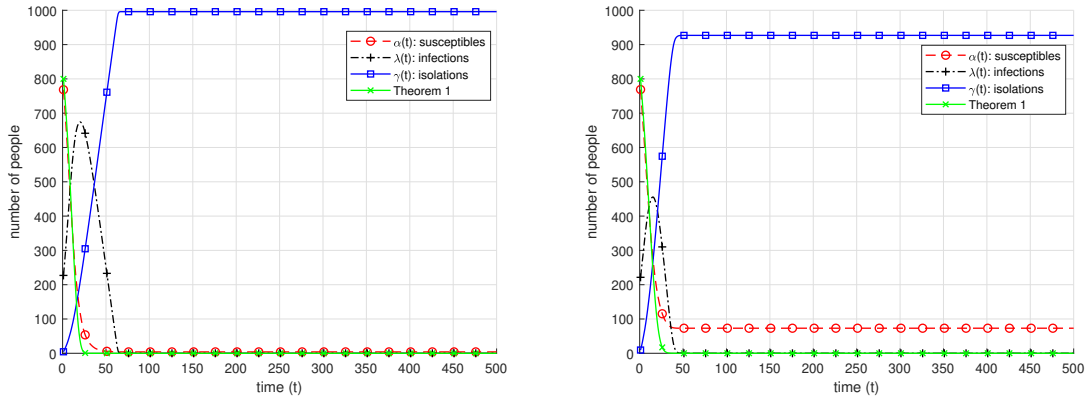


Figure 3.3: Average values of the random processes $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$, with obtained theoretical approximations given in Theorem 3.1 when $n = 1000$, $T = 40$, $q = 0.0002$, $p = 0.2$, for (**a**) dynamic Dorfman-type group testing algorithm, (**b**) dynamic individual testing algorithm.

In Figure 3.1, we present numerical results for the system with the parameters $n = 1000$, $T = 80$, $q = 0.00003$, and $p = 0.2$. Due to the relatively high number of initial infections in the system, we observe that the dynamic individual testing algorithm performs better than the dynamic Dorfman-type group testing algorithm in terms of the average steady-state $\alpha(t)$. In the weak versions of the algorithms, we observe that their performance is strictly worse than their respective original algorithms, at each time instant, in terms of the average $\alpha(t)$, as expected. The difference between the average $\alpha(t)$ curves of the original and weak versions of the dynamic Dorfman-type group testing algorithm is higher than the difference between the average $\alpha(t)$ curves of the original and weak versions of the dynamic individual testing algorithm. This is due to the fact that in the weak dynamic individual testing algorithm, we still utilize T tests at each time instant but can sample the isolated individuals to test, while in the weak dynamic Dorfman-type

group testing algorithm, we ignore the group tests and only consider $T/2$ individual tests. However, since the advantage of the group test is not effective for this set of parameters, as we present in Figure 3.1, even the weak dynamic Dorfman-type group testing algorithm provides a reasonable lower bound for its original version. Finally, we observe that our approximation results in Theorem 3.1 match with the average $\alpha(t)$ curves in both dynamic Dorfman-type group testing and dynamic individual testing algorithms. Similarly, the average $\alpha(t)$ curves that we obtain from the weak versions of the proposed algorithms are also closely approximated by the results that we obtain in Theorems 3.2 and 3.3.

In Figure 3.2, we run the same simulations as in Figure 3.1, for the parameters $n = 1000$, $T = 80$, $q = 0.0001$ and $p = 0.01$. Now, relative to the first set of parameters, the number of initial infections is lower but the infection spread probability is higher. Because of the targeted individual testing to the positive groups in the dynamic Dorfman-type group testing algorithm, it outperforms the dynamic individual testing algorithm for this set of parameters, as we present in Figure 3.2. Since the advantage of the group testing is more prevalent for this set of parameters, the weak version of the dynamic Dorfman-type group testing algorithm results in an average $\alpha(t)$ curve that is a loose lower bound for the average $\alpha(t)$ curve of the original version, while the weak dynamic individual testing algorithm results in a proper lower bound for the original version. Furthermore, for this set of parameters, despite the fact that the Theorem 3.1 approximation matches the average $\alpha(t)$ curves for both of the original versions of the proposed algorithms, the resulting Theorem 3.2 and Theorem 3.3 approximations cannot be used due to the non-convergent exponents

in the expressions.

In our third and final set of parameters, we consider a lower number of test capacity, T, than the first two sets of parameters, a high number of initial infections, p, and a high infection spread probability, q. As expected, for this set of parameters, for both of the algorithms, the system reaches a steady state when almost everyone in the population gets infected. Due to the high number of infections at each time instant in the system, the dynamic individual testing algorithm performs slightly better than the dynamic Dorfman-type group testing algorithm, even though it still fails to control the infection spread in an effective manner.

3.5 Conclusions

In this chapter, we considered a dynamic infection spread model over discrete time, inspired by the SIR model, widely used in the modeling of contagious infections in populations. Instead of recovered individuals in the system, we considered isolated infections, where infected individuals can be identified and isolated via testing. In our system model, the infection statuses of the individuals are random processes rather than random variables, such as the infection status of the individuals in the classical group testing problems. In parallel with the dynamic configuration of our system, we considered dynamic group testing algorithms: At each time instant, after the infection is spread by infected individuals to the susceptible individuals randomly, a given limited number of (possibly group) tests are performed to identify and isolate infected individuals. This dynamic infection spread and identi-

fication system is more challenging than the classical group testing problem setup since negative identifications are not finalized and can change over time, while only the positive identifications are isolated for the rest of the process. We analyzed the performance of dynamic testing algorithms by providing approximation results for the expected number of susceptible individuals (that have never gotten infected) when the infection is brought under control, where all infections are identified and isolated for symmetric and converging algorithms. Then, we proposed two dynamic algorithms: *dynamic individual testing algorithm* and *dynamic Dorfman-type group testing algorithm*. We considered the weak versions of these algorithms and used our general result to provide lower bounds on the expected number of susceptible individuals for these two algorithms. We compared the average identification performance of these two algorithms by deriving conditions when one algorithm outperforms the other. In our simulations, we implemented both the original and weak versions of the proposed algorithms and also simulated and compared the theoretical approximation results that we derived for three different sets of parameters, and we demonstrated various possible scenarios. Our work is unique in that the disease spread in our dynamic system is due to limited testing capacity as opposed to the delay in obtaining (unlimited) test results in the existing literature.

3.6 Appendix

Lemma 3.5 *When a symmetric and converging dynamic testing algorithm is implemented, and* $cov\left(P(U_i(t)=0|\lambda(t)), p'_{\lambda(t)}(t+1)\right)$ *and* $cov\left(P(U_i(t)=1|\lambda(t)), p'_{\lambda(t)}(t+1)\right)$

are arbitrarily small for all $t \geq 0$, we have

$$P(U_i(t) = 1) \approx p((1 + nq(1 - p)))^t \prod_{j=1}^{t} p'(j) \tag{3.53}$$

where the conditional probability of an individual not being identified in the tests at time t given $\lambda(t-1)$ is denoted by $p'_{\lambda(t-1)}$.

Proof: Conditioned on $\lambda(t-1)$, we have the following recursive relation

$$\begin{aligned}
&P(U_i(t) = 1) \\
&\quad = E[P(U_i(t) = 1|\lambda(t-1))] && (3.54)\\
&\quad = E[P(U_i(t-1) = 0|\lambda(t-1))(1 - (1-q)^{\lambda(t-1)})p'_{\lambda(t-1)}(t) \\
&\qquad + P(U_i(t-1) = 1|\lambda(t-1))p'_{\lambda(t-1)}(t)] && (3.55)\\
&\quad \approx E[1 - (1-q)^{\lambda(t-1)}]E[P(U_i(t-1) = 0|\lambda(t-1))p'_{\lambda(t-1)}(t)] \\
&\qquad + E[P(U_i(t-1) = 1|\lambda(t-1))p'_{\lambda(t-1)}(t)] && (3.56)\\
&\quad \approx E[p'_{\lambda(t-1)}(t)]E[1 - (1-q)^{\lambda(t-1)}]E[P(U_i(t-1) = 0|\lambda(t-1))] \\
&\qquad + E[p'_{\lambda(t-1)}(t)]E[P(U_i(t-1) = 1|\lambda(t-1))] && (3.57)\\
&\quad = p'(t)(P(U_i(t-1) = 0)(1 - E[(1-q)^{\lambda(t-1)}]) + P(U_i(t-1) = 1)) && (3.58)\\
&\quad \approx p'(t)((1-p)(1-q)^{n\sum_{j=0}^{t-2} P(U_i(j)=1)}(1 - (1-q)^{E[\lambda(t-1)]}) + P(U_i(t-1) = 1)) \\
& && (3.59)\\
&\quad = p'(t)((1-p)(1-q)^{n\sum_{j=0}^{t-2} P(U_i(j)=1)}(1 - (1-q)^{nP(U_i(t-1)=1)}) + P(U_i(t-1) = 1)) \\
& && (3.60)
\end{aligned}$$

$$=p'(t)((1-p)((1-q)^{n\sum_{j=0}^{t-2}P(U_i(j)=1)}-(1-q)^{n\sum_{j=0}^{t-1}P(U_i(j)=1)})+P(U_i(t-1)=1)) \tag{3.61}$$

$$\approx p'(t)((1-p)(qn\sum_{j=0}^{t-1}P(U_i(j)=1)-qn\sum_{j=0}^{t-2}P(U_i(j)=1))+P(U_i(t-1)=1)) \tag{3.62}$$

$$=p'(t)\left(nq(1-p)P(U_i(t-1)=1)+P(U_i(t-1)=1)\right) \tag{3.63}$$

$$=p'(t)\left(1+nq(1-p)\right)P(U_i(t-1)=1) \tag{3.64}$$

where (3.56) follows from the arbitrarily small variance of $(1-q)^{\lambda(t)}$ similar to the proof of Lemma 3.3, (3.57) follows from the given vanishing covariance assumptions in the statement of the lemma, (3.59) follows from Lemma 3.4, and (3.62) follows from the linear approximation $(1-q)^x \approx 1-qx$. Recursively applying (3.64) yields the desired result. ■

CHAPTER 4

Dynamic SAFFRON: Disease Control Over Time Via Group Testing

4.1 Introduction

In this chapter, we consider the discrete time, SIR-based dynamic infection spread model introduced in Chapter 3. In our model, at each time instant, the testing capacity is limited and fixed, and full identification of the infections in the system is not possible. Each discrete time instant consists of two phases: the infection spread phase and the testing phase. Identified infections are isolated, and further infection spread by them is prevented. Since the testing capacity is limited at each time instant, not all of the infections are detected, and infected individuals that are not detected and isolated keep spreading the infection. The dynamic infection spread model is based on the SIR model, where the population is divided into three disjoint groups: susceptible individuals (S), non-isolated infections (I), and isolated infections (R). At the initial time instant, we assume i.i.d. initial infections in the system. At each time instant that follows, non-isolated infections spread the disease to the susceptible individuals in the system. Consecutively, during the testing phase,

a limited number of tests are performed and detected infections are isolated. We assume that, for the rest of the testing process, isolated infections remain in the same state. This is a reasonable assumption since they eventually recover, and even when their isolation ends, they can be immune to the infection for a disease-specific time period. We assume that the testing process will be finalized before the end of that time period. The objective is not to identify all of the infections while minimizing the number of tests but to identify as many infections as possible at each time instant by using the limited number of T tests and to eventually control the spread of the disease.

We propose two novel performance metrics: disease control time, $\bar{t}$, and ϵ-disease control time $\bar{t}_\epsilon$. We characterize the performance of dynamic individual testing algorithms in terms of these novel performance metrics. Moreover, we introduce a novel dynamic algorithm: *dynamic SAFFRON based group testing algorithm*, which is inspired by the static SAFFRON scheme that is studied in [106] and introduced in [48]. We present the average case theoretical analysis of both the dynamic individual testing algorithm and dynamic SAFFRON-based group testing algorithm, in (4.6) and (4.12), respectively. Finally, we implement the discrete-time SIR-based dynamic infection spread model and the dynamic algorithms to simulate them and compare the numerical results with the theoretical results we obtain.

4.2 System Model

There are n individuals in the system, and their infection status, which we denote by $U_i(t)$ for individual i at time t, changes over discrete time instants. At any given time, there are three disjoint groups in the population: susceptible individuals who can get infected (S), non-isolated infections who have not been detected (I), and isolated infections who have been detected by performed tests and isolated indefinitely (R). $U_i(t)$ can take values 0 (susceptible), 1 (non-isolated infection), and 2 (isolated infection), representing the infection state of the individual i at time t. At $t = 0$, we introduce the initial i.i.d. infections to the system: each individual is infected with probability p independently. We introduce three random processes: $\alpha(t)$ denotes the number of susceptible individuals, $\lambda(t)$ denotes the number of non-isolated infections, and $\gamma(t)$ denotes the isolated infections, at time t.

Each discrete time instant, $t \geq 1$ consists of two consecutive phases: the infection spread phase and the testing phase. During the infection spread phase, each non-isolated and infected individual can infect each susceptible individual with probability q independently. Here, since each susceptible individual can be infected by each non-isolated and infected individual, this can be modeled as the realization of $\alpha(t)\lambda(t)$ independent Bernoulli random variables with parameter q, for each time $t \geq 1$. Following the infection spread phase, the testing phase starts in which the testing capacity is limited to a given, fixed number, T. Thus, the aim of designing the group testing algorithm is to construct $T \times n$ binary test matrices that specify the testing pools. Let $\boldsymbol{X}(t)$ denote the binary test matrix for the tests at time t,

and $\boldsymbol{X}_{ij}(t)$ denote the ith row and jth column of $\boldsymbol{X}(t)$, where $\boldsymbol{X}_{ij}(t)$ is 1 if the test sample of jth individual is mixed in the ith mixed test sample, and 0 otherwise. Let $y(t)$ denote the test result vector of the tests performed at time t and $y_i(t)$ denote the ith test result at time t. Then, we have

$$y_i(t) = \bigvee_{j\in[n]} \boldsymbol{X}_{ij}(t)\mathbb{1}_{\{U_j(t)=1\}}, \quad i \in [T] \tag{4.1}$$

We assume that results of the tests that are performed at time t are available before the infection spread phase at time $t+1$, i.e., $y(t)$ vector is known while designing the binary test matrix $\boldsymbol{X}(t+1)$. Moreover, when infected individuals are detected during the testing phase at time t, they are immediately isolated and, thus, prevented from spreading the infection for the rest of the testing process, i.e., $U_i(t) = 2$ for all times $t \geq t'$, if the individual i is detected to be infected by the tests performed at time t'. Note that the only possible infection state changes in our system are either from susceptible individuals to non-isolated infections, which happens via infection spread from non-isolated and infected individuals to susceptible individuals during the infection spread phase at each time, or from non-isolated infections to isolated infections which happens via infection detection during the testing phase at each time instant. A group testing policy π is a scheme that specifies the algorithm to construct binary test matrices $\boldsymbol{X}(t)$ for each time instant $t \geq 1$ until the infection is under control.

Since the source of the infection state evolution in our dynamic model is the non-isolated infections, we define the disease control time $\bar{t}$ to be the first time when

no non-isolated infections remain in the population, i.e., $\bar{t} = \min\{t \mid \lambda(t) = 0\}$. After time $\bar{t}$, every individual in the population is either susceptible or isolated, i.e., $U_i(t) = 0$ or $U_i(t) = 2$ for $i \in [n]$ and for all $t \geq \bar{t}$. Furthermore, we introduce ϵ-disease control time for probabilistic analysis, denoted by $\bar{t}_\epsilon$ where $E[\lambda(\bar{t}_\epsilon)] = \epsilon$ holds. While characterizing and analyzing the performance of a group testing algorithm π, there are two performance metrics associated with it: the disease control time $\bar{t}$ (or $\bar{t}_\epsilon$) and the number of individuals that have never gotten infected throughout the process, i.e., $\alpha(\bar{t})$. Lower $\bar{t}$ is favored to control the disease faster, and higher $\alpha(\bar{t})$ is favored to control the disease with a lower number of total infections.

4.3 Proposed Algorithms and Analysis

In this section, we introduce a novel dynamic group testing algorithm: dynamic SAFFRON-based group testing algorithm. This algorithm is inspired by the static group testing algorithm introduced in [48], coined as the SAFFRON scheme, which is further studied in [106] for the partial detection problem. We analyze the dynamic SAFFRON-based group testing algorithm in terms of two performance metrics: disease control time and the number of susceptible individuals when the disease is brought under control. Furthermore, we analyze the dynamic individual testing algorithm we introduced in Chapter 3 in terms of its ϵ-disease control time performance.

4.3.1 Related Prior Results

For completeness, here we present a revised version of the results from Chapter 3, which we use for the analysis in this chapter. To keep it compact, we only include the main results that we use in the analysis in this chapter.

We consider *symmetric and converging dynamic testing algorithms* which must satisfy the *symmetry* criterion:

$$P(U_i(t) = k) = P(U_j(t) = k), \quad k \in \{0, 1, 2\} \tag{4.2}$$

for each time instant $t \geq 0$ and for all $i, j \in [n]$ pair. Moreover, they must satisfy the *convergence* criterion:

$$\lim_{t \to \infty} P(U_i(t) = 1) = o(1/n), \quad i \in [n] \tag{4.3}$$

Let $p'(t)$ denote the probability of an individual not being identified during the testing phase at time t. Note that since we consider symmetric and converging dynamic testing algorithms, this probability is the same for all individuals. We consider dynamic testing algorithms where $p'(t)$ only depends on the testing capacity T, $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$. We assume the infection spread probability q scales as $o(1/n)$. Since the infection spread is realized for every susceptible and non-isolated infected individual pair and there can be $O(n^2)$ such pairs at each time instant, assuming q to be $o(1/n)$ is a mild assumption.

In the remainder of this subsection, we present our prior result from Chapter 3 that we use in our analysis in this chapter regarding symmetric and converging dynamic testing algorithms.

We start with the following lemma, where we prove that symmetric and converging dynamic testing algorithms guarantee the disease to be controlled eventually.

Lemma 4.1 *When a symmetric and converging dynamic testing algorithm is implemented,* $\lim_{t\to\infty} E[\lambda(t)] = o(1)$ *and thus, the system approaches the steady state, in the average-case.*

The statements of Lemma 4.2 and Theorem 4.1 give improved characterizations of the approximation terms compared to the original statements in Chapter 3. The approximations are characterized by $o(1)$ terms in this chapter, which follow from the linear approximations in Taylor series expansions for exponential terms and hold since $q = o(1/n)$. Moreover, arbitrarily small variance conditions in Chapter 3 are also characterized to scale as $o(1)$ here. In the following lemma, we characterize an approximation of the probability of an individual being infected and non-isolated at time t, i.e., the probability of the event of $U_i(t) = 1$.

Lemma 4.2 *When a symmetric and converging dynamic testing algorithm is implemented, we have*

$$P(U_i(t) = 1) = p((1 + nq(1 - p)))^t \prod_{j=1}^{t} p'(j) + o(1) \tag{4.4}$$

where the conditional probability of an individual not being identified in the tests at

time t *given* $\lambda(t-1)$ *is denoted by* $p'_{\lambda(t-1)}$ *and when cov* $\left(P(U_i(t)=0|\lambda(t)), p'_{\lambda(t)}(t+1)\right)$ *and cov* $\left(P(U_i(t)=1|\lambda(t)), p'_{\lambda(t)}(t+1)\right)$ *both scale as* $o(1)$ *with respect to* n *for all* $t \geq 0$.

The following theorem presents a general result that characterizes the approximation of the expectation of $\alpha(t)$. This result holds for all symmetric and converging dynamic testing algorithms that satisfy the small covariance conditions that we state in Lemma 4.2.

Theorem 4.1 *When a symmetric and converging dynamic testing algorithm is implemented and vanishing covariance constraints in Lemma 4.2 are satisfied for all* $t \geq 0$, *we have*

$$E[\alpha(t)] = n(1-p)(1-q)^{np\sum_{i=0}^{t-1}\left((1+nq(1-p))^i \prod_{j=1}^{i} p'(j)\right)} + o(1) \tag{4.5}$$

We use Theorem 4.1 to characterize the expected number of susceptible individuals when the disease is brought under control, which is one of the two performance metrics we consider. On the other hand, for the characterization of the disease control time $\bar{t}$, our analysis is based on Lemma 4.2.

4.3.2 Dynamic Individual Testing Algorithm

In the dynamic individual testing algorithm, we consider randomized, individual testing of T individuals at each time instant $t \geq 1$, where T individuals to be tested are uniformly randomly selected from the whole population, independent across both

individuals and time. In Chapter 3, we proved that the dynamic individual testing algorithm (weak dynamic individual testing algorithm in Chapter 3) is a symmetric and converging dynamic testing algorithm. Furthermore, we derived $E[\alpha(t)]$ results for the disease control time. Here, we derive ϵ-disease control time performance results of the dynamic individual testing algorithm.

In the following theorem, we present our result for the ϵ-disease control time performance metric when the dynamic individual testing algorithm is used in our proposed dynamic system.

Theorem 4.2 *When the dynamic individual testing algorithm is used, we have*

$$\bar{t}_\epsilon = \frac{\ln(\epsilon/np + o(1))}{\ln\left((1-\frac{T}{n})(1+nq(1-p))\right)} \tag{4.6}$$

Proof: We start with the mean of non-isolated infections

$$E[\lambda(t)] = nP(U_i(t)=1) \tag{4.7}$$

$$= np((1+nq(1-p)))^t \prod_{j=1}^{t} p'(j) + o(n) \tag{4.8}$$

where (4.7) follows from the definition of $\lambda(t)$ as the total number of non-isolated infected individuals and (4.8) follows from Lemma 4.2. Then, we have

$$\epsilon = np((1+nq(1-p)))^{\bar{t}_\epsilon} \prod_{j=1}^{\bar{t}_\epsilon} p'(j) + o(n) \tag{4.9}$$

$$= np\left((1-\frac{T}{n})(1+nq(1-p))\right)^{\bar{t}_\epsilon} + o(n) \tag{4.10}$$

where (4.9) follows from the definition of ϵ-disease control time and (4.10) follows from the fact that $p'(t) = (1 - T/n)$ for all $t \geq 1$ when dynamic individual testing algorithm is used. Finally, we get

$$\bar{t}_\epsilon = \frac{\ln(\epsilon/np + o(1))}{\ln\left((1 - \frac{T}{n})(1 + nq(1-p))\right)} \tag{4.11}$$

by arranging the terms. ■

In Theorem 4.2, we characterize the ϵ-disease control time metric for the dynamic individual testing algorithm. Since it is defined as the time instant when there are ϵ non-isolated infected individuals in the average case, the ϵ-disease control time metric presents a characterization of how fast a given algorithm can control the disease spread.

4.3.3 Dynamic SAFFRON Based Group Testing Algorithm

We propose and analyze a novel algorithm, which we coin as the dynamic SAFFRON-based group testing algorithm. It is inspired by the static group testing algorithm, SAFFRON, introduced in [48] and studied in [106] for partial detection.

The static SAFFRON scheme-inspired algorithm presented in [106] is based on constructing binary test matrices for the groups of individuals that contain approximately one infected individual. Constructed SAFFRON scheme test matrices can be used to recover exactly one infected individual within the tested group while also indicating whether there is zero or more than one infected individual within the tested group if there is not exactly one infected individual.

Here, we propose the novel dynamic SAFFRON-based group testing algorithm, where at each time instant $t+1$, groups of size $\lfloor (n-\gamma(t))/E[\lambda(t)] \rfloor$ are selected uniformly and randomly from the set of non-isolated individuals. For the $\lambda(t)$ values that are close to their mean,[1] each of these groups has approximately one infected and non-isolated individual.

By designing SAFFRON scheme-inspired binary test sub-matrices that are guaranteed to identify one infection from each group that has exactly one infection, we detect and isolate infections over time.

For a selected group of individuals, the binary test sub-matrix is constructed as follows:

- Let η denote the size of the selected individuals, i.e., $\eta = \lfloor (n-\gamma(t))/E[\lambda(t)] \rfloor$.
- First $\lceil \log(\eta) \rceil$ rows of the test sub-matrix are constructed where the column vector corresponding to column i is set to the binary representation of the number $i-1$.
- Then, the remaining $\lceil \log(\eta) \rceil$ rows of the test sub-matrix are constructed by concatenating the created sub-matrix in the previous step with the binary matrix where the value of each element is flipped, i.e., XORed with 1.

By using this construction, it is guaranteed that if there is exactly one infection within the tested individuals, there will be exactly $\lceil \log(\eta) \rceil$ positive tests, and the positive test indices will be the binary representation of $i-1$ where the infected

[1] Our numerical results suggest a concentration around the mean for $\lambda(t)$. An in-depth concentration analysis can be the subject of future works.

individual index is i. If there is no infection, then all tests will be negative. In the case of more than one infection, strictly more than $\lceil \log(\eta) \rceil$ tests will be positive. Thus, this construction guarantees the detection of a single infection within the selected group.

In the following lemma, we characterize the expected number of detected infections at each time instant.

Lemma 4.3 *When the dynamic SAFFRON-based group testing algorithm is used, at each time instant $t \geq 1$, in average*

$$\frac{T}{2} \log^{-1}\left(\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}\right)\left(1-\frac{E[\lambda(t-1)]}{n-\gamma(t-1)}\right)^{\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}-1} \tag{4.12}$$

infections are detected and isolated.

Proof: We choose the group size as $\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}$ since there will be approximately one infected individual within each group when the groups are chosen uniformly randomly from the set of all non-isolated individuals. Since the testing capacity is T at each time instant, there will be

$$\frac{T}{2} \log^{-1}\left(\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}\right) \tag{4.13}$$

such groups. Recall that the probability that a non-isolated individual is infected is $\frac{E[\lambda(t-1)]}{n-\gamma(t-1)}$ since the dynamic SAFFRON-based group testing algorithm uniformly randomly selects the groups at each time instant and since the infection statistics are symmetric over individuals. Furthermore, the group size is the inverse of this term.

Therefore, each group contains exactly one infected individual with probability

$$\left(1 - \frac{E[\lambda(t-1)]}{n - \gamma(t-1)}\right)^{\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}-1} \tag{4.14}$$

where the lemma statement follows. ■

Similar to our analysis for the dynamic individual testing algorithm, we can use the results of Lemma 4.2 and Theorem 4.1 for dynamic SAFFRON-based group testing algorithm where we have

$$p'(t) = \frac{\zeta}{E[\lambda(t)]} \tag{4.15}$$

where $\zeta = \frac{T}{2}\log^{-1}\left(\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}\right)\left(1 - \frac{E[\lambda(t-1)]}{n-\gamma(t-1)}\right)^{\frac{n-\gamma(t-1)}{E[\lambda(t-1)]}-1}$. To justify using Lemma 4.2 and Theorem 4.1 results, we need to prove that the dynamic SAFFRON-based group testing algorithm satisfies symmetry and convergence criteria in (4.2) and (4.3).

Since the selection of tested groups is independent across individuals for each time instant, the dynamic SAFFRON-based group testing algorithm is symmetric. To guarantee convergence, we consider using the dynamic individual testing algorithm whenever $T < 2\log\lfloor(n - \gamma(t))/E[\lambda(t)]\rfloor$, i.e., the regime where the expected number of non-isolated infections is small with respect to the total number of non-isolated individuals. In that regime, SAFFRON-based construction cannot be used efficiently to detect infections. In the regime where the expected number of non-isolated infections is high, the SAFFRON-based scheme can consistently detect infections as characterized in Lemma 4.3.

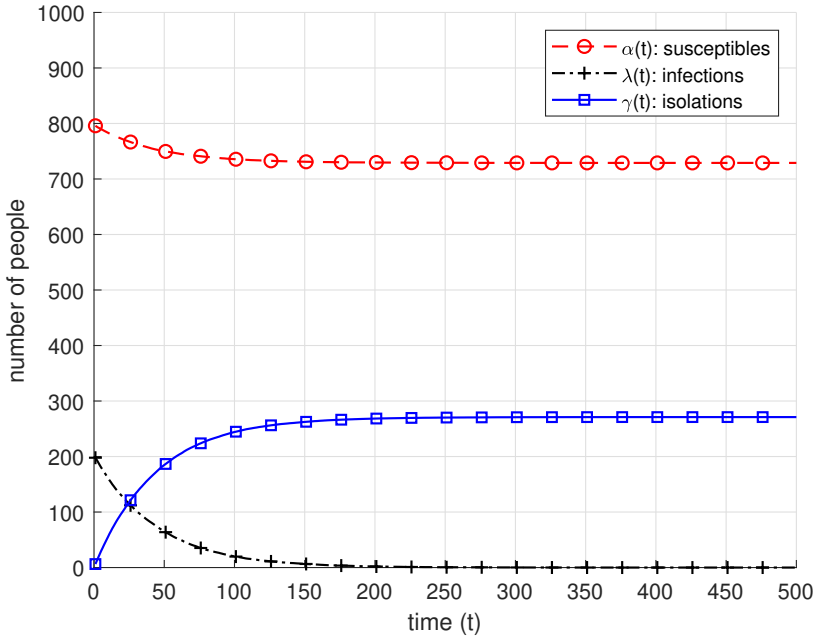


Figure 4.1: Numerical averages of the random processes $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$ for the system parameters $n = 1000$, $T = 30$, $q = 0.00001$, $p = 0.2$, for dynamic individual testing algorithm.

4.4 Numerical Results

In our simulations, we implement the dynamic discrete SIR-based infection spread system. We initialize our system with random independent initial infections with probability p. In the following time instants, we simulate the random disease spread from non-isolated infections to susceptible individuals, independently with probability q. For both dynamic algorithms, we run our system for 500 time instants and iterate, realizing this dynamic system 1000 times. We save the sample path trajectories of the random processes $\alpha(t)$, $\lambda(t)$, and $\gamma(t)$, take their averages, and plot.

To control disease spread, in our first simulation, we implement the dynamic individual testing algorithm, where randomly selected T individuals are individually

tested at each time instant. Here, by using our theoretical result in Theorem 4.2, we have $\bar{t}_\epsilon = 235.57$ when $\epsilon = 1$. This is consistent with the numerical results plotted in Figure 4.1. In our second simulation, we implement a hybrid version of the dynamic SAFFRON-based group testing algorithm. At each time instant, we randomly select groups of people where each group has one infection on average, where we use expected $\lambda(t)$ that we calculate by using Lemma 4.3, to obtain the group size. Then, we construct SAFFRON scheme-based test sub-matrices for each group and perform maximum T tests. In practice, SAFFRON scheme-based construction can have a non-utilized testing capacity at each discrete time instant since the testing capacity T may not be divisible by the calculated group size. We utilize this available testing capacity and perform random individual testing when available non-utilized tests remain. Moreover, later in the testing process, since the expected $\lambda(t)$ becomes smaller, the calculated group size grows larger, and the required number of tests for SAFFRON-based construction exceeds the testing capacity T. For these later times in the testing process, we switch to the dynamic individual testing algorithm to detect and isolate remaining infections. We also plot the theoretical calculation of expected $\lambda(t)$ in Figure 4.2, where we use Lemma 4.3 for calculation, which aligns with the numerical results that we obtain, as we plot in Figure 4.2. We observe that the steady state average of the number of susceptible individuals is slightly higher in the dynamic individual testing algorithm while the convergence time is slightly lower in the dynamic SAFFRON-based group testing algorithm. This observation for this set of parameters suggests that both algorithms can be used to optimize different performance metrics, depending on the system requirements and parameters, as well

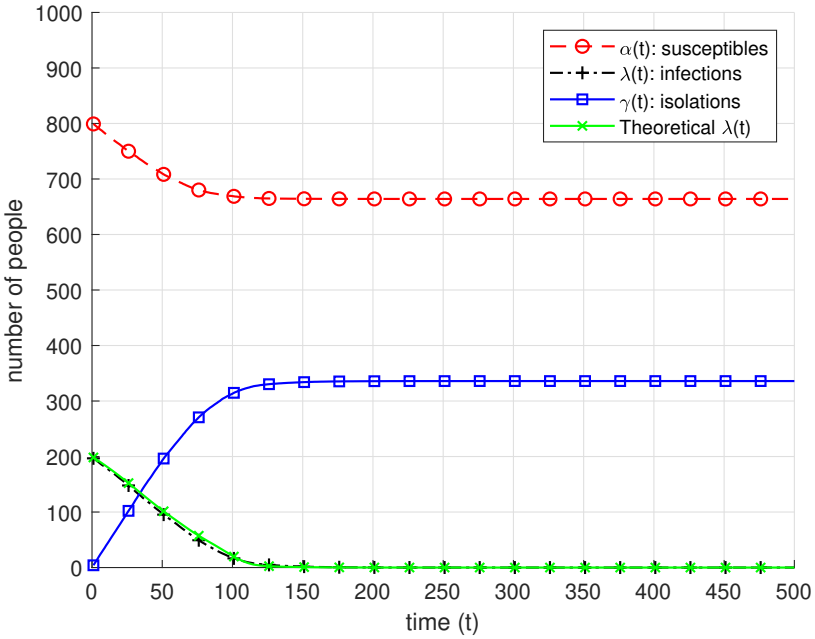


Figure 4.2: Numerical averages of the random processes $\alpha(t)$, $\lambda(t)$ and $\gamma(t)$, with theoretical calculation of $\lambda(t)$, for the system parameters $n = 1000$, $T = 30$, $q = 0.00001$, $p = 0.2$, for dynamic SAFFRON based group testing algorithm.

as hybrid usage of the dynamic algorithms is also possible as in our hybrid dynamic SAFFRON-based group testing algorithm implementation.

4.5 Conclusions

In this chapter, we considered a dynamic infection spread model based on the discrete SIR model. We aimed to efficiently utilize the available testing capacity T at each time instant to control disease spread within a community rather than minimize the number of required tests to determine every infection as in the static group testing problem. We presented two novel performance metrics: disease control time and ϵ-disease control time. Unlike our previous metric in Chapter 3, which is the expected number of susceptible individuals in the steady state, ϵ-disease con-

trol time measures the timeliness of the disease control. We refined our results in Chapter 3 and used them to characterize the performance of *dynamic individual testing algorithm* in terms of the ϵ-disease control time metric, in (4.6). We introduced *dynamic SAFFRON-based group testing algorithm* and presented average case performance results in (4.12), which can be further used in combination with our previous results regarding symmetric and converging dynamic testing algorithms. We simulated our dynamic system, implemented the dynamic individual testing algorithm and dynamic SAFFRON-based group testing algorithm, and presented the numerical results we obtained. The novel performance metrics that we introduced in this chapter add a novel dimension to our proposed dynamic model, where one can consider implementing and assessing the performance of a wider variety of algorithms depending on the system requirements.

CHAPTER 5

Conclusions

In this dissertation, we studied the applications of group testing in novel structured and dynamic networks.

In Chapter 2, we studied a random connection graph-based community structure. This results in a non-identical and correlated infection status distribution of the individuals. We proposed novel two-step sampled group testing algorithms, and we characterized their optimal parameters and their constructions. For exponentially split cluster formation trees, we explicitly calculated the expected number of false classifications and the required number of tests. We showed that by utilizing the community structure-based side information, even when the prevalence rate is high, group testing can be utilized to reduce the required number of tests significantly.

In Chapter 3, we proposed and analyzed a discrete-time SIR-based dynamically evolving network structured disease spread system model. We proposed two novel dynamic testing algorithms: the dynamic individual testing algorithm and the dynamic Dorfman-type group testing algorithm. We analyzed the performances of

these algorithms based on our average-case performance metric: the expected number of susceptible individuals when the disease spread is brought under control. We obtained general theoretical results for symmetric and converging dynamic group testing algorithms family. We characterized the conditions for when each algorithm outperforms the other one. We implemented the proposed discrete-time SIR-based disease spread network and proposed algorithms, and we ran simulations in varying parameter regimes.

In Chapter 4, we further expanded the dynamic disease spread network-based system model that we introduced in Chapter 3. We proposed two novel performance metrics (disease control time and ϵ-disease control time) motivated by the fact that one might have varying objectives while designing dynamic group testing algorithms to control disease spread within a population. We revised and expanded our results from Chapter 3, and also proposed the dynamic SAFFRON-based group testing algorithm. By using our general results, we characterized the performance of the proposed dynamic algorithms in terms of the ϵ-disease control time metric. We implemented the proposed algorithms and ran simulations to obtain numerical results for both the dynamic SAFFRON-based group testing algorithm and the dynamic individual testing algorithm.

The contents of Chapter 2 are published in [96, 107], Chapter 3 in [104, 108], Chapter 4 in [105]. Other publications of the author's Ph.D. research that are not included in this dissertation are [109–113].

Bibliography

[1] R. Dorfman. The detection of defective members of large populations. *Annals of Mathematical Statistics*, 14(4):436–440, December 1943.

[2] M. Aldridge, O. Johnson, and J. Scarlett. *Group Testing: An Information Theory Perspective*. Now Foundations and Trends, December 2019.

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

[4] H. M. Finucan. The blood testing problem. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 13(1):43–50, 1964.

[5] M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defectives in a binomial sample. *Bell System Technical Journal*, 38(5):1179–1252, September 1959.

[6] M. Sobel and P. A. Groll. Binomial group-testing with an unknown proportion of defectives. *Technometrics*, 8(4):631–656, 1966.

[7] 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, September 1972.

[8] M. C. Hu, F. K. Hwang, and J. K. Wang. A boundary problem for group testing. *SIAM Journal on Algebraic Discrete Methods*, 2(2):81–87, 1981.

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

[10] M. B. Malyutov. Search for sparse active inputs: A review. In *Information Theory, Combinatorics, and Search Theory*, pages 609–647, 2013.

[11] A. G. D'yachkov. Lectures on designing screening experiments, 2014. Available at arXiv:1401.7505.

[12] D.-Z. Du and F. K. Hwang. *Pooling Designs and Nonadaptive Group Testing: Important Tools for DNA Sequencing*. World Scientific, 1st edition, 2006.

[13] T. Wadayama. Nonadaptive group testing based on sparse pooling graphs. *IEEE Trans. on Info. Theory*, 63(3):1525–1534, 2017.

[14] M. Aldridge. Individual testing is optimal for nonadaptive group testing in the linear regime. *IEEE Trans. on Info. Theory*, 65(4):2058–2061, 2019.

[15] M. Mézard, M. Tarzia, and C. Toninelli. Group testing with random pools: Phase transitions and optimal strategy. *Journal of Statistical Physics*, 131(5):783–801, 2008.

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

[17] H. B. Chen and F. K. Hwang. Exploring the missing link among d-separable, d^--separable and d-disjunct matrices. *Discrete Applied Mathematics*, 155(5):662 – 664, March 2007.

[18] M. Ruszinko. On the upper bound of the size of the r-cover-free families. *Journal of Combinatorial Theory, Series A*, 66(2):302 – 310, May 1994.

[19] C. Shangguan and G. Ge. New bounds on the number of tests for disjunct matrices. *IEEE Trans. on Info. Theory*, 62(12):7518–7521, 2016.

[20] E. Porat and A. Rothschild. Explicit nonadaptive combinatorial group testing schemes. *IEEE Transactions on Information Theory*, 57(12):7982–7989, 2011.

[21] A. Mazumdar. Nonadaptive group testing with random set of defectives. *IEEE Trans. on Info. Theory*, 62(12):7522–7531, December 2016.

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

[23] P. Fischer, N. Klasner, and I. Wegenera. On the cut-off point for combinatorial group testing. *Discrete Applied Mathematics*, 91(1):83–92, 1999.

[24] M. Aldridge. Rates of adaptive group testing in the linear regime. In *IEEE ISIT*, July 2019.

[25] O. Johnson. Strong converses for group testing from finite blocklength results. *IEEE Transactions on Information Theory*, 63(9):5923–5933, 2017.

[26] A. Coja-Oghlan, O. Gebhard, M. Hahn-Klimroth, and P. Loick. Information-theoretic and algorithmic thresholds for group testing. *IEEE Trans. on Info. Theory*, pages 7911–7928, 2020.

[27] L. Baldassini, O. Johnson, and M. Aldridge. The capacity of adaptive group testing. In *IEEE ISIT*, July 2013.

[28] A. Allemann. An efficient algorithm for combinatorial group testing. In *Information Theory, Combinatorics, and Search Theory: In Memory of Rudolf Ahlswede*, pages 569–596, January 2013.

[29] T. Kealy, O. Johnson, and R. Piechocki. The capacity of non-identical adaptive group testing. In *Allerton Conference*, pages 101–108, 2014.

[30] L. Riccio and C. J. Colbourn. Sharper bounds in adaptive group testing. *Taiwanese Journal of Mathematics*, 4(4):669–673, December 2000.

[31] J. Wolf. Born again group testing: Multiaccess communications. *IEEE Trans. on Info. Theory*, 31(2):185–191, 1985.

[32] G. K. Atia and V. Saligrama. Boolean compressed sensing and noisy group testing. *IEEE Trans. on Info. Theory*, 58(3):1880–1901, March 2012.

[33] S. Cai, M. Jahangoshahi, M. Bakshi, and S. Jaggi. Efficient algorithms for noisy group testing. *IEEE Trans. on Info. Theory*, 63(4):2113–2136, April 2017.

[34] J. Scarlett and V. Cevher. Near-optimal noisy group testing via separate decoding of items. In *IEEE ISIT*, pages 2311–2315, 2018.

[35] J. Scarlett and O. Johnson. Noisy non-adaptive group testing: A (near-)definite defectives approach. *IEEE Trans. on Info. Theory*, 66(6):3775–3797, June 2020.

[36] J. Scarlett. Noisy adaptive group testing: Bounds and algorithms. *IEEE Trans. on Info. Theory*, 65(6):3646–3661, 2019.

[37] M. Cheraghchi. Improved constructions for non-adaptive threshold group testing. In *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming*, page 552–564, 2010.

[38] P. Damaschke. Threshold group testing. In *General Theory of Information Transfer and Combinatorics*, pages 707–718. Springer, 2006.

[39] D. Sejdinovic and O. T. Johnson. Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction. In *48th Annual Allerton Conference on Communication, Control and Computing*, pages 998 – 1003, 2010.

[40] T. Berger and V. I. Levenshtein. Asymptotic efficiency of two-stage disjunctive testing. *IEEE Transactions on Information Theory*, 48(7):1741–1749, 2002.

[41] P. Damaschke and A. S. Muhammad. Randomized group testing both query-optimal and minimal adaptive. In *Proceedings of the 38th International Conference on Current Trends in Theory and Practice of Computer Science*, page 214–225, 2012.

[42] M. Mezard and C. Toninelli. Group testing with random pools: Optimal two-stage algorithms. *Information Theory, IEEE Transactions on Information Theory*, 57(04):1736 – 1745, 2011.

[43] M. Aldridge. Conservative two-stage group testing, 2020. Available at arXiv: 2005.06617.

[44] M. Cheraghchi, R. Gabrys, and O. Milenkovic. Semiquantitative group testing in at most two rounds, 2021. Available at arXiv: 2102.04519.

[45] M. Cheraghchi, A. Karbasi, S. Mohajerzefreh, and V. Saligrama. Graph-constrained group testing. *IEEE Transactions on Information Theory*, 58, 2010.

[46] E. Karimi, F. Kazemi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson. Non-adaptive quantitative group testing using irregular sparse graph codes. In *Allerton Conference*, September 2019.

[47] P. Johann. A group testing problem for graphs with several defective edges. *Discrete Applied Mathematics*, 117:99–108, 2002.

[48] K. Lee, R. Pedarsani, and K. Ramchandran. Saffron: A fast, efficient, and robust framework for group testing based on sparse-graph codes. In *IEEE ISIT*, July 2016.

[49] H. Q. Ngo, E. Porat, and A. Rudra. Efficiently decodable error-correcting list disjunct matrices and applications. In *Proceedings of the 38th International Colloquim Conference on Automata, Languages and Programming*, page 557–568, 2011.

[50] S. Bondorf, B. Chen, J. Scarlett, H. Yu, and Y. Zhao. Sublinear-time non-adaptive group testing with $o(k \log n)$ tests via bit-mixing coding. Available at arXiv: 1904.10102.

[51] H. A. Inan, P. Kairouz, M. Wootters, and A. Ozgur. On the optimality of the kautz-singleton construction in probabilistic group testing. In *Allerton Conference*, October 2018.

[52] H. A. Inan and A. Ozgur. Strongly explicit and efficiently decodable probabilistic group testing. In *IEEE ISIT*, June 2020.

[53] H. A. Inan, P. Kairouz, and A. Ozgur. Sparse combinatorial group testing. *IEEE Transactions on Information Theory*, 66(5):2729–2742, 2020.

[54] J. Hayes. An adaptive technique for local distribution. *IEEE Transactions on Communications*, 26(8):1178–1186, 1978.

[55] T. Berger, N. Mehravari, D. Towsley, and J. Wolf. Random multiple-access communication and group testing. *IEEE Transactions on Communications*, 32(7):769–779, 1984.

[56] A. De Bonis and U. Vaccaro. Constructions of generalized superimposed codes with applications to group testing and conflict resolution in multiple access channels. *Theoretical Computer Science*, 306(1):223–243, 2003.

[57] S. Wu, S. Wei, Y. Wang, R. Vaidyanathan, and J. Yuan. Partition information and its transmission over boolean multi-access channels. *IEEE Trans. on Info. Theory*, 61(2):1010–1027, February 2015.

[58] G. Atia, S. Aeron, E. Ermis, and V. Saligrama. On throughput maximization and interference avoidance in cognitive radios. In *2008 5th IEEE Consumer Communications and Networking Conference*, pages 963–967, 2008.

[59] N. J. A. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. W. S. Chan. Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 697–705, 2007.

[60] A. Sharma and C. R. Murthy. Group testing-based spectrum hole search for cognitive radios. *IEEE Transactions on Vehicular Technology*, 63(8):3794–3805, 2014.

[61] J. Robin and E. Erkip. Capacity bounds and user identification costs in rayleigh-fading many-access channel, 2021. Available at arXiv: 2105.05603.

[62] J. Robin and E. Erkip. Access delay constrained activity detection in massive random access, 2021. Available at arXiv: 2111.03051.

[63] O. Yildiz, A. Khalili, and E. Erkip. Hybrid beam alignment for multi-path channels: A group testing viewpoint, 2021. Available at arXiv: 2111.08159.

[64] L. Ma, T. He, A. Swami, D. Towsley, K. K. Leung, and J. Lowe. Node failure localization via network tomography. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, page 195–208. Association for Computing Machinery, 2014.

[65] W. Xu, M. Wang, E. Mallada, and A. Tang. Recent results on sparse recovery over graphs. In *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 413–417, 2011.

[66] C. Lo, M. Liu, J. P. Lynch, and A. C. Gilbert. Efficient sensor fault detection using combinatorial group testing. In *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 199–206, 2013.

[67] M. Goodrich and D. Hirschberg. Improved adaptive group testing algorithms with applications to multiple access channels and dead sensor diagnosis. *Journal of Combinatorial Optimization*, 15, 05 2009.

[68] V. Arrigoni, N. Bartolini, A. Massini, and F. Trombetti. Static and dynamic failure localization through progressive network tomography, 2021. Available at arXiv: 2103.17221.

[69] J. Robin and E. Erkip. Sparse activity discovery in energy constrained multi-cluster iot networks using group testing, 2021. Available at arXiv: 2103.16174.

[70] A. Emad, K. R. Varshney, and D. M. Malioutov. A semiquantitative group testing approach for learning interpretable clinical prediction rules. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2015.

[71] S. Dash, D. M. Malioutov, and K. R. Varshney. Learning interpretable classification rules using sequential rowsampling. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3337–3341, 2015.

[72] D. Malioutov and K. Varshney. Exact rule learning via boolean compressed sensing. In *International Conference on Machine Learning (ICML)*, June 2013.

[73] A. Emad and O. Milenkovic. Poisson group testing: A probabilistic model for nonadaptive streaming boolean compressed sensing. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3335–3339, 2014.

[74] M. Shi, T. Furon, and H. Jegou. A group testing framework for similarity search in high-dimensional spaces. In *Proceedings of the 22nd ACM International Conference on Multimedia*, page 407–416, 2014.

[75] J. Engels, B. Coleman, and A. Shrivastava. Practical near neighbor search via group testing, 2021. Available at arXiv: 2106.11565.

[76] E. S. Hong and R. E. Ladner. Group testing for image compression. *IEEE Transactions on Image Processing*, 11(8):901–911, 2002.

[77] Y.-W. Hong and A. Scaglione. Group testing for sensor networks: The value of asking the right questions. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1297–1301, 2004.

[78] S. Khattab, S. Gobriel, R. Melhem, and D. Mosse. Live baiting for service-level dos attackers. In *IEEE INFOCOM - The 27th Conference on Computer Communications*, pages 171–175, 2008.

[79] Y. Xuan, I. Shin, M. T. Thai, and T. Znati. Detecting application denial-of-service attacks: A group-testing-based approach. *IEEE Transactions on Parallel and Distributed Systems*, 21(8):1203–1216, 2010.

[80] G. Cormode and S. Muthukrishnan. What's hot and what's not: Tracking most frequent items dynamically. *ACM Transactions on Database Systems*, 30(1):249–278, 2005.

[81] R. Clifford, K. Efremenko, E. Porat, and A. Rothschild. Pattern matching with don't cares and few errors. *Journal of Computer and System Sciences*, 76:115–124, 2010.

[82] A. Macula and L. Popyack. A group testing method for finding patterns in data. *Discrete Applied Mathematics*, 144:149–157, 11 2004.

[83] J. Wang, E. Lo, and M. L. Yiu. Identifying the most connected vertices in hidden bipartite graphs using group testing. *IEEE Transactions on Knowledge and Data Engineering*, 25(10):2245–2256, 2013.

[84] N. H. Bshouty and A. Costa. Exact learning of juntas from membership queries. In *Algorithmic Learning Theory*, pages 115–129. Springer International Publishing, 2016.

[85] A. Ambainis, A. Belovs, O. Regev, and R. De Wolf. Efficient quantum algorithms for (gapped) group testing and junta testing. In *27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 903–922. Association for Computing Machinery, 2016.

[86] 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.

[87] M. R. Stan, P. D. Franzon, S. C. Goldstein, J. C. Lach, and M. M. Ziegler. Molecular electronics: From devices and interconnect to circuits and architecture. *Proceedings of the IEEE*, 91(11):1940–1957, 2003.

[88] S. D. Lendle, M. G. Hudgens, and B. F. Qaqish. Group testing for case identification with correlated responses. *Biometrics*, 68(2):532–540, June 2012.

[89] T. Li, C. L. Chan, W. Huang, T. Kaced, and S. Jaggi. Group testing with prior statistics. In *IEEE ISIT*, July 2014.

[90] Y-J. Lin, C-H. Yu, T-H. Liu, C-S. Chang, and W-T. Chen. Positively correlated samples save pooled testing costs. Available at arXiv:2011.09794.

[91] E. Nebenzahl and M. Sobel. Finite and infinite models for generalized group-testing with unequal probabilities of success for each item. In *Discriminant Analysis and Applications*, pages 239–289. Elsevier, 1973.

[92] M. Doger and S. Ulukus. Group testing with non-identical infection probabilities. In *IEEE REDUNDANCY*, October 2021.

[93] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. Diggavi. Group testing for connected communities. In *AISTATS*, April 2021.

[94] P. Nikolopoulos, S. R. Srinivasavaradhan, T. Guo, C. Fragouli, and S. Diggavi. Group testing for overlapping communities. In *IEEE ICC*, June 2021.

[95] H. Nikpey, J. Kim, X. Chen, S. Sarkar, and S. S. Bidokhti. Group testing with correlation under edge-faulty graphs. Available at arXiv:2202.02467.

[96] B. Arasli and S. Ulukus. Group testing with a graph infection spread model. *Information, special issue on Advanced Technologies in Storage, Computing, and Communication*, 14(1), January 2023.

[97] S. Ahn, W.-N. Chen, and A. Ozgur. Adaptive group testing on networks with community structure. In *IEEE ISIT*, July 2021.

[98] M. Gonen, M. Langberg, and A. Sprintson. Group testing on general set-systems. Available at arXiv:2202.04988.

[99] T. B. Idalino and L. Moura. Structure-aware combinatorial group testing: A new method for pandemic screening. Available at arXiv:2202.09264.

[100] I. Lau, J. Scarlett, and Y. Sun. Model-based and graph-based priors for group testing. Available at arXiv:2205.11838.

[101] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi. Dynamic group testing to control and monitor disease progression in a population. Available at arXiv:2106.10765.

[102] S. R. Srinivasavaradhan, P. Nikolopoulos, C. Fragouli, and S. Diggavi. An entropy reduction approach to continual testing. In *IEEE ISIT*, July 2021.

[103] M. Doger and S. Ulukus. Dynamical Dorfman testing with quarantine. In *CISS*, March 2022. Also available at arXiv:2201.07204.

[104] B. Arasli and S. Ulukus. Group testing with a dynamic infection spread. In *IEEE ISIT*, July 2022.

[105] B. Arasli and S. Ulukus. Dynamic saffron: Disease control over time via group testing. *Algorithms, special issue on Combinatorial Optimization, Graph, and Network Algorithms*, 15(11), November 2022.

[106] M. Aldridge. Pooled testing to isolate infected individuals. In *CISS*, March 2021.

[107] B. Arasli and S. Ulukus. Graph and cluster formation based group testing. In *IEEE ISIT*, July 2021.

[108] B. Arasli and S. Ulukus. Dynamic infection spread model based group testing. *Algorithms, special issue on Combinatorial Optimization, Graph, and Network Algorithms*, 16(1), January 2023.

[109] K. Banawan, B. Arasli, Y.-P. Wei, and S. Ulukus. The capacity of private information retrieval from heterogeneous uncoded caching databases. *IEEE Transactions on Information Theory*, 66(6):3407–3416, June 2020.

[110] Y.-P. Wei, B. Arasli, K. Banawan, and S. Ulukus. The capacity of private information retrieval from decentralized uncoded caching databases. *Information, special issue on Private Information Retrieval: Techniques and Applications*, 10(12), November 2019.

[111] K. Banawan, B. Arasli, Y.-P. Wei, and S. Ulukus. Private information retrieval from heterogeneous uncoded caching databases. In *IEEE ISIT*, July 2019.

[112] Y.-P. Wei, B. Arasli, K. Banawan, and S. Ulukus. Private information retrieval from decentralized uncoded caching databases. In *IEEE ISIT*, July 2019.

[113] K. Banawan, B. Arasli, and S. Ulukus. Improved storage for efficient private information retrieval. In *IEEE Information Theory Workshop (ITW)*, August 2019.