The Dissertation Committee for Karthikeyan Shanmugam
certifies that this is the approved version of the following dissertation:

Graph Theoretic Results on Index Coding, Causal Inference and Learning Graphical Models

Committee:

Georgios-Alex Dimakis, Supervisor

Sujay Sanghavi

Sanjay Shakkottai

Constantine Caramanis

David Zuckerman

Graph Theoretic Results on Index Coding, Causal Inference and Learning Graphical Models

by

Karthikeyan Shanmugam, B.Tech.; M.Tech.; M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2016

To my late father and my mother.

Acknowledgments

I am extremely fortunate and grateful to have a great advisor like Alex Dimakis. He has been a great mentor and an inspiring role-model for me throughout my graduate study. His energy and enthusiasm to investigate new ideas, pursue them and formulate new problems has been very infectious and due to that I have come quite far in terms of research from where I started. There were many lean moments in my graduate life, when it was difficult to keep focus and persist owing to many reasons. He has been extremely patient, understanding and very encouraging in those tough times. In the last six years, I have primarily learnt the ropes of many aspects of research from him. When I came in as a graduate student I was ignorant about the actual mechanisms behind research. He was very instrumental in making me realize that one must not lose sight of making small but definite progress by engaging too much in trying to find grand proofs and results. Particularly, this bottom up approach was very crucial to my initial progress. I also would like to thank Sujay Sanghavi, Sanjay Shakkottai, David Zuckerman and Constantine Caramanis for agreeing to be in my committee and for many helpful suggestions regarding this dissertation and my graduate study. I have also had the great fortune to collaborate with other faculty members including Adam Klivans, Pradeep Ravikumar, Sujay Sanghavi and Sanjay Shakkottai over the years. I would like to thank Sanjay Shakkottai, David Zuckerman and Adam Klivans for teaching me three memorable courses which I enjoyed greatly. The WNCG group at UT Austin is an awesome place to be, work and collaborate. I am very indebted to Gulick Melanie, Melody Singleton and Karen Little for taking care of so many administrative and logistic issues throughout my time as a PhD student.

I had the fortune to be mentored very closely by Giuseppe Caire while I was at University of Southern California. I am thankful for many fruitful interactions with Andy Molisch and Michael Neely. My internship with Antonia Tulino and Jaime Llorca at Bell Labs was extremely fruitful and I am indebted to their mentorship. I am also extremely grateful to Michael Langberg for one of the most productive collaborations that set the pace for a lot of work in this dissertation. A significant part of my thesis was inspired by a single talk he gave at a workshop in 2011. I have had the fortune of having great student collaborators from whom I learnt a lot from close quarters. My group mate and a senior, Dimitris Papailiopoulos is a great inspiring influence. I have had the fortune to collaborate with Negin, Mingyue, Dimitris, Megasthenis, Murat, Rajat, Rashish, Ethan and Michael Borokhovich. It was a great delight to have the company of 'comrade-in-arms' Megasthenis who has been a great friend. He had to put up with my rants on life, research and beyond for six long years. I have long admired and learnt much from Ankit, Srinadh and Pravesh who were my peers and also from Siddhartha, Sharayu, Abhik, Praneeth and Avishek who were my seniors. Murat, Subashini, Suriya, Derya, Soumya, Rajat, Avik, Erik, Ethan, Abishek, Mandar, Rashish, Tasos and Rajiv have given me great company in and out of the lab. I am also incredibly lucky to have had Dinesh, Aditya, Siva, Akila, Akarsh, Sashvat, Gurpreet, Menal, Kushboo, Salomi, Shilpi and Neeraj who were such awesome friends instrumental in keeping my sanity during my stay at UT. A special mention should go to my roommate Himanshu who among many other things was responsible for embellishing my culinary skills. I am thankful to Koushik, Siddharth, Arjun, Deepan, Mmanu, Sriram and Saran who have kept regularly in touch with me throughout my graduate study from afar. I am greatly indebted to my USC gang of friends comprising Gopi, Srinivas, Sanjay, Aditya, Niharika, Harish, Akshara, Sunav, Chiranjeeb and Nachikethas for helping me greatly in my initial days in the US and for being such awesome friends. Srinivas, Sanjay and Gopi took a lot of trouble giving me sporadic 'orientation lessons' when I was very new to the country.

I would like to thank Srikrishna Bhashyam, Andrew Thangaraj, Harishankar Ramachandran and C.S. Ramanlingam from my undergrad days at IIT Madras for great mentorship and guidance throughout my undergrad without which I would not have pursued a doctoral degree. The initial motivation to pursue a PhD was simply because of the excellent faculty, courses and teaching at the EE department in IIT Madras. My extended family in India provided me great direct and indirect support. Usha Periyamma, Natarajan Chittappa and Uma Periyamma and their family were great pillars of strength for my mother while I was away from home. They also have been my lifelong well wishers. I am very grateful for the support and encouragement from Vijay, Gayathrie, Sree Mama and Arjun who form my family away from home. My late father and my mother made incredible sacrifices on my account all throughout their lives. I am more than grateful and lucky to have them as my parents. They fought great odds in their lives to provide me with the freedom to pursue my dreams. When my father fell ill back home in the middle of my graduate study, my mother left no stone unturned to make sure that my graduate study would not get affected. She had to endure too many hardships on my being away for the last few years. Her indomitable will, determination and mental strength to surmount obstacles is a great source of inspiration for me and I would never be able to repay my debt to her.

Graph Theoretic Results on Index Coding, Causal Inference and Learning Graphical Models

Karthikeyan Shanmugam, Ph.D.
The University of Texas at Austin, 2016

Supervisor: Georgios-Alex Dimakis

Exploiting and learning graph structures is becoming ubiquitous in Network Information Theory and Machine Learning. The former deals with efficient communication schemes in a many-node network. In the latter, inferring graph structured relationships from high dimensional data is important. In this dissertation, some graph theoretic results in these two areas are presented. The first part deals with the problem of optimizing bandwidth resources for a shared broadcast link serving many users each having access to cached content. This problem and its variations are broadly called Index Coding. Index Coding is fundamental to understanding multi-terminal network problems and has applications in networks that deploy caches. The second part deals with the resources required for learning a network structure that encodes distributional and causal relationships among many variables in machine learning. The number of samples needed to learn graphical models that capture crucial distributional information is studied. For learning causal relationships, when passive data acquisition is not sufficient, the number of interventions required is investigated.

In the first part, efficient algorithms for placing popular content in a network that deploys a distributed system of caches are provided. Then, the Index Coding problem is considered: every user has its own cache content that is given and transmissions on a shared link are to be optimized. All graph theoretic schemes for Index Coding, known prior to this work, are shown to perform within a constant factor from the one based on graph coloring. Then, ‘partial’ flow-cut gap results for information flow in a multi-terminal network are obtained by leveraging Index Coding ideas. This provides a poly-logarithmic approximation for a known generalization of multi-cut. Finally, optimal cache design in Index Coding for an adversarial demand pattern is considered. Near-optimal algorithms for cache design and delivery within a broad class of schemes are presented.

In the second part, sample complexity lower bounds considering average error for learning random Ising Graphical Models, sampled from Erdős-Rényi ensembles, are obtained. Then, the number of bounded interventions required to learn a network of causal relationships under the Pearls model is studied. Upper and lower bounds on the number of size bounded interventions required for various classes of graphs are obtained.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In this dissertation, we consider and present results for graph problems that arise in two broad areas: 1) Communication problems in Network Information theory related to bottlenecked broadcast networks with cached side information (variations of the *Index Coding* problem) and 2) Sample complexity requirements for learning Graphical Models that encode dependencies among variables from data and the number of experiments required to learn a network of causal relationships. Problems in both parts are high dimensional in nature. The structure underlying this high dimensional setting is captured in the form of a graph.

In Network Information theory, the most fundamental problem is to understand efficient communication schemes that enable a set of k sources to communicate to k destinations through a directed network of noiseless links. This is called the Multiple Unicast Network Coding (MU problems) Problem. In this problem, any efficient communication scheme depends on the explicit network structure. In this work, we consider only a special case of this problem that involves communicating through a single shared noiseless broadcast link to a large number of destinations. Each destination has some cached side-information, i.e. a subset of the messages that other users have desired. The source can potentially 'mix' various files and send through the common link which can be unscrambled by every user to get its desired file by exploiting side information. This is called the Index Coding problem. Although the network topology is not complicated, the side-information pattern involves a hidden graph structure that captures the actual complexity behind the problem. Interestingly, it turns out that this hidden graph structure is rich enough that any MU problem can be reduced to an Index Coding problem. Further, Index Coding problem mirrors some issues that wireless networks face today and hence it has become important to study variations of the Index Coding problem which we do here. All results in the first part are graph theoretic in nature. Most of the results analyze algorithms that compute lower and upper bounds on the communication rates for this problem. Analysis in this work, mostly depend on exploiting specific graph structures hidden in the problem or analyzing graph structures behind some extreme examples to establish some universal results about all graphs.

In Machine Learning, most algorithms try to learn two types of relationships between variables from data. One is distributional information in the form of a latent structure that can explain the data so that further inference can be made from this latent structure. The other type is causal relationships between variables of interest, i.e. questions of the type - Does variable X cause Y ? We consider a subset of problems relating to this theme in the first part. For the first case, we consider one of the most important latent structures: a graphical model that encodes conditional independence relationships. We study the sample complexity requirements to learn ferromagnetic Ising Graphical Models in particular. The most important contribution is that we derive lower bounds considering the average error when the graphical model is sampled from a random ensemble of graphs. Previous lower bounds relied on pathological counter examples from a class of graphs because the worst case error in learning a graph from a class of graphs was considered. We also investigate the number of experiments of bounded size needed to recover causal relationships between variables. It is known that experiments/interventions are required to learn causal relationships as opposed to distributional information that can be learnt from data. We derive algorithms and also lower bounds for the number of experiments needed as a function of the causal network to be learnt.

We now outline the results in both parts of the dissertation in detail in the next few sections.

1.2 Part 1 : Bottlenecked broadcast networks with side information

We start with a practical motivation for the results in the first part: Today's wireless networks are heavily band limited while they have to carry large files (e.g. videos) to users who demand them in a macro cell. At the 'wireless edge', a typical scenario is that several receivers request content while a single macro-base station services these requests using a common broadcast channel. This creates a bottleneck at the broadcast link. One of the solutions, is to cache popular content at the edge, i.e. among receivers or at other places like WiFi offloading points that have a faster connection to the users. If users already cache or have faster access to cache that contain the request, then this *cache hit* relieves the bottlenecked broadcasting agent.

First, we explore how to optimize aggregate cache hits in a system where users are distributed in a network with access to distributed caches with cache constraints. Every user accesses only a subset of all the caches. Since a user sees multiple but few distributed caches, it is not clear how to place popular files in a distributed manner. This depends on the topology governing user-cache connectivity. We consider our first question:

Question 1: *How to cache popular files in a system of distributed caches so as to maximize the aggregate cache hit for multiple receivers each of which sees a subset of the caches deployed ?*

We only consider a stylized abstract coverage problem that addresses this issue. The solution involves exploiting structural properties of the user-cache topology along with the nature of constraints (type of constraints akin to an assignment problem) in the resulting coverage problem.

Now, consider the scenario where every user only has its own cache. It seems that there is nothing to optimize in terms of cache hits. Every user would cache identically (the most popular files). However, there is another interesting possibility: Since everyone gets their request from a common channel, even if the users do not have cache hits, using coded transmissions (mixing of various requests) at the common channel, it is possible to use unrelated files from a user's cache (or the one the user has access to) to decode the request. Essentially, because of the 'broadcast medium' one can go beyond the benefit from *local cache hits*. In a simple example, two users A and B request packets p_A and p_B respectively. They have each other's request in the cache. Then the common channel can optimize the number of transmissions and send the XOR of both packets although both users do not have local cache hits, i.e. their individual caches cannot satisfy their demands. This saves one transmission as every user can *still* decode its request. We ignore the placement question (what is to be cached) and assume that every user cache contains some arbitrary cached content. We concentrate on optimizing coded transmissions. The problem happens to be graph theoretic - the cached information state (relevant to the problem) and requests can be represented as a *side information graph*. This problem is called *Index Coding*. This is a stylized problem that addresses the optimization involved in efficient mixing of requests in a broadcast network so that they can be decoded at the edge using side information. There have been many purely graph theoretic schemes and few algebraic schemes resulting in codes for transmissions proposed for this problem. We answer the following question:

Question 2: *How do purely graph theoretic schemes (fractional coloring and partial clique cover based schemes) for Index Coding compare against each other?*

The answer to this question involves the following basic idea: Comparing two combinatorial schemes on *any* graph can sometimes be studied by comparing the schemes for extreme cases. We introduce new graph theoretic schemes based on the *local chromatic number* for this problem that outperforms existing ones. We identify extreme cases where the new scheme and the old ones are the farthest from each other. The structural property of these extreme cases show that all the graph theoretic schemes (known before this work) are within a constant factor of each other.

Then, we move onto a set of results that further highlight the importance of Index Coding as a central problem in Network Information Theory. The Index Coding problem is a very important fundamental problem in Information Theory for the following reason: If Index Coding is exactly solved, then it is possible to solve all *Network coding* problems as noted before. In Network Coding, efficient communication scheme needs to be designed that sends messages from a set of sources to a set of destinations through a directed network with capacitated links. A special but representative case of that is multiple-unicast network coding where independent information needs to be sent from k sources to their unique corresponding destinations over a directed network. The aim is to get the best aggregate rate for all information flows. In the case of commodity flows, *mincuts* and *multi-cuts* characterize upper bounds and flow-cut gaps are known. For information flow, a new edge-cut bound called GNS (Generalized Network Sharing) cut is a generalization. We answer the following questions:

Question 3: *Can GNS-cut be approximated for a k-unicast problem over an acyclic network ? What is the gap between information flow and GNS-cut ?*

We show that the GNS-cut can be approximated within a poly-log factor in the number of sources k. The answer involves first, relating index codes to a class of codes from distributed storage called Locally repairable codes and then to Network Coding. We show that the approximation to GNS-cut essentially is a solution to the correlated information flow problem. When independence among messages is desired, for linear solutions the field matters and we show that fixing a particular field can be arbitrarily bad. Some of these results exploit specific properties of the network in a k-unicast problem and structural properties of some special Index Coding examples known in the literature.

Next, we consider another variation of the Index Coding problem where cache design and delivery are *both* optimized simultaneously. There are K users served by a common noiseless broadcast link. All the user requests are known to arise from a library of N files. Every user cache has a memory of size M. The objective is to design a caching scheme without knowing the request pattern such that, over all possible demand patterns from the library, the number of file transmissions in the worst case (called *peak rate*) is optimized. This problem has been studied in the asymptotic setting where the file size F is exponentially large in K, M and N or when F goes to infinity. It has been shown that the peak rate can be at most $\frac{N}{M}$, i.e. independent of the number of users. There is a class of caching schemes wherein every user caches parts of every file randomly and independently from other users. The peak average rate is still bounded by $\frac{N}{M}$ in this uncoordinated setting. A simple greedy graph coloring based index coding solution suffices in the delivery stage. We investigate the file size requirements F needed to achieve a target coding gain g (leading to a peak rate of $\frac{K}{g}$). We answer the following question:

Question 4: *What is the smallest F needed for the peak average rate to be at most $\frac{K}{g}$, resulting in a coding gain of g, for random uncoordinated caching schemes and coloring based delivery schemes?*

We show that the file size has to be exponential in g, i.e. $F = \Omega(\frac{1}{K}\left(\frac{N}{M}\right)^g)$. For a fixed number of users, the file size grows exponentially in g. We also propose a novel delivery scheme that modifies the existing greedy clique cover scheme such that $F = \Theta\left(\left(\frac{3eN}{M}\right)^{g+1}(\log(N/M))^{g+2}\right)$ approximately achieving the right scaling. Further, we show that even when $F = \Theta\left(\exp\left(\frac{KM}{N}\right)\right)$, previous algorithms with random placement schemes give at most a gain of 2. We demonstrate the effectiveness of this algorithm through numerical simulations comparing with existing algorithms.

1.3 Part 2: Bounds for Learning Graphical Models and Causal Networks

A ubiquitous statistical task in Machine Learning is inference (e.g. likelihood estimation) from uncertain data. When data consists of samples containing many variables (millions), then these inference tasks can be speeded up if a sparse graph structure encoding conditional independencies among variables is known. Undirected graphs with edges connecting the vertices representing variables is a way of representing conditional independencies. These are called Graphical Models. They are widely used in a number domains including natural language processing [1], image processing [2–4], statistical physics [5], and spatial statistics [6], gene-regulatory networks [7] and even in coding theory for decoding LDPC ensembles. Learning the graph structure from only samples is also an important task in view of the above uses of graphical models for inference purposes. In this work, we consider a general family of parameterized graphical models on boolean variables - Ising Models.

In this part of our proposal, we first consider *ferromagnetic Ising Models* on an undirected graph G drawn from a class of graphs. Previous works, have considered the class of graphs which are degree bounded and have derived sample complexity requirements for learning the structure of the graph in the worst case. Usually, these results indicate the range of inverse-temperatures (a real parameter that is associated with the Ising Model) in which learning is feasible (polynomial instead of exponential). Since the entire graph class is considered and we are interested in learning all graphs in the class, a small pathological subset of graphs, which are difficult, determine the sample complexity requirements. It is important to understand more real-life graph ensembles like power law graphs and that too in a typical sense (when the power law graph is obtained using say a random model like the Chung-Lu model). In this work, we consider an easier version of a random ensemble, namely Erdős-Rényi graphs. We show how to derive sample complexity bounds for average error when the graphs are drawn randomly from this ensemble. Pathological cases cannot be used to reason about sample complexity in such random ensembles. The question we are interested in is :

Question 4: *Under what inverse temperature ranges, learning a random graphical model from an Erdős-Rényi ensemble statistically requires huge number of samples (exponential in the number of variables or a bad polynomial scaling) ?*

We identify typical graph structures that enable us to answer this question for dense Erdős-Rényi ensembles. We show that when $G \sim G(p, c/p)$, $c = \Omega(p^{3/4+\epsilon'})$ and $\lambda = \Omega(\sqrt{p}/c)$, number of samples exponential in $\frac{\lambda^2 c^2}{p}$ is required.

Hence, for any efficient algorithm, we require $\lambda = O\left(\sqrt{p}/c\right)$ and in this regime $O\left(c \log p\right)$ samples are required to learn.

We then consider the problem of learning a network of causal relationships between variables. It is known that mere observations are not enough to identify causal relationships amongst variables. Experiments or Interventions are needed. y is said to cause x if $x = f(y, e)$ where y and x are random variables observed while e is a latent but an independent noise variable. However, the direction of causality cannot be determined from just distributional information of (x, y) since it may be possible to write $y = g(x, e')$ for some other independent variable e' and function g. When we intervene on the variable x, we force the variable to take a random independent value and rest of the system is undisturbed. If the first set of functional relations hold, then clearly after intervention x and y are independent which we can test. In the second model, even post-intervention there exists some correlation between x and y. In general, we adopt the Pearl's structural equation model for causal relationships among a set of variables which is a system of functional equations over many variables. There is a natural directed graph G where directions of the edges capture the set of causal relationships. It is possible to intervene on a set of k variables simultaneously by forcing each variable to take a random independent value. We consider the following question:

Question 5: *What is the minimum number of size-k interventions needed before we can learn the directions of all edges in a causal network G on n vertices?*

First, we consider the case when the skeleton of G (the undirected graph when directions are removed) is complete. We basically show that any adaptive algorithm that decides the sequence of interventions is equivalent to a non-adaptive combinatorial construction called the *separating system*. We also construct new separating systems which are close to optimal. Further, we also derive information theoretic lower bounds on all randomized adaptive algorithms and show that separating systems are optimal upto log factors even in this case. We derive new lower bounds for general graph skeletons. We show some extreme examples of sparse graphs where the number of interventions needed is that of a separating system on n nodes far away from the lower bound. We also propose an algorithm that performs very well empirically close to the lower bound, when the graphs are chosen randomly avoiding such extreme counter examples. We also show some theoretical guarantees for this algorithm.

1.4 Organization

Chapter 2 deals with the first problem of the first part, i.e. the coverage problem dealing with placing content in a distributed system of caches. The next chapter introduces the Index Coding problem and derives universal comparative results on performance of graph theoretic schemes known for this problem. Chapter 4 deals with the question of approximating sum-rate of network coding problems by leveraging connections to the Index Coding problem and other problems in distributed storage. Chapter 5 considers the problem of Index Coding with cache design. Part 2 of the dissertation starts from Chapter 6. This deals with the problem of deriving sample complexity bounds for learning a random Ising graphical model drawn randomly from the ensemble of Erdős-Rényi graphs. Chapter 7 deals with the problem of learning a causal network with bounded Interventions.

1.5 Publications related to the Dissertation

This dissertation is based on the following publications:

1. K. Shanmugam, N. Golrezaei , A.G. Dimakis, A.F. Molisch and G. Caire, " FemtoCaching: Wireless Content Delivery through Distributed Caching Helpers ". *IEEE Transactions on Information Theory*, 8402-8413, Vol:59(12), Dec 2013. (Chapter 2)

2. K. Shanmugam, A.G. Dimakis and M. Langberg, " Local Graph Coloring and Index Coding". *International Symposium on Information Theory (ISIT 2013)*, Istanbul, 2013.

   (and)

   K. Shanmugam, A.G. Dimakis and M. Langberg, " Graph Theory versus Minimum Rank for Index Coding". *International Symposium on Information Theory (ISIT 2014)*, HI, 2014. (journal version under preparation, Chapter 3)

3. K. Shanmugam, M. Asteris and A.G. Dimakis, " On approximating the sum-rate for multiple unicasts". *International Symposium on Information Theory (ISIT 2015)*, HongKong, 2015. **(and)**

   K. Shanmugam and A.G. Dimakis," Bounding Multiple Unicasts through Index Coding and Locally Repairable Codes". *International Symposium on Information Theory (ISIT 2014)*, HI, 2014. (journal version under preparation, Chapter 4).

4. K. Shanmugam, M. Ji, A.M. Tulino, J. Llorca and A.G. Dimakis, " Finite Length Analysis of Caching-Aided Coded Multicasting", *Allerton* 2014. (invited, journal version accepted to the Transactions on Information Theory, Chapter 5).

5. R. Tandon*, K. Shanmugam*, A. G. Dimakis, P. Ravikumar, " On the Information Theoretic Limits of Learning Ising Models", *NIPS, 2014* (*-equal contribution, Chapter 6).

6. K. Shanmugam*, M. Kocaoglu*, A.G.Dimakis and S. Vishwanath, "Learning Causal Graphs with Small Interventions", *NIPS 2015* (*-equal contribution, Chapter 7).

   Note: Arxiv versions of almost all of these papers are available. They can also be accessed at: *https://sites.google.com/a/utexas.edu/karthiksh/.*

Chapter 2

Femtocaching: A Coverage problem for Optimizing Cache-Content in a Bottlenecked Network

2.1 Introduction

Streaming [1] of video on-demand files is the main reason for the dramatic growth of data traffic over cellular networks – an increase of two orders of magnitude compared to current volume is expected in the next five years [8]. It is widely acknowledged that conventional current (3G) and near-future (4G-LTE) macro-cell architectures will not be able to support such traffic increase, even after allocating new cellular spectrum. In a macro-cell deployment in wireless communication, the base station is going to be the *bottleneck*. Similar bottlenecks can be formed in a wired network which has a core bottlenecked source from which files have to be delivered to users. In this work, we will concentrate on the wireless case although the results apply broadly in the wired context too.

[1] The material in this chapter is based on the published journal article: K. Shanmugam, N. Golrezaei , A.G. Dimakis, A.F. Molisch and G. Caire, "FemtoCaching: Wireless Content Delivery through Distributed Caching Helpers ". *IEEE Transactions on Information Theory*, 8402-8413, Vol:59(12), Dec 2013. The dissertation author's main contributions are towards the intractability of the content placement coverage problem and the efficient approximation algorithm proposed for it besides contributions towards other results in the above paper. Results in this chapter are primarily based on these main contributions.

The most promising approach to achieve the system area spectral efficiency consists of shrinking the cell size and essentially *bringing the content closer to the users*, by deploying small base stations that achieve localized communication and enable high-density spatial reuse of communication resources [9]. Such pico- and femto-cell networks, which are usually combined with macrocells into a heterogeneous network, are receiving a lot of attention in the recent literature, (e.g., see [10] and references therein). A drawback of this approach, though, is the requirement for high-speed backhaul to connect the small cell access points to the core network [10] which in general is very expensive.

We study a problem arising out of an architecture which we propose, nicknamed *FemtoCaching*, that consists of replacing backhaul capacity with storage capacity at the small cell access points. Using caching at the wireless edge, highly predictable bulky traffic (such as video on-demand) can be efficiently handled. In this way, the backhaul is used only to refresh the caches at a much slower rate at which the users' demand distribution evolves over time. For example, special nodes with a large storage capacity and only wireless connectivity (no wired backhaul at all) can be deployed in a cell and refreshed by the serving cellular base station at off-peak times. These nodes, henceforth referred to as *helpers*, form a wireless distributed caching infrastructure.

While there are many issues that are very complex and would go well beyond the scope of this work, (e.g. see [10–13]), here we focus on a particular key system aspect: the content placement problem in the caches: *Which files should be cached by which helpers, given a certain network topology (helpers-users connectiv-*

ity) and file popularity distribution? In particular, we wish to minimize the expected total file downloading delay (averaged over the users' random demands) over the cache content placement when the delay is primarily due to the bottlenecked link. We refer to the problem as the *uncoded content placement problem* because only complete files are stored in the helper caches.

2.1.1 Related Work

Prior to this work, in the wireless context, the idea of using caching to support mobility in networks has been explored in [14–17]. The main underlying theme behind this body of work is to use caching at wireless access points to enable mobility in publish/subscribe networks. In other words, when a user moves from one location to another, the delay experienced by the user during "hand-off" between two access points can be minimized if the data is cached at the access points when the user connects to it. For more references, we refer the reader to references in [17]. In another line of work, [18,19], cache placement problems have been considered for the content distribution networks which form a distributed caching infrastructure on the wired network. There is a substantial amount of prior work on caching algorithms for web and video content (e.g., [20–22] and references therein). These prior works focus on wired networks, do not rely on content popularity statistics and do not have the topology aspects that are central in our formulation.

The scenario that allows coding (mixing packets) over the bottlenecked link, posed as an index coding problem, has been studied in [23,24] where a single base station is the only transmitter, the users cache content individually and user demands are arbitrary. We will explore this scenario of coding in the bottlenecked link in the next few chapters. Scaling laws of wireless networks with caching and device-device communication was recently explored in [25–27]. Further, [28] focuses on asymptotic scaling laws for joint content replication and delivery in ad-hoc wireless networks with multi-hop relaying and obtaining scaling laws for large networks.

2.1.2 Contributions

The contributions of this work are as follows:

- Uncoded FemtoCaching: We introduce and define the uncoded distributed caching problem as a special coverage problem that involves optimizing file placement in helpers to minimize the total average delay of all users. We show that finding the optimum placement of files is NP-complete.

- We express the problem as a maximization of a submodular function subject to matroid constraints [29], and describe approximation algorithms using connections to matroids and covering problems. In particular, we provide a low-complexity greedy algorithm with a $1/2$ approximation guarantee. Further, we exhibit another algorithm for a special case, which involves solving a Linear Program (LP) with an additional rounding step, that provides a $1 - (1 - 1/d)^d$ approximation to the placement problem where d is the maximum number of helpers connected to a user.

2.2 Distributed caching placement model and assumptions

We consider a region where some wireless User Terminals (UTs) place random requests to download files of a finite library from a set of dedicated content distribution nodes (helpers). The helpers have a limited cache size and limited transmission range, imposing topology constraints both on the content placement bipartite graph (involving files and helpers) and on the network connectivity bipartite graph involving helpers and UTs. We also assume the presence of a cellular base station (BS) which contains the whole library and can serve all the UTs in the system. Fig. 2.1 illustrates qualitatively the system layout. The key point is that if there is enough content reuse, *i.e.*, many users are requesting the same file, caching can replace backhaul communication.

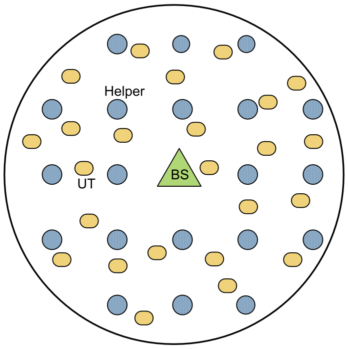


Figure 2.1: An example of the single-cell layout. UTs are randomly distributed, while helpers can be deterministically placed in the coverage region.

The content placement problem treated in this work can be formulated as follows: *for a given file popularity distribution, helper storage capacity and network topology, how should the files be placed in the helper caches such that the average sum downloading delay of all users is minimized?*

Since users experience shorter delay when they are served locally from helpers in their neighborhoods, minimizing the average delay for a given user is equivalent to maximizing the probability of finding the desired content in the neighboring helpers. The solution is trivial when there are few helpers in the cell, i.e. when each UT can connect only to a single helper. In this case, each helper should cache the most popular files. However, if the helper deployment is dense enough, UTs will be able to communicate with several such helpers and each sees a distributed cache given by the union of the helpers' caches. In this situation, the question on how to best assign files to different helpers becomes a much more complicated and interesting issue, because each UT sees a different, but correlated, distributed cache.

We define the set of helpers $\mathcal{H}$ of size $H+1$ (where the additional helper is the BS), the set of users $\mathcal{U}$ of size U and a library $\mathcal{F}$ comprising F files. The wireless network is defined by a bipartite graph $\mathcal{G} = (\mathcal{H}, \mathcal{U}, \mathcal{E})$ (see example in Fig. 2.2) where edges $(h, u) \in \mathcal{E}$ denote that a communication link exists from helper h to user u. We let $\mathcal{U}(h) \subseteq \mathcal{U}$ and $\mathcal{H}(u) \subseteq \mathcal{H}$ denote the sets of neighbors of helper h and user u, respectively. We assume that all users in the system can download from the BS, which is conventionally identified with helper $h = 0$. Hence, we have $\mathcal{U}(0) = \mathcal{U}$. The communication links $\{(h, u) : h \in \mathcal{H}(u)\}$ are characterized by different rates. In particular, we define the $(H + 1) \times U$ matrix $\mathbf{\Omega}$ with elements $\omega_{h,u}$ indicating the *average downloading time per information bit* for link $(h, u) \in \mathcal{E}$.

In this work, we assume that $\omega_{h,u} = \omega_1, \ \forall h \neq 0$. In other words, all helper user communication rates except that involving the base station is a constant signifying a high rate link compared to that between the base station and the user . We assume $\omega_{0,u} \geq \omega_1$ (i.e., for all UTs, the delay for downloading from the BS is larger than the delay from any other helper). Without loss of generality we set $\omega_{h,u}$ equal to an arbitrarily large constant $\omega_\infty \gg \max_u \omega_{0,u}$, for all $(h, u) \notin \mathcal{E}$ (i.e., non-existing links can be regarded as links for which downloading one bit of information takes an arbitrarily large amount of time). Without loss of fundamental generality, we assume that the files in the library $\mathcal{F}$ have the same size of B bits. A probability mass function $\{P_f : f = 1, \ldots, F\}$ is defined on $\mathcal{F}$, and we assume that users make independent requests to the files $f \in \mathcal{F}$ with probability P_f.

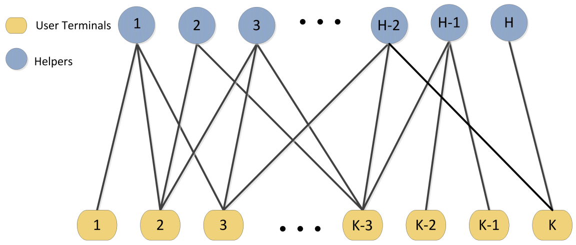


Figure 2.2: Example of a connectivity bipartite graph indicating how UTs are connected to helpers.

2.3 Coverage problem: Uncoded content placement

An uncoded cache placement is represented by a bipartite graph $\widetilde{\mathcal{G}} = (\mathcal{F}, \mathcal{H}, \widetilde{\mathcal{E}})$ such that an edge $(f, h) \in \widetilde{\mathcal{E}}$ indicates that a copy of file f is contained in the cache of helper h. We let $\mathbf{X}$ denote the $F \times H$ adjacency matrix of $\widetilde{\mathcal{G}}$, such that $x_{f,h} = 1$ if $(f, h) \in \widetilde{\mathcal{E}}$ and 0 otherwise. By the cache size constraint, we have that the column weight of $\mathbf{X}$ is at most M.

Consider a user u and its helper neighborhood $\mathcal{H}(u)$. We sort the links $(h, u) \in \mathcal{E}$ in increasing order such that $(j)_u$ denotes the j-th helper from user u (in some fixed but arbitrary order). By convention, we have $(|\mathcal{H}(u)|)_u = 0$ (the BS is sorted in $|\mathcal{H}(u)|$-th position by the *helper sorting function* $(\cdot)_u$) and for all $j > |\mathcal{H}(u)|$ we have $\omega_{(j)_u,u} = \omega_\infty$. [2] With this notation, the average delay per information bit for user u can be written as: [3]

$$\bar{D}_u = \sum_{j=1}^{|\mathcal{H}(u)|-1} \omega_1 \sum_{f=1}^{F} \left[\prod_{i=1}^{j-1} (1 - x_{f,(i)_u}) \right] x_{f,(j)_u} P_f$$
$$+ \omega_{0,u} \sum_{f=1}^{F} \left[\prod_{i=1}^{|\mathcal{H}(u)|-1} (1 - x_{f,(i)_u}) \right] P_f. \tag{2.1}$$

In order to see this, notice that $\left[\prod_{i=1}^{j-1}(1 - x_{f,(i)_u})\right] x_{f,(j)_u}$ is the indicator function (defined over the set of feasible placement matrices $\mathbf{X}$) for the condition that file f is in the cache of the helper $(j)_u$ (the j-th helper for user u), and it is not in any of the helpers with lower index $(i)_u$, for $i = 1, \ldots, j - 1$. Also, $\left[\prod_{i=1}^{|\mathcal{H}(u)|-1}(1 - x_{f,(i)_u})\right]$ is the indicator function for the condition that file f is not found in the neighborhood $\mathcal{H}(u)\backslash\{0\}$ of user u.

The minimization of the sum (over the users) average per-bit downloading delay can be expressed as the following integer programming problem:

$$\text{maximize} \qquad \sum_{u=1}^{U} \left(\omega_{0,u} - \bar{D}_u\right)$$

[2] By construction, the links $((j)_u, u)$ with $j > |\mathcal{H}(u)|$ do not exist in $\mathcal{E}$.

[3] We use the convention that the result of $\prod_{i=a}^{b}$ is 1 when $b < a$, and that the result of $\sum_{i=a}^{b}$ is zero when $b < a$.

$$\begin{aligned} \text{subject to} \quad & \sum_{f=1}^{F} x_{f,h} \leq M, \quad \forall \ h, \\ & \mathbf{X} \in \{0,1\}^{F \times H}. \end{aligned} \tag{2.2}$$

Now, we rewrite problem (2.2). Letting $\widetilde{\omega}_u = \omega_{0,u} - \omega_1$, the problem becomes:

$$\begin{aligned} \text{maximize} \quad & \sum_{f=1}^{F} P_f \sum_{u=1}^{U} \widetilde{\omega}_u \left[1 - \prod_{h \in \mathcal{H}(u): h \neq 0} (1 - x_{f,h}) \right] \\ \text{subject to} \quad & \sum_{f=1}^{F} x_{f,h} \leq M, \quad \forall \ h, \\ & \mathbf{X} \in \{0,1\}^{F \times H} \end{aligned} \tag{2.3}$$

where $\left[1 - \prod_{h \in \mathcal{H}(u): h \neq 0} (1 - x_{f,h})\right]$ is the indicator function (over the set of feasible $\mathbf{X}$) for the condition $f \in \mathcal{A}_u$, and $\mathcal{A}_u$ is the union of the helpers' caches in the neighborhood of user u, excluding the BS. The above objective function can be interpreted as the sum of *values* seen by each user. A value (the difference in download between the BS and the helper that is saved by caching) is seen by a user if it is in one of the neighborhood helpers (a coverage condition).

The value of each user u is equal to $\widetilde{\omega}_u \sum_{f \in \mathcal{A}_u} P_f$, which is proportional to the probability of finding a file in the union of the helpers' caches, multiplied by the incremental delay to download such files from the BS rather than from the helpers. Our goal here is to maximize the sum of values seen by all users.

2.3.1 Computational intractability

In this section, we show that problem (2.3) is NP-complete. To prove that the problem is NP-hard, we consider its corresponding decision problem, referred to as the *Helper Decision Problem.*

Problem 2.3.1. *(Helper Decision Problem) Given the network connectivity graph* $\mathcal{G} = (\mathcal{U}, \mathcal{H}, \mathcal{E})$*, the library of files* $\mathcal{F}$*, the popularity distribution* $\mathcal{P} = \{P_f\}$*, the set of positive real numbers* $\Omega = \{\widetilde{\omega}_u : u \in \mathcal{U}\}$ *and a real number* $Q \geq 0$*, determine if there exists a feasible cache placement* $\mathbf{X}$ *with cache size constraint* M *such that*

$$\sum_{u=1}^{U} \widetilde{\omega}_u \sum_{f \in \mathcal{A}_u} P_f \geq Q. \tag{2.4}$$

Let the problem instance be denoted by $\mathrm{HLP}(\mathcal{G}, \mathcal{F}, \mathcal{P}, \Omega, M, Q)$. ◊

It is easy to see that the helper decision problem is in the class NP. To show NP-hardness, we will use a reduction from the following NP-complete problem.

Problem 2.3.2. *(2-Disjoint Set Cover Problem) Consider a bipartite graph* $G = (A, B, E)$ *with edges* E *between two disjoint vertex sets* A *and* B*. For* $b \in B$*, define the neighborhood of* b *as* $\mathcal{N}(b) \subseteq A$*. Clearly,* $A = \bigcup_{b \in B} \mathcal{N}(b)$*. Do there exist two disjoint sets* $B_1, B_2 \subset B$ *such that* $|B_1| + |B_2| = |B|$ *and* $A = \bigcup_{b \in B_1} \mathcal{N}(b) = \bigcup_{b \in B_2} \mathcal{N}(b)$*? Let the problem instance be denoted by* $2\mathrm{DSC}(G)$. ◊

It is known that the 2-disjoint set cover problem is NP-complete [30]. We show in the following lemma that given a unit time oracle for the helper decision problem, we can solve the 2-disjoint set cover problem in polynomial time (a polynomial time reduction is denoted by $\leq_L$).

Theorem 2.3.1. *2-Disjoint Set Cover Problem $\leq_L$ Helper Decision Problem.*

2.3.2 Computationally efficient approximations

In this section, we show that Problem 2.2 can be formulated as the maximization of a submodular function subject to matroid constraints. This structure can be exploited to devise computationally efficient algorithms for Problem 2.2 with provable approximation gaps. The definitions of matroids and submodular functions can be found in [31]. First, we define the following ground set:

$$S = \{s_1^1, s_2^1, \ldots, s_F^1, \ldots, s_1^H, s_2^H, \ldots, s_F^H\}, \tag{2.5}$$

where s_f^h is an abstract element denoting the placement of file f into the cache of helper h. The ground set can be partitioned into H disjoint subsets, $S_1, \ldots, S_H$, where $S_h = \{s_1^h, s_2^h, \ldots, s_F^h\}$ is the set of all files that might be placed in the cache of helper h. In (2.2), a cache placement is expressed by the adjacency matrix $\mathbf{X}$. We define the corresponding cache placement set $X \subseteq S$ such that $s_f^h \in X$ if and only if $x_{f,h} = 1$. Notice that the set $\{x_{f,h} : f \in \mathcal{F}\}$ can be considered as the Boolean representation of $X_h = X \cap S_h$, in the sense that $x_{f,h} = 1$ if $s_f^h \in X_h$ and $x_{f,h} = 0$ otherwise.

Lemma 2.3.1. *The constraints in (2.2) on X can be written as a partition matroid on the ground set S defined in (2.5).*

Lemma 2.3.2. *The objective function in Problem (2.2) is a monotone submodular function with respect to the cache placement set X.*

2.3.3 Greedy Algorithm

A common way to maximize a monotonically non-decreasing submodular function subject to a matroid constraint consists of a greedy algorithm that starts with an empty set and at each step it adds one element with the highest marginal value to the set while maintaining the feasibility of the solution. Since the objective function is submodular, the marginal value of elements decreases as we add more elements to the placement set X. Thus, if at one iteration, the largest marginal value is zero, then the algorithm should stop. For our case, the running time would be $O(F^2H^2U)$. Classical results on approximation of such maximization problems [32] establish that the greedy algorithm achieves an objective function value within a factor $\frac{1}{2}$ of the optimum.

2.3.4 A better algorithm with high computational complexity

For maximization of a general monotone submodular function subject to matroid constraints, a randomized algorithm which gives a $(1-1/e)$-approximation has been proposed in [33].

Although this algorithm gives a better performance guarantee than greedy placement, when $|S| = HF$ becomes large, its complexity is still too computationally demanding for implementation. Specifically, the running time of the algorithm in [33] is $O(n^8)$ where n is rank of the matroid. In our formulation, the rank of the matroid is MH. Hence, the time complexity is $O(MH)^8$. When M is a constant fraction of F, the time complexity is $O((HF)^8)$.

2.3.5 A Computationally efficient algorithm with $1-(1-1/d)^d$ approximation ratio

In the special case when $\omega_{h,u} = \omega_1$ (fixed value independent of (h,u)) for all $(h,u) \in \mathcal{E}, h \neq 0$, we provide a different approximation algorithm that runs in $O\left((U+H)^{3.5}F^{3.5}\right)$ time and provides an approximation ratio of $1-(1-1/d)^d$, where $d = \max_u\{|\mathcal{H}(u)|-1\}$ is the maximum number of helpers a user is connected to in $\mathcal{G}$ (excluding the BS). When no bounds on d can be established, this algorithm recovers the ratio of $1-1/e$. We mention that although the worst case time complexity guarantee is $O((U+H)^{3.5}F^{3.5})$, the algorithm involves solving a linear program (LP) with $O((U+H)F)$ variables and a simple deterministic rounding algorithm. The worst cast time complexity of the algorithm is dominated by the time complexity of interior point methods that could be used for the LP involved.

2.3.6 Main result: Algorithm with the Improved Approximation guarantee

In this section, we provide an improved approximation algorithm for the uncoded caching problem in the special case where $\omega_{h,u} = \omega_1$ for all $(h,u) \in \mathcal{E}$ with $h \neq 0$, and $\omega_1 < \omega_{0,u}$ for all $u \in \mathcal{U}$. Recall that, in this case, the optimization problem is given by (2.3). For convenience, we define $g_{f,u}(\mathbf{X}) = \left[1 - \prod_{h\in\mathcal{H}(u):h\neq 0}(1-x_{f,h})\right]$ and write the objective function in (2.3) as

$$g(\mathbf{X}) = \sum_{u,f} P_f \, \widetilde{\omega}_u \, g_{f,u}(\mathbf{X}), \tag{2.6}$$

The program (2.3) fits the general framework of maximizing a function subject to integral assignment constraints involving assignment variables ($\mathbf{X}$ in our case) corresponding to the edges of a bi-partite graph, ($\widetilde{\mathcal{G}}$ in our case). This general framework has been studied in [34]. The authors of [34] provide sufficient conditions under which the optimum of a relaxation of a suitable problem *related* to (2.3) can be rounded using the technique of *pipage rounding* to achieve a constant approximation guarantee for problem (2.3) . In [34], this was carried out for the maximum coverage problem (or max k-cover problem), i.e., the problem of choosing k sets out of a fixed collection of m subsets of some ground set in order to maximize the number of covered elements of the ground set. Our uncoded caching problem, is similar in structure to the maximum coverage problem, although we are not aware of any reduction between the two problems. However, this structural similarity allows us to apply the tools developed in [34] in order to obtain a constant factor approximation to problem (2.3). The result hinges on the machinery developed in [34]. For the sake of clarity, we describe a part of that machinery.

2.3.6.1 Background on Pipage Rounding

We first describe the pipage rounding technique for the following template problem on the bi-partite graph $G = (A, B, E)$, where the matrix $\mathbf{R} \in \mathbb{R}_+^{|A|\times|B|}$ contains the optimization variables $\rho_{a,b}$ for all edges $(a, b) \in E$ (these variables are fixed to 0 for the elements $(a, b) \notin E$).

$$\text{maximize} \quad \phi(\mathbf{R}) \tag{2.7}$$

$$\text{subject to} \quad \sum_{a:(a,b)\in E} \rho_{a,b} \le p(b), \quad \forall \;\; b, \tag{2.8}$$

$$\text{subject to} \quad \sum_{b:(a,b)\in E} \rho_{a,b} \le p(a), \quad \forall \;\; a, \tag{2.9}$$

$$\mathbf{R} \in [0,1]^{|A| \times |B|} \tag{2.10}$$

Here, $p(y) \in \mathbb{Z}_+, \forall y$. Observe that it is the relaxed version of an integer program where the objective function is $\phi(\mathbf{X})$, variables $\rho_{a,b}$ are replaced by $x_{a,b} \in \{0,1\}$ and $\mathbf{X} \in \{0,1\}^{|A| \times |B|}$. The pipage rounding algorithm takes as input $\phi, G, \{p(y)\}$ and a real feasible solution $\mathbf{R}$ and outputs a feasible integral solution $\bar{\mathbf{X}}$. For the sake of completeness, this procedure is provided as Algorithm 7 in the Appendix.

In our case, it is easy to particularize the general template program (2.7) – (2.10) to the program at hand (2.3), by letting $\phi(\cdot) = g(\cdot)$, defined in (2.6), by identifying the graph G with the complete bipartite graph $\mathcal{K}_{\mathcal{F},\mathcal{H}}$ formed by the vertices $\mathcal{F}$, $\mathcal{H}$ and all possible edges connecting the elements of $\mathcal{F}$ (files) with the elements of $\mathcal{H}$ (helpers), and the edge node constraints as $p(h) = M$ for all $h \in \mathcal{H}$ and $p(f) = H$ for all $f \in \mathcal{F}$.

2.3.6.2 Main algorithm and result

The main algorithm involves optimizing a different objective function subject to the constraints relaxed in (2.3) which is identical to the template program (2.7) – (2.10) but rounding with respect to the original function g in (2.3). Now, we state the main theorem.

Theorem 2.3.2. *Let* $\mathbf{R} = \{\rho_{f,h}\}, L = \sum_{f,u} P_f \widetilde{\omega}_u L_{f,u}(\mathbf{R})$ *and* $L_{f,u}(\mathbf{R}) = \min\{1, \sum_{h \in \mathcal{H}(u): h \neq 0} \rho_{f,h}\}$. *Consider* $\mathbf{R}_{opt}$ *to be the optimal solution obtained by maximizing* $L = \sum_{f,u} P_f \widetilde{\omega}_u L_{f,u}(\mathbf{R})$ *subject to the constraints in program (2.3) where* $x_{f,h}$ *is replaced by relaxed variables* $\rho_{f,h} \in [0,1]$ *as follows:*

$$\begin{aligned}
\textit{maximize} \quad & \sum_{f=1}^{F} P_f \sum_{u=1}^{U} \widetilde{\omega}_u \min\{1, \sum_{h\in\mathcal{H}(u):h\neq 0} \rho_{f,h}\} \\
\textit{subject to} \quad & \sum_{f=1}^{F} \rho_{f,h} \leq M, \quad \forall\ h, \\
& \sum_{h=1}^{H} \rho_{f,h} \leq H, \quad \forall\ f \\
& \mathbf{R} \in [0,1]^{F\times H} && (2.11)
\end{aligned}$$

Let $\mathbf{X}_{int}$ be the solution obtained by running Pipage Rounding($\mathcal{K}_{\mathcal{F},\mathcal{H}}, \phi = g, \{p(f) = H, p(h) = M\}, \mathbf{R}_{opt}$). Then, $g(\mathbf{X}_{int}) \geq \left(1-(1-1/d)^d\right) g(\mathbf{X}_{opt})$ where $\mathbf{X}_{opt}$ is the optimum to problem (2.3) and $d = \max_u\{|\mathcal{H}(u)| - 1\}$.

We note that the terms $\min\{1, \sum \rho_{f,h}\}$ in $L(\cdot)$ can be replaced by variables $t_{f,u}$ with additional constraints $t_{f,u} \leq 1$ and $t_{f,u} \leq \sum_{h\in\mathcal{H}(u):h\neq 0} \rho_{f,h}$, in order to turn (A.6) into a linear program (LP) with $(U + H)F$ variables and constraints. Therefore, the algorithm (including pipage rounding) runs with time complexity $O((U + H)^{3.5}F^{3.5})$. We note that the running complexity of the LP dominates that of the rounding step.

We note two important features of this improved approximation ratio compared to the generic scheme [35] for submodular monotone functions that gives $(1 - 1/e)$ approximation ratio. First, the generic algorithm runs in time $O(n^8)$ where n is the rank of the matroid. As argued before, in our case, this is typically too complex. Hence, the improved approximation algorithm is faster by orders of magnitude compared to the generic one. Second, in typical practical wireless networks scenarios any user is connected to only a few helpers (e.g., 3 or 4). The reason is due to spacing between helpers needed to handle interference issues. For example, for the case when every user is connected to at most 4 helpers, the approximation ratio is $1 - (3/4)^4 \approx 0.6836$ while $1 - 1/e \approx 0.6321$. Without any constraints on d, our result recovers the $1 - 1/e$ guarantee of the generic algorithm.

2.4 Conclusion

In this chapter, we focused on the content placement problem in a wireless network formed by helper nodes and wireless users, placing requests to files in a finite library according to a known file popularity distribution. We formulated the problem as the minimization of the total expected downloading delay for a given popularity distribution and network topology, reflected by the connectivity graph and by the link average rates. This results in a coverage problem. We showed intractability and developed efficient approximation algorithms for this. For future work, we would like to point out that this new coverage problem adds to traditional ones like set cover and maximum coverage. It would be actually very interesting to find better approximation algorithms in terms of running time.

Chapter 3

Index Coding: Separation between Combinatorial and Algebraic Schemes

3.1 Introduction

In this chapter [1], we consider an important problem in information theory that addresses traffic over the bottlenecked broadcast link issue in networks with caches, similar to the previous chapter. The problem is called *index coding* and it is a noiseless broadcast problem where m messages need to be sent to n users each requesting one of the m messages through a broadcasting agent. In addition, every user has some *side information packets* (cached in a local cache which they can access) which is a subset of the m messages not including the request. Over the noiseless channel, the transmissions involve *coding* (e.g. linearly combining packets over a specific field) across all the messages. The objective is to characterize the *index coding capacity*. This refers to the minimum number of (coded) transmissions required to satisfy all users when they use side-information to decode. This problem was first formulated in [36], where the motivation was to reduce traffic

[1] The material presented in this chapter is based on the results from two conference papers: 1) K. Shanmugam, A.G. Dimakis and M. Langberg, "Local Graph Coloring and Index Coding". *International Symposium on Information Theory (ISIT 2013)*, Istanbul, 2013. and 2) K. Shanmugam, A.G. Dimakis and M. Langberg, "Graph Theory versus Minimum Rank for Index Coding". *International Symposium on Information Theory (ISIT 2014)*, HI, 2014. The dissertation author is the primary contributor to the above papers and the results in this chapter.

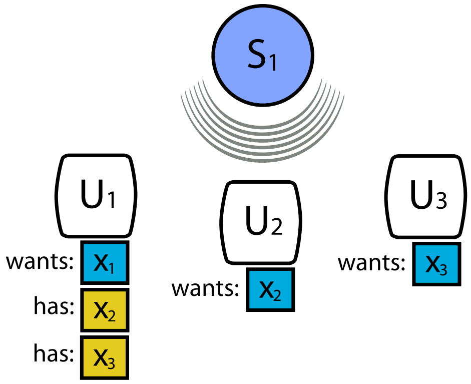


Figure 3.1: Index coding example: We have three users U_1, U_2, U_3 and a broadcasting base station S_1. Each user has some side information packets and requests a distinct packet from S_1. The base station S_1 knows everything and can simultaneously broadcast to all three users noiselessly. User U_i requests packet x_i. User U_1 has packets x_2 and x_3 as side information while users U_2 and U_3 have no side information. In this example three transmissions are required.

in a wireless broadcast link serving multiple terminal each of which have cached content. Note that, in the previous chapter, the benefit of a cache in a similar system was *cache hits*. However, in the current problem given an arbitrary cache state as an input and much simpler topology (a users sees only its own local cache) and requests that do *not have* any cache hits, the problem is to still save on the number of transmissions through coding. An example of the index coding problem and its setting is given in Fig. 3.1

3.1.1 Importance of Index Coding

Index coding is a fundamental network information theory problem with deep connections with combinatorial optimization and graph theory [36–41]. Interest in index coding is further increasing due to the following reasons: The first is that it was recently shown [42, 43] that any arbitrary *network coding* problem where information needs to be sent from a set of sources to their corresponding destinations through a network can be mapped to a properly constructed index coding instance. Therefore, statements about index coding can be translated to constructions or bounds for general networks, showing the surprising expressiveness of the problem. Second, it captures the effect of caching in a network with bottlenecked link serving users with caches through coding other than the conventional *cache hit* approach. Third, *interference alignment*, which is a popular technique for coding for interference channels, alongside information theoretic approaches have been recently applied for index coding [40, 41, 44–46] introducing new interesting techniques for code constructions.

3.1.2 Representation

In the special but important case when $m = n$ and user requests do not overlap, the problem can be represented in terms of a directed side information graph G_d. A directed edge (i, j) means that user i has packet requested by user j. Depending on the structure of the side information graph, index coding can be investigated for undirected (*i.e.* symmetric side information) or, more generally directed graphs. A side information graph representation of the index coding problem of Fig. 3.1 is given in Fig. 3.2. Further, few other representations of the graph like the interference graph is also presented.

In even greater generality, if we allow multiple users to request the same

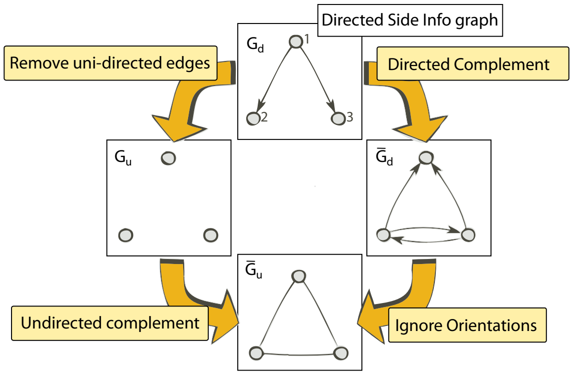


Figure 3.2: Index coding representation using the directed side information graph G_d. There are two alternate ways to reach $\bar{G}_u$. One through G_u, the underlying undirected side information graph. The other way is through $\bar{G}_d$, the interference graph.

packet ($m \neq n$) we can describe the problem with a hypergraph or with a bipartite directed graph [39, 45, 47]. We refer to directed graph problems as unicast index coding (UIC) and the one on more general hypergraphs as groupcast index coding (GIC). To keep the exposition simple, we explain most of our results using UIC and only state the results for the more general case of GIC.

3.1.3 Review of known Coding Techniques

Methods for constructing index codes (*i.e.* upper bounds for index coding) can be broadly separated in two categories: graph theoretic methods and algebraic methods relying on rank minimization. The focus of this chapter is on the former. Graph theoretic methods start from the well-known fact that all the users forming a clique in the side information digraph can be simultaneously satisfied by transmitting the XOR of their packets [36]. This is because, in a clique, every user already has other users' demands as side information except its own. This idea shows that the number of cliques required to cover all the vertices of the graph (the clique cover number) is an achievable upper bound. It is easy to see that the chromatic number of the complement graph is equal to the clique cover number. This is because all the vertices assigned to the same color cannot share an edge and hence must form a clique in the complement graph. It turns out that the idea based on coloring leads to a stronger bound, starting with an LP relaxation called fractional chromatic number [39]. Instead of covering with cliques, one can cover the vertices with cycles and obtain cycle cover bounds [37]. Another achievable scheme called *partition multicast* was proposed [47][36] which generalized both cycle and clique covers. In partition multicast, one first partitions the graph into subgraphs which have *clique-like* connectivity, each corresponding to sub-problems of the given index coding problem, and solves each of them separately.

The second family of bounds is algebraic and requires minimizing the rank over all matrices that respect the structure of the side information graph over a finite field. It turns out [37] that (for a given field size), scalar linear index coding is equal to *minrank*, a quantity introduced by Haemers [48] in 1978 to obtain a bound for the Shannon graph capacity [49]. Therefore, minrank characterizes the best possible scalar linear index code for a given finite field. Throughout this paper, we refer to the former family of bounds as graph-theoretic and the latter as algebraic.

3.1.4 Problem

The main question we investigate in this paper is: How do quantities in the first family of combinatorial bounds for Index Coding compare with each other (i.e. coloring versus partition multicast)? We introduce a new graph theoretic bound based on what is called the *local chromatic number* (or local coloring) of the interference graph which combines interference alignment with the combinatorial concept of coloring. We show that it provably outperforms coloring based bounds. Further, we generalize the above bound by combining all previous graph theoretic ideas: local coloring and partition multicast into another bound that outperforms both coloring and partition multicast. We then prove a rather strong negative result: *all these graph theoretic bounds (local coloring and partition multicast) are within a constant (a universal constant) factor from the bound based on fractional chromatic number.* Previous work has established that the fractional chromatic number is within a $\log n$ factor from the coloring number [50]. Therefore, all these graph bounds can improve, at most, a $\log n$ factor from the bound based on the chromatic number. This is in striking contrast to minrank where prior work has shown [39, 51] that it can outperform the chromatic number by a polynomial factor. We emphasize that this performance benefit of minrank is shown only for special graph constructions [51] and there are other examples where the fractional chromatic number can outperform minrank.

We outline our contributions as follows:

3.1.5 Our Contributions:

1. In 1986, Erdős *et al.* [52] defined the *local chromatic number* of a graph which is smaller than the chromatic number of a graph. The local chromatic number $\chi_\ell(G)$ is defined as the maximum number of different colors that appear in the closed neighborhood of any vertex, minimized over all proper colorings. Here, a closed neighborhood of vertex v includes v and all its neighboring vertices. First, we show that the directed version of local chromatic number, introduced in [53], is an upper bound on unicast index coding. We generalize this to the more general Groupcast setting by defining new parameters called *local hyperclique cover* and its fractional version and its corresponding coding schemes. These are the group cast analogues to local and fractional local chromatic numbers. We show that these have index coding achievable schemes. Further, we show that these parameters are within a factor of $\frac{5}{4}e^2$ (this bound has been improved to e in a parallel and independent work [54] for the UIC case) away from the fractional hyperclique cover. This is the natural generalization of the fractional chromatic number for the groupcast case.

2. We define another parameter, called *partitioned local hyperclique cover* and its fractional version for the groupcast setting. We show that this scheme is stronger than the ones based on local hyperclique cover and partition multicast and therefore all graph-theoretic bounds discussed before. This parameter combines the ideas behind local coloring and partition multicast to provide a better index coding scheme.

3. Finally, we show that all the schemes discussed above, including partition multicast, are within *a constant factor* from the fractional hyperclique cover. The key technical tool used to establish the constant gap results relies on graph homomorphisms [54–56].

3.1.6 Discussion of Techniques: Local chromatic number and its relation to Index Coding

Linear index coding problem on an interference graph $\bar{G}_d$ (as in Fig. 3.2) can be mapped into a *vector assignment* problem. For ease of exposition, we describe the case of scalar linear index coding. The goal is to design n vectors $v_1, v_2, \dots v_n$ that satisfy a set of linear independencies. Specifically, for each vector v_i, one is given a set of indices $S(i)$ and a requirement that

$$v_i \notin \text{span}(v_{S(i)}),$$

where we indicate the set of vectors having indices in $S(i)$ by $v_{S(i)}$. From the interference alignment perspective [44], $S(i)$ are just the set of indices each corresponding to a packet that the user i does not have as side information. We call it the *interfering set* of indices (users) $S(i)$ for user i. $S(i)$ form the out-neighborhood in the interference graph $\bar{G}_d$ which is the directed complement of the side information graph G_d.

The goal of scalar linear index coding is to minimize the dimension k of these vectors while maintaining the required linear independencies. Now, we look at a related problem, i.e. a coloring problem. A valid *proper coloring* of the indices is an assignment of colors to indices such that i and $S(i)$ are assigned different colors. It is possible to obtain a vector assignment from a k coloring solution by simply assigning vectors from the canonical basis of length k to each different color. When canonical basis vectors are used, the resulting assignment of vectors is called a *coloring assignment*.

We use coloring of indices as an *alignment guide* to later assign vectors to colors. Clearly, in any coloring intended for an assignment later (not necessarily a coloring assignment), i and $S(i)$ have different colors since they must be assigned different vectors. However, it is possible to reduce the dimension in which the vectors lie by assigning a different set of vectors. We used MDS codes for this purpose. An (p, q) Maximum-Distance Separable (MDS) code is a set of p vectors of length q that are in general position, *i.e.* any q of the p are linearly independent. The idea is to first color the indices and then create an MDS code and assign one MDS vector for each color.

The key idea to go beyond the chromatic number is to realize that the bottleneck to the dimension of the vectors is not really the total number of colors used but the maximum number of colors in an interfering set over all interfering sets $S(i)$. Therefore, if we can assign colors to the vertices so that the number of colors in the most colorful interfering set, i.e., the *local chromatic number*, is bounded by k^*, we can construct an index code of that length: first create an $(p, k^* + 1)$ MDS code over a sufficiently large field where p is the total number of colors used and assign a vector to each color. Although each of the n indices get a vector assignment, they lie in $k^* + 1$ dimensions. We, therefore, still use the proper coloring of indices as an alignment guide but *the local chromatic number is the metric that lim-*

its the code length or equivalently the dimension of the vectors in the assignment.

3.2 Definitions and review of existing parameters

In this section, we will first provide the definitions for the general GIC problem (group cast problem) and mention how the definitions change for the special UIC case.

Notation: For ease of notation, let $[n]$ denote the set $\{1, 2 \ldots n\}$. $A - B$ is the set difference between sets A and B. Let $G_d(V, E_d)$ be a directed graph on n vertices. If $u \in V$, let $N(u)$ denote the directed out-neighborhood, i.e. $N(u) = \{v \in V : (u, v) \in E_d\}$. Let $(N(u))^c = V - N(u) - u$. Let $\bar{G}_d \left(V, \bar{E}_d\right)$ denote the *directed complement* of G_d which is another directed graph where out-neighborhood of vertex u is $(N(u))^c$. Let 2^A be the power set of A. We define a groupcast index coding problem input instance using a directed bipartite graph as follows.

3.2.1 Formal Definition of Index Coding

Definition 3.2.1. *A Groupcast Index Coding problem (GIC) instance is given by the set $\{U, P, \mathcal{H}(U, P, L)\}$. $U = \{1, 2, 3 \ldots n\}$ is the set of users with $|U| = n$, $P = \{x_1, x_2 \ldots x_m\}$ is the set of packets with $|P| = m$, $n \geq m$. $\mathcal{H}$ is a directed bipartite graph between the sets U and P with L as the set of directed edges. Each packet $x_i \in \Sigma$ where Σ is some alphabet. Every user u requests a single packet $R(u) \in P$ and it has $S(u) \subset P - R(u)$ as side information. If the request of user u is $R(u) = p$, then the directed edge $(u, p) \in L$. If $p \in S(u)$, then the directed edge $(p, u) \in L$.* ◊

We assume w.l.o.g. that for all u, $|R(u)| = 1$. Let $(S(u))^c = P - S(u) - u$. Let $W(p)$ denote the set of all users who want packet p. $R(A) = \bigcup\limits_{u \in A} R(u)$ and $W(P) = \bigcup\limits_{p \in P} W(p)$. Note that a packet can be requested by multiple users.

Now, we define a valid index code for the GIC problem as follows:

Definition 3.2.2. *(Valid index code) Here, for notational reasons, assume $R(u) = x_i \in \Sigma$ is the packet desired by user i. A valid index code over the alphabet Σ is a set $(\phi, \{\gamma_i\})$ consisting of:*

1. *An encoding function $\phi : \Sigma^m \to \{0,1\}^p$ which maps the m packets to a transmitted message of length p bits for some integral p.*

2. *n decoding functions γ_i such that for every user i, $\gamma_u(\phi\,(x_1, x_2 \ldots x_m)\,, S(i)) = x_i$ for all $[x_1\ x_2 \ldots x_n] \in \Sigma^m$. In other words, every user would be able to decode its desired message from the transmitted message and the side information.* ◊

The *broadcast rate* $\beta_\Sigma(\mathcal{H}, \phi, \{\gamma_i\})$ of the $(\phi, \{\gamma_i\})$ index code for the GIC on $\mathcal{H}$ is the number of transmitted bits per received message bit at every user, i.e. $\beta_\Sigma(G_d, \phi, \{\gamma_i\}) = \frac{p}{\log_2 |\Sigma|}$.

Definition 3.2.3. *(Minimum broadcast rate) The minimum broadcast rate $\beta(\mathcal{H})$ is the minimum possible broadcast rate of all valid index codes over all alphabets Σ, i.e. $\beta(\mathcal{H}) = \inf\limits_{\Sigma} \inf\limits_{\phi, \{\gamma_i\}} \beta_\Sigma(\mathcal{H}, \phi, \{\gamma_i\})$.* ◊

In the unicast index coding problem (UIC), where $m = n$ and $P = U$ (packets and users are indistinguishable since user i requests packet x_i). Therefore, one can represent a UIC problem using a directed side information graph G_d with vertex set U where the out-neighborhood of user u is $N(u) = S(u)$.

Definition 3.2.4. *(Interference graph) The interference graph, denoted by $\bar{G}_d(V, \bar{E}_d)$ of an UIC problem is a directed complement $\bar{G}_d$ of the side information graph G_d.* ◊

We now present a number of previously studied upper bounds on $\beta(\mathcal{H})$ for GIC. The first is a bound from [39], referred to as the *fractional hyperclique cover* and denoted here by $\psi_f(\mathcal{H})$. Our definition below slightly differs from that in [39] but nevertheless is equivalent.

3.2.2 Graph theoretic bound based on (hyper)Clique cover or Coloring number

Definition 3.2.5. *(Weak Hyperclique) A weak hyper clique $C \subseteq U$ is such that for any pair $u, v \in C$, we have $(u \in S(v)$ AND $v \in S(u))$ OR $R(u) = R(v)$.* ◊

Observe that in the GIC problem, one can satisfy all the users in C by XORing their requests $R(C)$. This implies that a "cover" of the hypergraph by weak hypercliques implies a corresponding valid index code. In the rest of the paper, we use the term "hyperclique" instead of "weak hyperclique".

Definition 3.2.6. *The hyperclique cover of $\mathcal{H}$, denoted by $\psi(\mathcal{H})$, is given by the following Integer Program:*

$$\min \sum_{C \in \mathcal{C}} y_C$$
$$\text{s.t.} \sum_{C: u \in C} y_C = 1, \ \forall u \in U$$

$$y_C \in \{0,1\},\ \forall C \in \mathcal{C} \tag{3.1}$$

where $\mathcal{C}$ is the set of all hypercliques in $\mathcal{H}$. ◊

The LP relaxation of (3.1) is the fractional hyperclique cover $\psi_f(\mathcal{H})$. A feasible solution to (3.1) is a set of chosen hypercliques such that every user is covered exactly by one hyperclique. This implies that: $\beta \leq \psi(\mathcal{H})$ by our discussion above.

In the UIC problem, a hyperclique is equivalent to a clique on G_d (a clique in a directed graph is a complete subgraph where there are edges in both directions between any two vertices). Therefore, the fractional chromatic number, defined on the directed complement $\bar{G}_d$, is the equivalent of ψ_f. It is denoted by $\chi_f\left(\bar{G}_d\right)$. The chromatic number defined on $\bar{G}_d$ is denoted $\chi(\bar{G}_d)$.

3.2.3 Graph Theoretic bound based on Partition Multicast

We now turn to discuss an additional scheme for GIC, partition multicast, introduced in [47]. The scheme is a generalization of both cycle cover and hyperclique cover. Formal definition is given below:

Definition 3.2.7. *The partition multicast number of $\mathcal{H}$, denoted $\psi^p(\mathcal{H})$, is given by the following integer program:*

$$\begin{aligned} \min\ & \sum_M a_M d_M \\ \text{s.t.} & \sum_{M: v \in M} a_M = 1,\ \forall u \in U \\ & a_M \in \{0,1\},\ \forall M \in 2^U - \{\emptyset\} \end{aligned} \tag{3.2}$$

where $\mathcal{C}$ is the set of hypercliques in $\mathcal{H}$ and $d_M = |R(M)| - \min_{u \in M} |R(M) \bigcap S(u)|$. ◊

We provide some intuition behind (3.2). A feasible solution chooses a family of subsets of users (based on the value of a_M). We call each subset a multicast group. Every user is covered by exactly one such group. The bipartite subgraph, induced by a multicast group M and packets demanded by M is denoted $\mathcal{H}(M, R(M))$. Every user has at least $\min_{u \in M} |R(M) \bigcap S(u)|$ packets from $R(M)$. It was shown in [47] that d_M coded transmissions using an $(|R(M)|, d_M)$ MDS code allows users in group M to recover their packet. The program (3.2) partitions the user set into an optimum set of multicast groups depending on the cost (d_M) of transmission for each group.

Definition 3.2.8. *The fractional partition multicast number of $\mathcal{H}$, denoted $\psi_f^p(\mathcal{H})$, is given by the LP relaxation of ψ^p.* ◊

As far as we know, the fractional version of ψ^p has not been studied before this work. It is possible to show that $\beta(\mathcal{H}) \leq \psi_f^p \leq \psi_f$ (simple extension to arguments in [47]).

3.3 Definitions for New parameters

In this section, we provide definitions of new parameters that will be shown to have achievable index coding schemes for the GIC problem.

3.3.1 Graph theoretic bound based on Local Chromatic number

We briefly consider two other graph parameters previously studied in [53][57], namely, the *local chromatic number*, denoted by $\chi_\ell(\bar{G}_d)$, and the *fractional local chromatic number*, denoted by $\chi_{f\ell}(\bar{G}_d)$ of a directed graph $\bar{G}_d$ (the parameters are actually defined with respect to the complement graph G_d for them to make sense with respect to Index Coding). We briefly describe χ_ℓ first.

Consider a directed graph $\bar{G}_d$. Let $c : V \to [k]$ be any proper k-coloring for the graph (if two vertices are connected by edges in either direction then they are colored differently) for some integer k. Let $|c(N^+(i))|$ denote the number of colors in the closed out-neighborhood (out-neighborhood of a vertex taken along with the vertex) of the directed graph taking the orientation into account. Then,

$$\chi_\ell(\bar{G}_d) = \min_c \max_{i \in V} |c(N^+(i))|.$$

In words, the local chromatic number of a directed graph $\bar{G}_d$ is the maximum number of colors in any out-neighborhood minimized over all proper colorings of it.

We define the GIC analogues of $\chi_\ell\left(\bar{G}_d\right)$ and its fractional version $\chi_{f\ell}\left(\bar{G}_d\right)$. As far as we are aware, we have not encountered these generalizations for the GIC problem on directed bipartite graphs.

Definition 3.3.1. *The local hyperclique cover of $\mathcal{H}$, denoted $\psi_\ell\left(\mathcal{H}\right)$, is given by the following integer program:*

$$\begin{aligned} &\min \ t \\ \text{s.t.} &\sum_{C: W(R(u) \bigcup (S(u))^c) \bigcap C \neq \emptyset} y_C \leq t, \ \forall u \in U \end{aligned}$$

$$\sum_{C:u\in C} y_C = 1,\ \forall u \in U$$
$$y_C \in \{0,1\}\ \forall C \in \mathcal{C},\ t \in \mathbb{Z}^+ \tag{3.3}$$

where $\mathcal{C}$ is the set of hypercliques in $\mathcal{H}$. ◊

The LP relaxation of (3.3) is defined to be the *fractional local hyperclique cover*, denoted $\psi_{f\ell}(\mathcal{H})$. Note that, the UIC analogues of ψ_ℓ and $\psi_{f\ell}$ are $\chi_\ell\left(\bar{G}_d\right)$ and $\chi_{f\ell}(\bar{G}_d)$ [55] respectively. The definition given before for $\bar{G}_d$ and the above LP definition are equivalent. Now, we provide a brief description about the feasible solution to (3.3). For a user u, let us call the set of users that request packets not in S_u to be the *interference neighborhood*. The interference neighborhood consists of: 1) users requesting the same packet as the user ($R(u)$). 2) users requesting packet neither in S_u nor $R(u)$. For any user u, given the feasible hyperclique cover, we count the number of hypercliques, belonging to the cover, in user u's interference neighborhood. Let us call this *local hyperclique count* of user u. t denotes the maximum local hyperclique counts over all users. Then finally minimizing t over all possible hyperclique covers, gives ψ_ℓ. In this work we will show that ψ_ℓ is an upper bound to β.

3.3.2 Combining Local coloring ideas with partitioning

We define a new achievable scheme for the GIC problem by combining ideas from local hyperclique cover and partition multicast. This new scheme is called *partitioned local hyperclique cover* denoted by ψ_ℓ^p. Now, we briefly discuss the motivation behind defining ψ_ℓ^p.

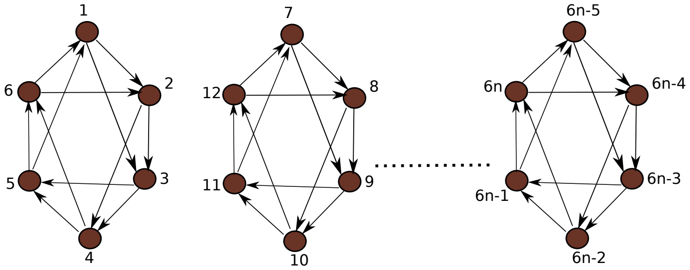


Figure 3.3: An example UIC problem with a side information graph G_d for which $\chi_f\left(\bar{G}_d\right) = 6n$. The partition multicast number and its fractional versions are both $4n$. Partitioning into component 6-vertex graphs and adding up the local chromatic numbers of their complements gives $4n$, i.e. $\chi^p_{f\ell} = 4n$.

For simplicity, let us consider the UIC problem on directed side information graphs. Recall that $\chi_f\left(\bar{G}_d\right)$ is the optimal way of fractionally covering a digraph G_d with cliques. Since, a subset of a clique is a clique, partitioning a graph into different groups and then adding up the clique covers of each group is not going to be better than covering the whole graph with cliques without partitioning. But partitioning the whole graph and calculating local chromatic numbers for each of the partitions and adding them can be beneficial. An example, that illustrates this is given in Fig. 3.3.

Definition 3.3.2. *The partitioned local hyperclique cover number of $\mathcal{H}$, denoted $\psi^p_\ell(\mathcal{H})$, is given by the following integer program:*

$$\begin{aligned} &\min \sum_{M} a_M t_M \\ \text{s.t.} \quad &\sum_{C:W(R(u)\bigcup(S(u))^c)\bigcap C\bigcap M\neq\emptyset} y_C \leq t_M, \ \forall u\in M, \ \forall M\in 2^U \\ &\sum_{M:v\in M} a_M = 1, \forall u\in U \end{aligned}$$

$$\sum_{C:v\in C} y_C = 1, \forall u \in U$$

$$a_M, y_C \in \{0,1\} \ \forall M \in 2^U - \{\emptyset\},\ C \in \mathcal{C},\ t_M \in \mathbb{Z}^+ \tag{3.4}$$

where $\mathcal{C}$ is the set of hypercliques in $\mathcal{H}$ and M is a multicast group. $\Diamond$

The fractional version of $\psi_\ell^p(\mathcal{H})$, denoted by $\psi_{f\ell}^p(\mathcal{H})$ is the LP relaxation of (3.4). Let us denote the UIC analogue of $\psi_{f\ell}^p$ by $\chi_{f\ell}^p\left(\bar{G}_d\right)$ and call it partitioned fractional local chromatic number. In a feasible solution to (3.4), we first partition the set of users into a family of multicast groups. Separately, we cover all users using a hyperclique cover. Over all users in every group M, we get the maximum local hyperclique count t_M, restricting the interference neighborhood of every user to that group. Optimizing the sum of all such counts from different multicast groups over all possible hyperclique covers and multicast group allocations gives ψ_ℓ^p.

3.4 Achievable Schemes

We first show the existence of achievable index coding schemes for parameters based on local chromatic number and its fractional versions. ψ_ℓ and $\psi_{f\ell}$

.

Theorem 3.4.1. *There are achievable linear index codes corresponding to $\psi_\ell(\mathcal{H})$ and $\psi_{f\ell}(\mathcal{H})$ implying $\beta(\mathcal{H}) \le \psi_{f\ell}(\mathcal{H}) \le \psi_\ell(\mathcal{H})$.*

Proof. Here, we provide only a sketch of the proof for the case of a UIC index coding problem on the side information graph G_d. The full proof of the hypergraph case is in the Appendix as it is involved. We show that there is an index coding scheme whose number of transmissions is the local chromatic number of the complement, i.e. $\beta(G_d) \leq \chi_\ell(\bar{G}_d)$.

Let $\mathfrak{I}$ denote the family of independent sets (a set of vertices where there are no edges in any direction between them) in $\bar{G}_d$. Coloring this graph, involves assigning 0's and 1's to the independent sets in the graph. Let $\mathcal{J} \subseteq \mathfrak{I}$ be the set of color classes in the optimal local coloring. Let $\chi_\ell(\bar{G}_d)$ be the local coloring number. Let $J : V \longrightarrow \mathcal{J}$ be the coloring function. To each color class (independent set assigned 1), we assign a column vector from $\mathbb{F}_q^m$ of a suitable length m and over a suitable field $\mathbb{F}_q$ by a map $\mathbf{b} : \mathcal{J} \longrightarrow \mathbb{F}_q^m$. If the message desired by each user is from the finite field $\mathbb{F}_q$, i.e. $x_i \in \mathbb{F}_q$, $\forall i \in V$, then we transmit the vector

$$[\mathbf{b}(J(1)), \mathbf{b}(J(2)) \ldots \mathbf{b}(J(n))] \, [x_1, x_2 \ldots x_n]^T.$$

Clearly the length of the code is m field symbols. If the index code is valid, then the broadcast rate is m.

We now exhibit a mapping $\mathbf{b}$ with $m = \chi_\ell(\bar{G}_d)$ and $q \geq |\mathcal{J}|$. Let the colors classes in $\mathcal{J}$ be ordered in some way. Consider the generator matrix $\mathbf{G}$ of a $(|\mathcal{J}|, \chi_\ell(\bar{G}_d))$ MDS code over a suitable field $\mathbb{F}_q$,where $q \geq |\mathcal{J}|$. For instance, Reed Solomon code constructions could be used. Assign the different columns of $\mathbf{G}$ to each color class, i.e. $\mathbf{b}(j) = \mathbf{G}_j$, $\forall j \in \mathcal{J}$ where $\mathbf{G}_j$ is the j-th column. Under this mapping $\mathbf{b}$ and the previous description of the index code, it remains to be shown that this is a valid code. For any vertex i, the closed out-neighborhood $N^+(i)$ contains $|J\left(N^+(i)\right)|$ colors. Because, the coloring J corresponds to the optimal local coloring, there are at most m colors in any closed out neighborhood. Therefore,

$|J\left(N^+(i)\right)| \le \chi_\ell(\bar{G}_d) = m$.

Every vertex (user) i must be able to decode its own packet x_i. User i possesses packets x_k as side information when k is not in the closed out-neighborhood $N^+(i)$ in the interference graph $\bar{G}_d$. Hence, $\mathbf{b}(J(k))x_k$ can be canceled for all $k \notin N^+(i)$. The only interfering messages for user i are $\{\mathbf{b}(J(k))x_k\}_{k \in N^+(i)-\{i\}}$. If we show that $b(J(i))$ is linearly independent from all $\{\mathbf{b}(J(k))\}_{k \in N^+(i)-\{i\}}$, then user i would be able to decode the message x_i from its *interferers* in $N^+(i) - \{i\}$.

Since $\mathcal{J}$ represents a proper coloring over $\bar{G}_d$, $\mathbf{b}(J(i))$ is different from $\mathbf{b}(J(k))$ for any $k \in N^+(i)-\{i\}$. Also, any m distinct vectors are linearly independent by the MDS property of the generator $\mathbf{G}$. Since, $|J\left(N^+(i)\right)| \le \chi_\ell(\bar{G}_d) = m$, i.e. the number of colors in any closed out-neighborhood is at most χ_ℓ, the distinct vectors in any closed-out neighborhood are linearly independent . This implies that $b(J(i))$ is linearly independent from $\{\mathbf{b}(J(k))\}_{k \in N^+(i)-\{i\}}$. Hence, every user i would be able to decode the message it desires. Hence it is a valid index code and the broadcast rate is $\chi_\ell(\bar{G}_d)$. □

Now, we show that achievable index coding schemes exist for all parameters that are based on partition multicast.

Theorem 3.4.2. *For a GIC on $\mathcal{H}$, there exist achievable index coding schemes whose broadcast rates equal ψ_f^p, ψ_ℓ^p and $\psi_{f\ell}^p$.*

3.5 Relationship between different parameters

In this section, we provide bounds for ratios between different parameters. First, we prove the following result about multiplicative gap between the fractional chromatic and version of the local chromatic number for the UIC problem on a directed side information graph G_d.

Theorem 3.5.1. $\max_{G_d} \frac{\chi_f(\bar{G}_d)}{\chi_\ell(\bar{G}_d)} \leq \max_{G_d} \frac{\chi_f(\bar{G}_d)}{\chi_{f\ell}(\bar{G}_d)} \leq \frac{5}{4}e^2$. *Further, there exists digraphs* G_d *such that* $\frac{\chi_f(\bar{G}_d)}{\chi_\ell(\bar{G}_d)} \geq 2.5244$

Proof. We provide the proof sketch of how to bound χ_f/χ_ℓ. The main idea we use is the following: Suppose for a directed graph $\bar{G}_d$, the local chromatic number $\chi_\ell = k$ and the total number of colors used to get this optimum is m. Then $\bar{G}_d$ is homomorphic to a directed graph $U_d(m,k)$ whose local chromatic number is k. A digraph G is homomorphic to another digraph H if there is a function f mapping vertices of G to the vertices of H such that edges (including their directionality) are preserved after the map, i.e. (u,v) being a directed edge implies that $(f(u), f(v))$ is a directed edge in H. Further, all chromatic number variations (fractional version in particular) only can increase from left to right in a homomorphism. Therefore, the problem reduces to comparing the χ_f and χ_ℓ for this restricted class of graphs $U_d(m,k)$ which has nice symmetry properties like vertex transitivity. We obtain a constant factor upper bound on that after some combinatorial arguments. A similar proof applies for the case of comparing χ_f and $\chi_{f\ell}$ □

Note: A parallel and independent work [54] has shown a tighter upper bound of e. This means that the performance of the achievable scheme due to $\chi_{f\ell}$ is at most e away from the one based on χ_f for the UIC problem.

Now, using the generalizations to the GIC problem, we also show that the same constant factor bound also holds for the general problem.

Theorem 3.5.2. $\frac{\psi_f(\mathcal{H})}{\psi_\ell(\mathcal{H})} \leq \frac{\psi_f(\mathcal{H})}{\psi_{f\ell}(\mathcal{H})} \leq \max_{G_d} \frac{\chi_f(\bar{G}_d)}{\chi_{f\ell}(\bar{G}_d)}$.

Now, we show that $\psi_{f\ell}^p$ is better than all the graph theoretic schemes.

Theorem 3.5.3. *The achievable scheme based on $\psi_{f\ell}^p$ is better than all known previous achievable schemes based on the concepts of hyperclique covers, local graph coloring and partitioning. Formally, $\psi_{f\ell}^p \leq \psi_{f\ell} \leq \psi_f$ and $\psi_{f\ell}^p \leq \psi_f^p$.*

Now, we state the final result of the paper. This implies that rates of all the achievable schemes discussed in this work are at most a constant factor (actually at most e according to the improved result) far away from ψ_f $(\mathcal{H})$.

Theorem 3.5.4. $\psi_f/\psi_{f\ell}^p \leq \max_{G_d} \frac{\chi_f(\bar{G}_d)}{\chi_{f\ell}(\bar{G}_d)}$.

This effectively compares the partition multicast scheme with the ones based on coloring by introducing a few parameters based on the concept of local graph coloring.

3.6 Conclusion

In this chapter, we provided several new index coding schemes based on the local chromatic number for both the unicast and groupcast settings. We showed that one of these schemes is better than all known (prior to this work) graph theoretic schemes and generalizes schemes based on clique covers, cycle covers, partition multicast and local coloring. We show that this scheme is multiplicatively at most a constant far away from the scheme based on fractional chromatic number for both groupcast and unicast settings. An interesting future direction is to find new index coding schemes based on Integer or Linear Programs which are polynomially better than the one based on the fractional chromatic number.

Chapter 4

Network coding multiple unicast: On Approximating the Sum-Rate using Vector Linear Codes

4.1 Introduction

In this chapter [1], we will explore connections between the Index Coding problem, the problem of optimizing traffic in a bottlenecked broadcast link serving users with cached information, and other problems in information theory. There are two problems that are related to this: 1) Network Coding problem and 2) Locally repairable Codes in distributed storage. We first prove a connection between Index Coding and the latter problem. We use this connection to show some results about approximating an edge-cut bound, a generalization of mincut, to information flows on directed networks. Briefly, in the index coding problem, a single broadcasting agent needs to communicate n distinct messages to n receivers (one message per receiver) over a noiseless broadcast channel. A subset of the source messages is available as side-information to each receiver. The objective is to de-

[1] The material in this chapter is based on these two conference papers: 1) K. Shanmugam, M. Asteris and A.G. Dimakis, "On approximating the sum-rate for multiple unicasts". *International Symposium on Information Theory (ISIT 2015)*, HongKong, 2015. and 2) K. Shanmugam and A.G. Dimakis, "Bounding Multiple Unicasts through Index Coding and Locally Repairable Codes". *International Symposium on Information Theory (ISIT 2014)*, HI, 2014. The dissertation author is the primary contributor to the above papers and the results in this chapter.

sign a broadcast scheme that uses minimum number of transmissions to deliver the n messages.

The *multiple-unicasts network coding problem* is one of the fundamental problems in network information theory. In this problem, k source nodes need to communicate independent information to k corresponding destinations through a directed acyclic network. Information is encoded at the sources and flows through links with limited (typically integral) capacity, while intermediate nodes create (possibly non-linear) combinations of the incoming messages. The canonical question is: *what is the set of transmission rates supported by a given network $\mathcal{G}$ with k independent sources?* A related objective is determining *the optimum achievable sum-rate*, *i.e.*, the optimum joint source entropy rate for the k independent sources. The problem has been extensively studied (see, *e.g.*, [58–60] and references therein). It is known that *non-linear* codes are required to achieve the capacity [61], but few papers have studied the question of approximating the rate for multiple unicasts (e.g. [62, 63]).

4.1.0.1 Related Work on flows and cuts

If one considers an information flow problem between a single source and a single destination in a directed network, the maximum flow is exactly equal to the minimum cut, which is the celebrated max-flow min-cut theorem [64]. A generalization of this problem is the maximum multi-commodity flow problem over directed and undirected networks. Here, there are k source-destination pairs and each has a 'commodity' flow associated with it that has to be routed. A given link in the network, can 'share' flows of different commodities respecting capacity constraints. It is known that the capacity of a *multi-cut*, a set of edges that disconnects all source-destination pairs upon removal, is an upper bound on the maximum multi-commodity flow. However, there is a flow-cut gap of at most $O(\log k)$ in the undirected case while in the directed case, there are examples where the gap is $\Omega(k)$ [65]. We consider the problem of independent *information flow* (multiple unicast network coding) over a directed acyclic network between k different sources and their corresponding destinations. We first note that, the multi-cut is not an upper bound on the information flow [66].

A significant body of work has focused on developing upper bounds on the joint source entropy rate for multiple-unicasts with independent sources. Several of these bounds belong to the class of *edge cut* bounds, in which the sum-rate is upper bounded by the cumulative capacity of a appropriately selected set of network links like in the case of commodity flows. *Cut set* bounds are a prominent representative of this family, but they are outperformed by a newer member of this class: the *GNS (Generalized Network Sharing) cut bound* [66] which is a generalization of mincut for information flows. There are several other related bounds including the PdE [67], Information dominance [68] and Functional dependence [69]. With few exceptions (GNS cut and Functional dependence bounds are equivalent), it is not known how these bounds compare. However, all these bounds share one thing in common with the GNS cut: *they are NP-hard to compute*. In this chapter, we show that the GNS cut bound can be approximated efficiently (within polylog k) and the approximation provides a good code for *correlated information flow*. We also show that sum-rate (bits per use of the network) of independent information flow in unicast networks for vector linear codes depend on the field chosen.

4.1.0.2 Connections between Index Coding and Locally repairable codes:

Some of our results on the unicast problem are established using a duality property between linear index coding and Locally Repairable Codes (LRCs). Locally repairable codes were recently developed [70–74] to simplify repair problems for distributed storage systems and are currently used in production [75]. Here, we show that a natural extension that we call Generalized Locally Repairable Codes (GLCRs) are exactly dual to linear index codes. Specifically, in a GLRC, every node stores some information that is decodable from information stored in a specific *recoverability* set of other nodes. These specifications induce a *recoverability directed graph*. We show that the dual linear subspace of a GLRC is a solution to an index coding instance where the side information graph is taken to be the recoverability graph of the GLRC. Then, we view the unicast problem with correlated sources as a special case of the GLRC problem and that is useful in establishing some of our results about the GNS-cut in this work.

4.1.0.3 Our Contributions:

1. We show that vector linear GLRCs are dual to vector linear Index Codes when both of them are defined on the same directed graph.[2].

[2] At the time of submission of the work, we became aware of a concurrent independent work by Mazumdar [76] establishing approximate duality results for non-linear GLRC and Index coding.

2. We tensorize the GNS-cut bound as follows: We use an argument that uses strong graph products to obtain a sequence of rate upper bounds that are valid for vector-linear codes – we show that the weakest bound in this sequence is the GNS cut bound.

3. We define a new communication problem that we call the *relaxed-correlated* multiple-unicasts. In this problem, independence across sources is relaxed: the code designer is allowed to introduce *any* correlation structure in the sources in order to maximize the joint source entropy rate. GNS cut is also an upper bound on the optimum joint source entropy rate for this relaxed-correlated multiple-unicasts problem.

4. We develop a polynomial time algorithm to provably approximate the GNS cut bound from above within an $O(\log^2 k)$ factor, where k is the number of sources in the network. Our algorithm also yields a *vector-linear code* for the relaxed-correlated sources problem achieving joint source entropy rate within an $O(\log^2 k)$ factor from the optimum over all (even non-linear) network codes.

5. One important question is: How does the finite field used by the vector-linear code influence the sum-rate? We show that the choice of the field matters tremendously. For *any two fields* $\mathbb{F}_p$ and $\mathbb{F}_q$ and for any $\delta > 0$, there exist multiple-unicast networks for large k such that the optimal sum-rates over $\mathbb{F}_p$ and $\mathbb{F}_q$ differ by a factor of $k^{1-\delta}$, for (Theorem 4.5.1). Note that a $1/k$-approximation can be achieved by having a single unicast and ignoring all other sources. Our result shows that this kind of separation can almost be caused by a poor choice

of field. This partially negatively answers an open problem stated recently in [63], asking whether vector-linear codes can approximate the network capacity within a logarithmic factor. Our result shows that the answer is negative for the sum-rate over a fixed field. This relies on a similar result for the symmetric-rates ([51, 77]).

This implies: for *any given field* $\mathbb{F}$, there exists a multiple-unicasts network (with sufficiently large k) for which the optimum vector-linear joint entropy rates for independent and correlated sources are separated by a factor of $k^{1-\delta}$, for any constant $\delta > 0$ (Theorem 4.5.1).

Note that our results do not rule out the approximation of the optimum sum-rate for multiple-unicasts by linear codes in general. They do imply, however, that the achievability must use a field that depends on the network.

Remark: Detailed proofs are provided in [78] and in the Appendix. Only brief sketches are provided wherever applicable in this chapter.

4.2 Definitions

We begin with a set of formal definitions that are useful for our subsequent developments. Some definition, although similar to definitions in previous chapters are slightly modified to state results in this chapter.

Definition 4.2.1. (Directed Index Coding) *Consider a set of n independent messages* (symbols) $\mathbf{x}_i \in \mathbb{F}^p, i = 1, \ldots, n$*, each consisting of $p \in \mathbb{N}_+$ packets* (subsymbols) *in some alphabet $\mathbb{F}$, and a set of n users $\{1, \ldots, n\}$, such that user i: 1. wants mes-*

sage x_i, and 2. has messages x_j, $j \in S_i \subseteq \{1, \ldots, n\} \backslash \{i\}$ as side-information. A sender wishes to broadcast all n messages to the corresponding users over a noiseless channel. The objective is to design a coding scheme that minimizes the number of transmissions required for all users to decode their respective messages. ◊

An Index Coding instance is fully characterized by its *side-information graph G*. The side-information graph G is a directed graph on n vertices corresponding to n users. An edge (i, j) exists in G if and only if $j \in S_i$, *i.e.*, user i *has* message $\mathbf{x}_j$ as side-information.

Let $\mathbf{x} = \left[\mathbf{x}_1^T\ \mathbf{x}_2^T \cdots \mathbf{x}_n^T\right]^T$ be the (pn)-dimensional vector formed by stacking the n symbols $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{F}^p$. The sender transmits one symbol (or equivalently p-subsymbols) per *channel use*. An $(\mathbb{F}, p, n, r)$ *vector-linear index code* for this problem consists of r linear combinations of symbols in $\mathbf{x}$ over a field $\mathbb{F}$ that satisfies the decodability criterion at every user.

The *broadcast rate* $\beta_{\mathrm{VL}}^{\mathbb{F}}(G, \mathcal{C})$ of an $(\mathbb{F}, p, n, r)$-vector-linear index code $\mathcal{C}$ is the ratio $^r/_p$; *the number of channel uses* required for all users to receive their message. Note that the broadcast rate is at most n. [3]

Definition 4.2.2. (Generalized Locally Repairable Code) *A (Σ, p, n, k) vector linear generalized locally repairable code (GLRC) of dimension k over field Σ is a k dimensional subspace $\mathcal{C} \subseteq (\Sigma^p)^n$ where every contiguous set of p subsymbols is grouped into one codeword supersymbol. Further, a codeword supersymbol i satisfies*

[3] Recall that a channel use is the transmission of a symbol, or equivalently the transmission of p-subsymbols.

the following recoverability condition: every subsymbol of the ith supersymbol is a linear combination of the subsymbols belonging to a set S_i of codeword supersymbols not containing i. These conditions can also be represented in the form of a directed recoverability graph G where the vertices correspond to the n supersymbols and the directed out-neighborhood of a vertex i is the recoverability set S_i. ◊

The *normalized rate* of the GLRC is given by k/p.The maximum normalized rate over all the linear codes for a given recoverability graph $\bar{G}$ is denoted by $R^{GLRC}\left(\bar{G}\right)$.

Definition 4.2.3. *(***Multiple-Unicasts (MU) Network***) A multiple-unicasts network instance is an acyclic directed network $\mathcal{G}(\mathcal{N}, \mathcal{E})$ on a set $\mathcal{N}$ of nodes, with the following components:*

1. *$\mathcal{E}$ is the set of links (edges) in the network. Links have unit capacity; they carry at most one bit per channel use. We use $c_{(a,b)}$ to denote the total capacity from node a to node b, i.e., the number of links from a to b. Finally, $h(e)$ and $t(e)$ denote the head and tail of an edge $e \in \mathcal{E}$, respectively. For an edge e from node a to node b, the tail node is a while the head node is b.*

2. *(*Source/Destination nodes*) $\mathcal{S} \triangleq \{s_1, s_2, \ldots, s_k\} \subseteq \mathcal{N}$ is a set of k source nodes, and $\mathcal{T} \triangleq \{t_1, t_2, \ldots, t_k\} \subseteq \mathcal{N}$ is a set of k destination nodes.*

3. *(*Source links*) $\mathcal{E}_i \subset \mathcal{E}$ is a set of* $\text{mincut}(s_i, t_i)$ *edges each with* no *tail node and with the head node $h(e) = s_i$, $\forall e \in \mathcal{E}_i$, $i = 1, \ldots, k$. Here,* $\text{mincut}(s_i, t_i)$ *is the number of unit-capacity links in the minimum cut between source s_i and destination t_i. We refer to $\mathcal{E}_i$ as the set of* source links *of source s_i.*

Each source node s_i wants to transmit information to its corresponding destination t_i, $i = 1, \ldots, k$. Information is fed into the network through the source links $\cup_{i=1}^{k} \mathcal{E}_i$. $\Diamond$

The *multiple-unicasts network coding problem* is the problem of designing a *network code*: the set of rules that govern how information is encoded and flows through the network. One of the canonical objectives of *multiple-unicasts network coding* is to maximize the total amount of information transmitted through the network per channel use, *i.e.*, to maximize the *joint source entropy rate.* Here, we focus only on *vector-linear codes, i.e.*, codes in which encoding and decoding involve only vector-linear operations.

Definition 4.2.4. (Vector Linear MU Network Code) *An $(\mathbb{F}, p, m, r)$ vector-linear MU network code $\mathcal{C}$ is a collection of vectors $\mathbf{z}_e \in \mathbb{F}^p, \forall e \in \mathcal{E}$ that depend on the aggregate source message vector $\mathbf{x} \in \mathbb{F}^r$ (consisting of r independent subsymbols) satisfying:*

1. Coding at intermediate nodes: *For* source link *e, each component of $\mathbf{z}_e$ is a linear combination of sub-symbols in $\mathbf{x}$. For each* non-source *link $e \in \mathcal{E}$, each component of $\mathbf{z}_e$ is a linear combination of entries of $\mathbf{z}_a$'s of the edges incident on it, i.e. $\{\mathbf{z}_a\}_{a:h(a)=t(e)}$.*

2. Decoding at destinations: *At every destination t_i, every variable $\mathbf{z}_e$ for $e \in \mathcal{E}_i$, is linearly decodable from information flowing into t_i, i.e. $\{\mathbf{z}_a\}_{a:t(a)=t_i}$.*

3. Independence between sources: *The variables of one source, i.e. $\{\mathbf{z}_e\}_{e \in \mathcal{E}_i}$ are mutually independent of those of other sources.*

The joint source entropy rate achieved by such a code is equal to $^{r}/_{p}$ bits per channel use. Due to the independence among sources, the joint source entropy rate is equal to the *sum-rate* of the k sources. We use $R^{\text{MU}}(\mathcal{G};\mathbb{F})$ to denote the *optimum sum-rate achievable over all vector-linear network codes defined over the field* $\mathbb{F}$, and $R^{\text{MU}}(\mathcal{G})$ to denote the optimum vector-linear sum-rate over all fields.

Relaxed-Correlated Sources. Consider a variant of the multiple-unicasts network coding problem, in which the requirement that source information is independent across sources is overlooked. We refer to the modified version as the problem of *relaxed-correlated sources.* In the modified problem, we still seek to maximize the joint source entropy, but the code designer is allowed to pick arbitrary correlations among sources.

Definition 4.2.5. (**Vector-Linear Relaxed-Correlated MU Network Code**) *A vector-linear code* $\mathcal{C}$ *for the multiple-unicasts network coding problem with* relaxed-correlated sources *is defined as in Def. 4.2.4 omitting requirement* (3). ◊

We use $R^{\text{CO}}(\mathcal{G};\mathbb{F}) = r/p$ to denote the optimum joint source entropy rate achievable by vector-linear codes over a given field $\mathbb{F}$ in the *relaxed-correlated sources* problem, and $R^{\text{CO}}(\mathcal{G})$ to denote the optimum rate over all fields, accordingly. Clearly, $R^{\text{MU}}(\mathcal{G}) \leq R^{\text{CO}}(\mathcal{G})$.

Remark 4.2.1. *In Def. 4.2.3, we require* $|\mathcal{E}_i| = \text{MINCUT}(s_i, t_i)$*. This is only a useful convention and does not affect the value of* $R^{\text{MU}}(\mathcal{G})$*. It does, however, affect* $R^{\text{CO}}(\mathcal{G})$*. In this work, we upper bound* $R^{\text{MU}}(\mathcal{G})$ *by developing bounds on* $R^{\text{CO}}(\mathcal{G})$*. Hence, the convention becomes essential.*

4.3 New Bounds on the Vector Linear sum-rate of an MU Network

4.3.1 Duality between Index Coding and GLRC

We first show that vector linear codes for Index Coding on a side-information graph G is dual to a vector linear GLRC when G is taken to be the recoverability graph.

Theorem 4.3.1. *Let $\mathcal{C}$ be a linear subspace (or a subspace) of dimension k. Let the dual code (or the dual subspace) of $\mathcal{C}$ of dimension $np - k$ be denoted by $\mathcal{C}^{\perp} \in \Sigma^{pn}$. Then, $\mathcal{C}$ is a valid (Σ, p, n, k) vector linear index code of normalized rate k/p for the directed side information graph G on n vertices iff $\mathcal{C}^{\perp}$ is a valid $(\Sigma, p, n, np-k)$ vector linear GLRC of normalized rate $n - k/p$ when G is taken as a directed recoverability graph.* ◊

We develop upper bounds on $R^{\text{MU}}(\mathcal{G})$, the optimum sum-rate supported by an MU network with independent sources under vector-linear codes. In fact, our bounds are developed for $R^{\text{CO}}(\mathcal{G})$, the optimum vector-linear joint source entropy rate in the relaxed-correlated sources problem.

4.3.2 From Multiple-Unicasts Network Coding to Index Coding

Consider a multiple-unicasts network $\mathcal{G}$ with k sources and m links. Let $\mathcal{G}'$ be a directed cyclic network constructed from $\mathcal{G}$ by setting $t(e) = t_i$, $\forall e \in \mathcal{E}_i$, $i = 1, \ldots, k$, *i.e.*, setting the destination node t_i to be the tail of every source link of source s_i.

Let G be the *(reversed)* [4] *line graph* of $\mathcal{G}'$, *i.e.*, a directed graph on m vertices corresponding to the m links in $\mathcal{G}'$, with a directed edge from vertex v to $\widehat{v}$ corresponding to links e and $\widehat{e}$, respectively, iff $h(e) = t(\widehat{e})$ in $\mathcal{G}'$.

Theorem 4.3.2. *Consider a multiple-unicasts network $\mathcal{G}$ with m links, and a vector-linear code $\mathcal{C}$ with correlated sources, achieving joint source entropy rate r. The dual code $\mathcal{C}^{\perp}$ is a vector-linear index code achieving rate $m-r$ in the index coding instance with side-information graph G constructed based on $\mathcal{G}$ as described in Section 4.3.2.*

Proof. We give a brief proof sketch. The proof of this uses the observation that a vector linear code for the relaxed-correlated sources problem on $\mathcal{G}$ is equivalent to a vector linear code for a GLRC problem on G constructed as described above in Section 4.3.2. The duality result of Theorem 4.3.1 is then applied to get the above result. □

Corollary 4.3.1. *If G is the directed graph constructed based on the network $\mathcal{G}$ as described in Section 4.3.2, then*

$$R^{\mathrm{CO}}(\mathcal{G}) \;=\; m - \beta_{\mathrm{VL}}(G)\,.$$

We exploit the connection established in Cor. 4.3.1 to develop upper bounds on the joint source entropy rate $R^{\mathrm{CO}}(\mathcal{G})$, by lower bounding the vector linear rate $\beta_{VL}(G)$ of the associated index coding problem on the side information graph G.

[4] We refer to G as the *reversed* line graph of $\mathcal{G}$ because the direction of its edges is reversed compared to the typical definition of a line graph.

Definition 4.3.1. $\textsf{MAIS}(G)$ *of a directed graph G is the cardinality of the largest set $\widehat{V} \subseteq V(G)$ such that the subgraph of G induced by $\widehat{V}$ is acyclic.*

A Feedback Vertex Set (FVS) $\mathcal{F}$ is a set of vertices such that the subgraph of G induced by $V(G) - \mathcal{F}$ is acyclic. By definition, $m - \textsf{MAIS}(G)$ is the cardinality of the *minimum feedback vertex set* in G.

It is known that the size of the maximum acyclic subgraph of G is a lower bound on $\beta_{\text{VL}}(G)$. Tighter bounds can be obtained via graph tensorization.

Lemma 4.3.1. *The optimum broadcast rate $\beta_{\text{VL}}(G)$ of an index coding instance with side-information graph G, satisfies*

$$\sqrt[q]{\textsf{MAIS}(\otimes^q G)} \;\leq\; \beta_{\text{VL}}(G)\,, \quad \forall q \in \mathbb{Z}^+,$$

where $\otimes^q$ denotes the strong product of G with itself q times.

Proof. The proof of this requires a lengthy treatment of strong graph products. Hence, we refer the reader to the long arxiv version [78] of the paper [79]. □

Theorem 4.3.3. *Consider a multiple-unicasts network $\mathcal{G}$ with k sources and m links. Further, let G be the digraph on m vertices obtained from $\mathcal{G}$ as described in Section 4.3.2. Then,*

$$\begin{aligned} R^{\text{MU}}(\mathcal{G}) \;\leq\; R^{\text{CO}}(\mathcal{G}) &= m - \beta_{\text{VL}}(G) \\ &\leq\; m - \sqrt[q]{\textsf{MAIS}(\otimes^q G)}, \quad q \in \mathbb{Z}^+. \end{aligned}$$

4.3.3 GNS cut bound and its relation to Index Coding bounds

We now show that the GNS-cut bound (on a slightly modified but equivalent network) on the multiple unicast network coding problem with independent sources is effectively equivalent to the $\textsf{MAIS}(G)$ bound for index coding. First, recall the definition of the GNS cut:

Definition 4.3.2 ([66]). *A GNS cut of a multiple-unicasts network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with k sources, is a subset $S \subset \mathcal{E}$ such that for $\mathcal{G} - S$ (i.e., the network obtained by removing the links in S from $\mathcal{G}$) the following holds: there exists a permutation $\pi : [k] \to [k]$ such that $\forall i, j \in [k]$, if $\pi(i) \geq \pi(j)$, then no path exists from source s_i to destination t_j.*

The size of the smallest (in terms of capacity) GNS cut, denoted by $\textsf{GNSCUT}(\mathcal{G})$, is an upper bound on the *non-linear* sum-rate of the multiple-unicasts problem with independent sources [66]. In other words, the bounds of Theorem 4.3.3 are at least as tight as the GNS cut bound.

The GNS cut can be redefined on an equivalent network to yield an upper bound exactly equal to $m - \textsf{MAIS}(G)$. We achieve that by obtaining the GNS cut bound on a modified, yet equivalent network. Given a multiple-unicasts network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with k sources and m links, consider a network $\widetilde{\mathcal{G}}(\widetilde{\mathcal{V}}, \widetilde{\mathcal{E}})$ obtained from $\mathcal{G}$ as follows:

1. Introduce k nodes $\widetilde{s}_1, \ldots, \widetilde{s}_k$ to $\mathcal{G}$, *i.e.*, $\widetilde{\mathcal{V}} = \mathcal{V} \cup \{\widetilde{s}_i\}_{i=1}^{k}$.
2. Set $t(e) = \widetilde{s}_i$, $\forall e \in \mathcal{E}_i$, $i = 1, \ldots, k$, that is, set $\widetilde{s}_i$ as the tail of all source links of source s_i.

3. Introduce a set $\widetilde{\mathcal{E}}_i$ of $|\mathcal{E}_i|$ new links with head $\widetilde{s}_i$ and no tail, for all $i \in \{1, \ldots, k\}$.

The modified network $\widetilde{\mathcal{G}}$ is a multiple-unicasts network with k sources $\widetilde{s}_1, \ldots, \widetilde{s}_k$ and respective destinations $t_1, \ldots, t_k$. One can verify that $R^{\text{CO}}(\widetilde{\mathcal{G}}) = R^{\text{CO}}(\mathcal{G})$. The key difference is that the $|\mathcal{E}_i|$ source links of source s_i in $\mathcal{G}$ have become regular links in $\widetilde{\mathcal{G}}$ and can be used in a GNS cut. Thus, the bound obtained on the modified network is potentially tighter, *i.e.*, $\text{GNSCUT}(\widetilde{\mathcal{G}}) \leq \text{GNSCUT}(\mathcal{G})$.

Theorem 4.3.4. *Consider a multiple-unicasts network $\mathcal{G}$ with k sources and m links. Let G be the digraph on m vertices obtained from $\mathcal{G}$ as described in Section 4.3.2, and $\widetilde{\mathcal{G}}$ the modified network constructed as described above. Then, any feasible feedback vertex set of G corresponds to a GNS cut in $\widetilde{\mathcal{G}}$ with the same capacity. In turn,*

$$m - \text{MAIS}(G) = \text{GNSCUT}(\widetilde{\mathcal{G}}).$$

Proof. We provide a brief proof sketch. The proof relies on showing that each GNS cut in $\widetilde{\mathcal{G}}$ corresponds to a *Feedback Vertex Set* (FVS) of equal capacity in the digraph G. □

Remark 4.3.1. *(Thm. 1 in Chapter 2 of [66], Thm. 2 in [80]) can be generalized to show that $\text{GNSCUT}(\mathcal{G})$ also upper bounds the non-linear joint source entropy rate in the problem of relaxed-correlated sources.*

4.4 Approximating the GNS-cut bound

We describe an algorithm to approximately compute the GNS cut bound for a given acyclic network $\tilde{\mathcal{G}}$. We exploit Theorem 4.3.4, the special structure of a multiple-unicasts network $\widetilde{\mathcal{G}}$, and a known approximation algorithms for the *Feedback Vertex Set problem* on a digraph.

The Feedback Vertex Set (FVS) problem, *i.e.*, the problem of finding the smallest set $\mathcal{F}$ of vertices such that the subgraph of a digraph G induced by $V(G) - \mathcal{F}$ is acyclic. This problem is is NP-complete [81]. The LP dual of its LP relaxation is the *fractional cycle packing problem* [81, 82]. A fractional cycle packing is a function $q(C)$ from the set of cycles $\mathcal{C}$ in G to $[0,\ 1]$, satisfying $\sum_{C \in \mathcal{C}: v \bigcap C \neq \emptyset} q(C) \leq 1$, for each $v \in V(G)$. Letting $|q| = \sum_{C \in \mathcal{C}} q(C)$, the *fractional cycle packing number* $r_{\text{CP}}(G)$ of G is defined to be the maximum of $|q|$ taken over all fractional cycle packings q in G. It is known that:

$$r_{\text{CP}}(G) \ \leq \ m - \textsf{MAIS}(G).$$

An *optimal* fractional cycle packing [83] [82] (or an $(1+\epsilon)$ approximation, $\epsilon > 0$) can be computed in polynomial time (in m, ϵ^{-1}). G has a special structure when it is the (reverse) line-graph of a multiple-unicasts network $\mathcal{G}'$ (itself a modification of a network $\mathcal{G}$) as described in Section 4.3.2. For this case, we show that there exists a polynomial-time algorithm that exploit this additional structure to compute a feedback vertex set in G which is within a $O(\log^2 k)$ factor from $r_{\text{CP}}(G)$ [82].

Theorem 4.4.1. *Consider a multiple-unicasts network $\mathcal{G}$ with k sources and m unit-capacity links. Let G be the digraph on m vertices obtained from $\mathcal{G}$ as described in Section 4.3.2. Then, a feedback vertex set of size at most $r_{\text{CP}}(G) \cdot O\big(\log^2 k\big)$ can be* *computed in polynomial time. This satisfies:*

$$\begin{aligned} r_{\text{CP}}(G) &\leq m - \beta_{\text{VL}}(G) \\ &\leq m - \textsf{MAIS}(G) = \textsf{GNSCUT}(\widetilde{\mathcal{G}}) \leq r_{\text{CP}}(G) \cdot O\left(\log^2 k\right), \end{aligned}$$

where $r_{\text{CP}}(G)$ *is the* fractional cycle packing number *of* G. *Further,* $r_{\text{CP}}(G)$ *also equals the joint source entropy rate supported by a feasible (and polynomial-time computable) vector-linear multiple-unicasts network code for the relaxed-correlated sources problem on* $\mathcal{G}$.

This with Theorem 4.3.4 gives an $O(\log^2 k)$ approximation algorithm for GNS-cut of a multiple unicast network $\tilde{\mathcal{G}}$.

4.5 Separation results

The GNS cut, similar to the novel bounds of Theorem 4.3.3, upper bound the optimum vector-linear joint source entropy rate for the relaxed-correlated sources, and in turn for independent sources since $R^{\text{CO}}(\mathcal{G}) \geq R^{\text{MU}}(\mathcal{G})$. However, it remains unclear how the gap between the two rates scales. The following theorem takes a step towards addressing this question.

Theorem 4.5.1. *For any prime field* $\mathbb{F}_p$, *for any constant* $\delta > 0$, *there is a* k *sufficiently large and there exists a family of multiple-unicasts network instances* $\mathcal{G}$ *with* k *sources (*k *sufficiently large) for which* $R^{\text{CO}}\left(\mathcal{G}; \mathbb{F}_p\right) \geq k^{1-\delta} \cdot R^{\text{MU}}\left(\mathcal{G}; \mathbb{F}_p\right)$. *Further, for any two fields* $\mathbb{F}_p$ *and* $\mathbb{F}_q$, *for any* $\delta > 0$, *there is a large enough* k *and a multiple-unicasts network* $\mathcal{G}$ *such that* $R^{\text{MU}}\left(\mathcal{G}; \mathbb{F}_q\right) \geq k^{1-\delta} \cdot R^{\text{MU}}\left(\mathcal{G}; \mathbb{F}_p\right)$. [5]

[5] After our work in [78], we discovered that the uncertainty principle for vector linear index

Proof. The proof of this requires a lengthy treatment of strong graph products. Hence, we refer the reader to the long arxiv version [78] of the paper [79]. Here, we give a brief proof sketch. The counter examples are the same index coding counter examples given in [51]. The previous results separated symmetric broadcast rates of scalar linear index codes over two different fields. Now, we use two ideas to derive the above result using the same set of examples: 1) An 'uncertainty principle' for vector linear index codes, i.e. $\beta_{\mathrm{VL}}^{\mathbb{F}}(G)\,\beta_{\mathrm{VL}}^{\mathbb{F}}\left(\bar{G}\right) \geq |G|$ and 2) A symmetrization argument that shows that the maximum sum-rate for these examples is obtained at the symmetric rate point. □

Theorem 4.5.1 effectively states that for a *fixed* field, there exists networks for which the optimum sum-rate over all vector-linear codes *over that field* is almost a k-factor away from the GNS cut bound. Second, when designing a vector-linear code for a given multiple-unicasts network, the choice of field can have a tremendous impact on performance: it can affect the achievable sum-rate by almost a factor of k.

4.6 Conclusion

We showed a duality result between vector linear index codes and locally repairable codes. Inspired by this connection, we showed that the GNS cut bound

codes is implied by results in the PhD Thesis [77]. However, we provide an independent graph theoretic proof in [78] involving blow-ups and strong products of graphs recovering some of the results relating to the shannon capacity of the graphs proved in [77]. Further, the machinery developed is also used to prove stronger results regarding tensorization of the MAIS(G) bound in Theorem 4.3.3.

can be approximated within an $O(\log^2 k)$ factor in polynomial time. This is, to the best of our knowledge, the only case in the family of generalized cut-set bounds [66] that can be efficiently approximated. Finally, we show the importance of the field used by the vector-linear code: the GNS cut and the capacity can be very far from the best vector-linear code over a poorly chosen field. These together can be seen to form a set of 'partial' information flow-cut gap results for the multiple unicast problems.

Chapter 5

Index Coding with Cache Design: Finite File Size Analysis

5.1 Introduction

In the previous chapters [1], we have considered the index coding problem whose aim is to *design* the number of coded transmissions for a bottlenecked broadcast link. The link serves files requested by many users where each user has access to some side information (subset of the files requested by all users) distinct from the file they have requested. Prior to that we have considered the Femto-caching problem wherein each user had fast access to a subset of caches in a distributed caching system. If a user file request has no cache hit in the nearby caches, that file is served by a common broadcast link. The question was to design all the caches *apriori* by placing popular files from a known library as to maximize sum cache hit of all users in the system reducing the congestion in the bottleneck link. The near-optimal design depended on the topology of the connections between the caches and the user.

[1] The material in this chapter is based on the conference paper: K. Shanmugam, M. Ji, A.M. Tulino, J. Llorca and A.G. Dimakis, " Finite Length Analysis of Caching-Aided Coded Multicasting", *Allerton* 2014 (invited). A journal version with the same title has been accepted to the Transactions on Information Theory. The dissertation author is the primary contributor to the above paper and the results in this chapter.

In this chapter, we take a step forward and consider the cache design problem and the transmission design problem *together*. We call this problem Index Coding with Cache Design [2]. In a line of work initiated by [23, 84], the problem of optimizing cache placement and coded delivery has been considered. This problem has been referred to as either the coded caching or the caching-aided coded multicasting problem. Hereafter, we refer to this simply as the Index Coding with Cache Design (ICCD) problem. The setting is same as the index coding problem where there is a library of N files from which user requests arise and every device has a memory of size M. All users are served by a single broadcast link which is noiseless. There are totally K users. The difference is that there is a *placement phase*, which is free of cost, that involves populating all user caches with files from the library before the requests of users are revealed. Then the demand pattern is revealed and the delivery phase involves coded transmissions. The aim is to design the placement and the delivery phase such that the peak transmissions rate over all demand patterns is minimized in the delivery phase. It has been shown that the peak rate need only be roughly $\frac{N}{M}$, independent of the number of users [23]. It is possible even if each user cache populates itself randomly and independently of other users in the placement phase. Order optimal average rate under the uniform and Zipf demand distributions have also been characterized. However, all the achievable schemes work in the *asymptotic* regime when the number of packets per file scales to infinity. In this paper, we consider the case of peak rate over the

[2] This is known as either Coded Caching or Caching-aided Coded Multicasting problem in the literature. However, we adopt the alternate name so that it reflects the flow of ideas from previous chapters.

worst-case demand pattern for the ICCD problem. The main question we consider in this chapter is the file size needed to achieve a target coding gain.

Problem 5.1.1. *Let K users served by a common broadcast link. Let every user possess a cache of size M (i.e. a cache that can store M files). Let the user requests arise from a library of N files. What is the minimum number of file packets per file, denoted F, is needed such that the peak broadcast rate with coded transmissions is roughly K/g where g is the target coding gain. ?*

For a class of random uncoordinated placement schemes (called decentralized schemes in the literature) where each user cache caches randomly and independently of each other, we derive a lower bound for F needed for a given target gain g under a broad class of delivery schemes called clique cover schemes. We also give a new delivery scheme that works with existing random placement schemes that achieves this lower bound up to polynomial factors in K, N, M. We also show that the previous asymptotic information theory results give only a coding gain of at most 2 when F is exponential in $\frac{KM}{N}$.

5.1.1 Related Work

In the ICCD problem, there is a common broadcasting agent serving K users through a noiseless broadcast channel. Every user requests a file from a set of N files. Each file consists of F bits or packets. Every user has a cache of size M files. Files or parts of it ('packets') are placed in every cache *prior* to transmissions assuming that the library of file requests is known in advance. The objective is to design a placement scheme and delivery scheme that optimizes (or approximately optimizes) the maximum number of file transmissions required over all possible demand patterns. This problem has been well studied in the asymptotic regime when $F \to \infty$.

A deterministic caching and delivery scheme that requires $\binom{K}{KM/N}$ packets per file to achieve a gain of $t = KM/N$ was proposed in [84]. Following this, a random placement scheme that allows populating user caches independently of each other was proposed in [23]. In this uncoordinated placement phase, every user caches MF/N packets of each file $n \in [1 : N]$ chosen uniformly at random and independently of other caches. The delivery scheme is a greedy clique cover on the side information graph induced by the underlying index coding problem (refer Section 5.2). In any clique cover scheme, a set of packets of different files are XORed if for all packets, at least one user desiring it can recover it only by using its cache contents. For example if $A+B+C$ was sent, a user wanting A could recover A if the user has B and C stored in its user cache. The *peak broadcast rate* (number of file transmissions) of a specific cache configuration is the number of transmissions needed in the worst case over all demand patterns of the K users from the library. When random placement schemes are used, the broadcast rate is averaged over the randomness in caching for a given demand pattern and the worst case average rate over all demand pattens is obtained. We call this the *peak average broadcast rate. The peak average rate of the greedy clique cover delivery scheme under an uncoordinated random placement was shown to be [23] (in the limit $F \to \infty$):*

$$R_p(M) = \frac{K\,(1 - M/N)}{(KM/N)}\left(1 - (1 - M/N)^K\right) \approx \frac{K}{KM/N} = \frac{K}{t} \tag{5.1}$$

Here, $R_p(M)$ denotes the peak average broadcast rate. Note that, if coded multicasting is not used then the rate is given by $K(1 - M/N)$ from the gain due to just local cache hits. It was shown through cut-set bounds that the result in (5.1) was optimal up to a constant factor. The asymptotic multiplicative gain over the uncoded setting due to coding is roughly $t = KM/N$. The placement and delivery algorithms that achieve this peak average rate are given in Algorithms 1 and 2 respectively.

Algorithm 1 OldPlacement (Placement Algorithm in [23])

```
Input: Parameters $K, M, N$ and $F$.
for every user $k \in [1 : K]$ do
    for every file $n \in [1 : N]$ do
        Choose a random $MF/N$ subset of $F$ packets of file $n$ and place it in cache $k$.
    end for
end for
Output: Cache configuration for every user $k \in [1 : K]$.
```

Algorithm 2 OldDelivery (Delivery Algorithm in [23]). XORing (⊕'ing) vectors of different lengths means that all shorter vectors are zero padded to match the longest and then XORed.

```
Input: Parameters $K, M, N$ and $F$, caches for all users $k \in [1 : K]$ and demand set $\mathbf{d} = [d_1, d_2 \ldots d_K]$.
for every subset $\mathcal{S} \subseteq [1 : K]$ do
    Let $V_{k,\mathcal{S}-k}$ be the vector of packets from file requested by user $k$ but stored exactly in the set of caches $\mathcal{S} - k$.
    Transmit $\oplus_{k \in \mathcal{S}} V_{k,\mathcal{S}-k}$.
end for
```

This was followed by the works of [85] and [86] where they analyze the case of average number of transmissions when the user demand follows a popularity distribution over the library. Specifically, authors in [86] consider the case in which file requests follow a Zipf popularity distribution. They provide caching and delivery schemes that achieve order optimal average number of transmissions in the asymptotic regime. The caching distribution, unlike in the worst-case, has to be designed with respect to the collective demand distribution. Interestingly, they also showed that for the Zipf parameter between 0 (uniform popularity) and 1, even the peak rate scheme given above is sufficient for order optimality in the asymptotic regime $F \to \infty$.

5.1.2 Our Contribution

We consider the ICCD problem with K users, N files in the library and a cache size of M files. We are interested in the peak broadcast rate (number of file transmissions) for the worst-case demand. Our contributions are:

1. *Impossibility results for existing schemes* - We show that the existing random uncoordinated placement scheme (Algorithm 1) for this problem and its delivery scheme (Algorithm 2) has a broadcast rate above $\frac{K(1-M/N)}{2}$ when $F \leq \frac{(N/M)}{K} \exp\left(KM/N\right)$. We prove this in Theorem 5.3.1.

2. We propose a slightly modified placement scheme (Algorithm 3) that simplifies analysis for an efficient achievable scheme that we propose later in this paper. It also helps simplify analysis in several places. We show that the old delivery algorithm (Algorithm 2) coupled with the new placement scheme has similar file size requirements suggesting a needed change in the delivery scheme. We show this in Theorem 5.3.2

3. *General Impossibility results*- We show that, under any random placement scheme which is independent and symmetric across users (every file packet placement in a user cache is independent of its placement in other caches, every file packet has equal marginal probability of being placed in a cache), any clique cover based scheme (using clique cover on the side-information graph) requires a file size of approximately $\Omega(\frac{g}{K}(N/M)^{g-1})$ for achieving a peak average rate of $\frac{K}{\frac{4}{3}g}(1 - M/N)$. Here, the average is over the random caching involved. We show this universal impossibility result for random placement and clique cover delivery schemes in Theorem 5.3.3.

4. *Approximately optimal Scheme* - This is our main algorithmic contribution. We finally exhibit a modified delivery scheme (Algorithm 5) that improves on Algorithm 2 through an extra pre-processing step. This modified delivery scheme applied with a specific user grouping along with the new placement scheme provably achieves a rate of roughly $\frac{4K}{3(g+1)}$ with a file size of $\Theta((\lceil N/M \rceil)^{g+1}(\log(N/M))^{g+2}(3e)^{g})$ approximately matching the lower bound. The new placement scheme we propose plays an important role in the analysis of this algorithm. We show this in Theorems 5.4.1 and 5.4.2.

5. *Numerical Results*- We show the effectiveness of our new delivery scheme Algorithm 5 compared to the existing schemes through numerical results for reasonable parameters. We show that for small target gains (i.e. gain g=3), our delivery scheme achieves the number of transmissions promised by our theory (i.e. $\frac{4K}{3(g+1)}$) while the existing algorithm almost lose all their coding gain for finite file sizes. We show this in Section 5.5.

Remark: If $N/M = \Theta(K^{\delta})$ for some $0 < \delta < 1$ and K large, then for a constant gain g, the our result requires $\Theta\left(K^{\delta(g+1)}\right)$ packets whereas the previous best known uncoordinated random caching schemes require a file size of $\Omega(\exp(K^{1-\delta}))$ for obtaining a gain of 2.

In Section 5.2, we provide the definitions of two random placement schemes ('old' placement scheme used in the literature and a 'new' placement scheme) and few delivery schemes, both previously used and new. In Section 5.3, we show that the previous delivery scheme, that works asymptotically very well, gives only a constant gain (of 2) even for exponentially large file sizes. We also show that any clique cover scheme with a random placement scheme that is 'symmetric' requires exponential file size in the 'target gain'. For constant target gains, the file size requirement is polynomial in the ratio of library size to the cache memory size per user. In Section 5.4, to bridge the gap, we design an efficient delivery scheme based on clique cover, which together with the new placement scheme, achieves the file size lower bound approximately. In Section 5.5, we provide numerical results to demonstrate the effectiveness of the new delivery scheme for reasonable system parameters.

5.2 Definitions and Algorithms

5.2.1 Problem Setup

We consider the problem of designing placement and delivery schemes when K users request files from a library of N files ($N > K$) and each user has a cache of size M files. When the number of users $K > N$, the problem is easier and has been already dealt with in [23]. In the placement phase, a file is divided into F packets/bits. Then each packet is placed in different user caches (randomly or deterministically). We are interested in an efficient placement scheme and an efficient delivery scheme consisting of coded XOR transmissions of various packets that optimizes the peak rate over worst-case demands. An efficient delivery scheme computes the coded transmissions needed in time polynomial in parameters N, K, F, M, while an efficient placement scheme ensures F is as small as possible, in particular, polynomial in N, K and M. Let us denote a set of demands by $\mathbf{d} = [d_1, d_2 \dots d_K]$, $d_k \in [1 : N]$. A packet f belonging to file $n \in [1 : N]$ is denoted by (n, f).

Definition 5.2.1. *After a placement scheme, cache configuration $\mathcal{C}$ is given by the family of sets $\mathcal{S}_{n,f}$ for all files n and $1 \leq f \leq F$ where $S_{n,f} \subseteq [1 : K]$ is the set of user caches in which packet (bit) f of file n is stored.*

Every demand $\mathbf{d}$ and a cache configuration induces a directed side information graph $G = (V, E)$ where there are KF nodes where (d_k, f) is the label for each node representing the f-th packet of file d_k. There is a directed edge from (d_k, f) to (d_j, f') if the file packet f' of file d_j is stored in the user cache k.

Definition 5.2.2. *A clique cover delivery scheme corresponds to covering nodes of G by cliques. A clique is a set of vertices where there are edges in either direction between all vertices.*

It is easy to see that, XORing all the packets in the clique formed by (d_{k_1}, f_1),(d_{k_2}, f_2) $\dots(d_{k_m}, f_m)$ implies that user $k_j, 1 \leq j \leq m$ will be able to decode the packet

(d_{k_j}, f_j) by using all other packets in the XOR from its cache. Note that, here we do not require the demands to be distinct. In all the algorithms we discuss in this work, XOR operator $\oplus$ operating on a set of non-distinct packets is equivalent to the XOR operator $\oplus$ operating only on the distinct packets.

Let $R^A(\mathcal{C}, \mathbf{d})$ be the number of normalized transmissions (total number of bits broadcast divided by file size F) achieved by a **given** generic clique cover scheme A on the side information graph induced by the placement $\mathcal{C}$ and demand $\mathbf{d}$. In the literature, sometimes $R^A(\mathcal{C}, \mathbf{d})$ is also called *broadcast rate* or simply *rate* of algorithm A. We replace A by a short italicized string to denote various algorithms. In this work, we are broadly interested in the following question:

Over all possible random placement schemes c and clique cover delivery schemes A, what is the minimum file size F required (as a function of K, M and N) such that $\max_{\mathbf{d}} \mathbb{E}_c[R^A(\mathcal{C}, \mathbf{d})] \approx \frac{K}{g}$ for a fixed target coding gain $g \leq t = \frac{KM}{M}$ when $N > K$?

5.2.2 New Placement and Delivery Schemes

In this section, we first provide a description for the following:

1. Algorithm 3 - A new placement scheme that is slightly different from the already existing one in Algorithm 1. This placement scheme simplifies analysis and is a part of many results in this work. The new scheme reduces lots of unwanted correlations between different packets belonging to the same file helping us in analysis in several places. This helps us simplify analysis in many of our proofs.

2. Algorithm 4 - A new delivery scheme which is just an efficient polynomial time (in both K and F) implementation of the old delivery scheme in Algorithm 2. We prove the equivalence in this section.

3. Algorithm 5 - This is our **main** algorithmic contribution and this is a new delivery scheme which gives approximately the best performance among all clique cover delivery schemes when suitable random placement schemes are used. Results regarding the optimality of this scheme are proven in Section 5.4.

Algorithm 3 NewPlacement

```
Input: Parameters $K, M, N$ and $F$.
Let $F = \lceil N/M \rceil F'$ packets and $F'$ is an integer. Let every file be divided into $F'$ groups each of size $\lceil N/M \rceil$ each.
for every user $k \in [1 : K]$ do
    for every file $n \in [1 : N]$ do
        for $f' \in [1 : F']$ do
            $f'$-th packet of file $n$ in user $k$'s cache is randomly uniformly chosen from the set of $\lceil \frac{N}{M} \rceil$ packets of group $f'$ of file $n$.
        end for
    end for
end for
Output: Cache configuration for every user $k \in [1 : K]$.
```

5.2.2.1 Notation

$x_{d_k,f}$ in Algorithm 4 refers to the content of packet f of the file d_k. Let $R^{nd}(\mathcal{C}, \mathbf{d})$ denote the normalized transmissions achieved by Algorithm 4. Here, the string nd denotes the delivery scheme in Algorithm 4. Here, A in $R^A(\mathcal{C}, \mathbf{d})$ is replaced by a string nd to denote Algorithm 4. Let $\mathbf{d}_u$ denote a set of distinct

Algorithm 4 NewDelivery

```
**Input:** Parameters $K, M, N$ and $F$, caches for all users $k \in [1:K]$ and demand set $\mathbf{d} = [d_1, d_2 \dots d_K]$.
Let $C = \emptyset$. Let $S_{d_k,f} \subseteq [1:K], \; \forall k \in [1:K], \; f \in [1:F]$ be the exact subset of users in which the $f$-th packet of file requested by user $k$ is stored.
Let $D \subset [1:K] \times [1:F]$ be the file packets that are stored in the user requesting the corresponding file, i.e. $D = \{(d_k, f) : k \in S_{d_k,f}\}$.
**for** $(d_k, f) \in [1:K] \times [1:F] - D$ **do**
    **if** $(d_k, f) \notin C$ **then**
        Let $A = \emptyset$.
        **for** $j \in [1:K] - k$ **do**
            **if** $\exists (j, f') \notin C$ for some $f' : S_{d_j,f'} = S_{d_k,f} \bigcup k - j$ **then**
                $A \leftarrow A \bigcup (j, f')$
            **end if**
        **end for**Transmit $x_{d_k,f} \oplus_{(j,f') \in A} x_{d_j,f'}$.
$C \leftarrow C \bigcup (d_k, f) \bigcup A$.
    **else**
        Proceed with the next iteration.
    **end if**
**end for**
```

Algorithm 5 Modified Delivery

```
**Input:** Parameters $K, M, N, g$ and $F$, caches for all users $k \in [1:K]$ and demand set $\mathbf{d} = [d_1, d_2 \dots d_K]$.
Let $S_{d_k,f} \subseteq [1:K], \; \forall k \in [1:K], \; f \in [1:F]$ be the exact subset of users in which the $f$-th packet of file requested by user $k$ is stored.
**for** $(d_k, f) \in [1:K] \times [1:F]$ **do**
    **if** $|S_{d_k,f}| \geq g + 1$ **then**
        $S_{d_k,f} \leftarrow$ a random $g$-subset of $S_{d_k,f}$
    **end if**
**end for**
Run Algorithm 4 with this new cache configuration.
```

demand requests by users, i.e. every user requests a distinct file. Let $R^{\text{opt}}(\mathcal{C}, \mathbf{d})$ denote the number of normalized transmissions under the optimal clique cover scheme on the side information graph due to the cache configuration $\mathcal{C}$ and the demand pattern $\mathbf{d}$.

When $\mathcal{C}$ is chosen randomly, $R^{nd}(\mathcal{C}, \mathbf{d})$ is a random variable. Let $\mathbb{E}_c$ denote expectation taken over the cache configuration according to a specified random placement described by the string c. Further, let $\mathbb{E}_d$ denote expectation over a demand distribution described by d. Let $\mathbb{E}_{c,d}$ denote the expectation with respect to both. Let c_{op} denote the 'old' random placement according to Algorithm 1. Let c_{np} denote 'new' random placement according to Algorithm 3. Let $\mathcal{R}^{md}(\mathcal{C}, \mathbf{d})$ be the random number of transmissions under Algorithm 5 given a fixed cache configuration $\mathcal{C}$ and demand pattern $\mathbf{d}$. In this case, there is further randomness that is a part of the delivery phase. Let $\mathbb{E}^{md}\left(\mathcal{R}^{md}(\mathcal{C}, \mathbf{d})\right)$ denote the expected number of transmissions with respect to the randomness in Algorithm 5 for a fixed cache configuration $\mathcal{C}$ and demand pattern $\mathbf{d}$.

5.2.2.2 Relationship between Algorithm 4 and Algorithm 2

For any cache configuration $\mathcal{C}$ it is easy to see that Algorithm 4 runs in time polynomial in K and F. More specifically, the run time is $O(K^2F^2)$. However Algorithm 2, according to the way it is implemented, checks every subset and therefore requires $\Omega(2^K + KF)$ running time in the worst case.

Therefore, we make the implementation of Algorithm 2 efficient, for the sake of clarity, to obtain Algorithm 4 which provably has running time polynomial in K and F. We now show that Algorithm 4 performs identically to Algorithm 2 in terms of number of transmissions.

Theorem 5.2.1. *The number of transmissions of Algorithm 4 is identical to the number of transmissions of Algorithm 2 for a given placement and a set of demands.*

From the above theorem and according to the code of Algorithm 2, the number of file transmissions under both Algorithm 2 and Algorithm 4 for a given cache configuration and demand is given by:

$$R^{nd}\left(\mathcal{C}, \mathbf{d}\right) = \sum_{\mathcal{S} \neq \emptyset} \frac{\max_{k \in \mathcal{S}} |V_{k,\mathcal{S}-k}|}{F} \tag{5.2}$$

5.2.2.3 Comments on Algorithm 5

Algorithm 5 has a preprocessing step, that we call the ‘pull-down phase’, in addition to Algorithm 4. Algorithm 5 emulates a ‘virtual’ alteration of any underlying cache configuration. The change in $S_{d_k,f}$ happens in such a way that the algorithm pretends that a file packet is being stored in a subset of a set of caches where it has been actually stored. We use the same notation $S_{d_k,f}$ to represent such a ‘virtual cache configuration’ that will be used for the delivery. For example, if a particular packet was stored in caches $\{1, 2, 3, 4, 5, 6\}$ and if $g = 3$, a random subset from this is chosen. So the resultant virtual cache configuration could be $\{1, 2, 3\}$ after this virtual re-assignment. The re-assignment phase is what we call the ‘pull down’ phase. This will allow us to ‘target’ the gain g (which is typically a lot lesser compared to best gain possible which is KM/N) more effectively if we use Algorithm 5 for delivery. Algorithm 5 also runs in time polynomial in K, F as it depends on Algorithm 4 apart from the pre-processing step that runs in polynomial time.

5.3 File Size Requirements under existing random placement schemes

5.3.1 Requirements for existing delivery schemes and placement schemes

Given any cache configuration $\mathcal{C}$ and demand $\mathbf{d}$, according to Theorem 5.2.1, the number of transmissions of Algorithm 2 and Algorithm 4 are *identical.* According to the code of Algorithm 2, the number of file transmissions under both Algorithm 2 and Algorithm 4 for a given cache configuration and demand is given by:

$$R^{nd}\left(\mathcal{C},\mathbf{d}\right)=\sum_{\mathcal{S}\neq\emptyset}\frac{\max\limits_{k\in\mathcal{S}}|V_{k,\mathcal{S}-k}|}{F} \tag{5.3}$$

We will analyze the above quantity in this section with respect to different placement schemes (Algorithms 1 and Algorithm 3). Therefore, the expected number of transmissions for both Algorithm 2 and Algorithm 4 under the new placement algorithm is $\mathbb{E}_{c_{np}}\left[R^{nd}\left(\mathcal{C},\mathbf{d}\right)\right]$. Under the old placement it is $\mathbb{E}_{c_{op}}\left[R^{nd}\left(\mathcal{C},\mathbf{d}\right)\right]$.

Consider any demand distribution for $\mathbf{d}$. Let $F=F'\lfloor\frac{N}{M}\rfloor$. Every file consists of F packets. Let $\mathbf{1}^{\mathcal{S}}_{d_k,f}$ be the indicator that the packet f $(1\leq f\leq F)$ in file d_k is placed exactly in the set $\mathcal{S}\subseteq[1:K]$ of caches. For the old placement of Algorithm 1, we have :

$$\mathbb{E}\left[\mathbf{1}^{\mathcal{S}}_{d_k,f}\right]=\left(\frac{1}{\frac{N}{M}}\right)^{|\mathcal{S}|}\left(1-\frac{1}{N/M}\right)^{K-|\mathcal{S}|} \tag{5.4}$$

For the placement scheme in Algorithm 3, we have:

$$\mathbb{E}\left[\mathbf{1}^{\mathcal{S}}_{d_k,f}\right]=\left(\frac{1}{\lceil\frac{N}{M}\rceil}\right)^{|\mathcal{S}|}\left(1-\frac{1}{\lceil N/M\rceil}\right)^{K-|\mathcal{S}|} \tag{5.5}$$

Also, for $\mathcal{S}$ that contain k, we have:

$$|V_{k,\mathcal{S}-k}| = \sum_{f=1}^{F} \mathbf{1}_{d_k,f}^{\mathcal{S}-k} \tag{5.6}$$

The expected number of transmissions for all stages $\mathcal{S}$ in Algorithm 2 (and Algorithm 4) with respect to the any placement c is given by:

$$\mathbb{E}_{\mathrm{c}}\left[R^{nd}(\mathcal{C},\mathbf{d})\right] = \sum_{\mathcal{S}\neq\emptyset} \frac{\mathbb{E}[\max\limits_{k\in\mathcal{S}}|V_{k,\mathcal{S}-k}|]}{F} \tag{5.7}$$

Denote by $\mathbf{d}_u$ the demand pattern in which all users request distinct files. We will consider this case. We assume that $N > K$ here. We are interested in the question: How far are $\mathbb{E}_{c_{np}}\left[R^{nd}(\mathcal{C},\mathbf{d}_u)\right]$ and $\mathbb{E}_{c_{op}}\left[R^{nd}(\mathcal{C},\mathbf{d}_u)\right]$ for finite F?

Now, we show that coding gain is at most 2, even when F is exponential in the asymptotic targeted gain $t = \frac{K}{\lceil\frac{N}{M}\rceil}$ for both placement schemes, namely Algorithm 1 and Algorithm 3.

We now show that, the coding gain is at most 2 even when the file size is exponential in the targeted gain $t = \frac{K}{N/M}$ when the existing delivery scheme (Algorithm 2 or equivalently Algorithm 4) is applied with the existing placement scheme of Algorithm 1.

Theorem 5.3.1. *Let $N > K$. Consider the case when demands are distinct, i.e. $\mathbf{d} = \mathbf{d}_u$. Then, $\mathbb{E}_{c_{op}}\left[R^{nd}\left(\mathcal{C},\mathbf{d}_u\right)\right] \geq \frac{1}{2}\left(1-\frac{M}{N}\right)K$ when*

$$F \leq \frac{N/M}{2K}\left(1-\frac{1}{N/M}\right)\exp\left(2t\left(1-\frac{t}{K}\right)\left(1-\frac{1}{K}\right)\right),\ t = KM/N. \tag{5.8}$$

Now, we prove an analogous result for the placement scheme in Algorithm 3 too.

Theorem 5.3.2. *Let $N > K$. Then,* $\mathbb{E}_{c_{np}}\left[R^{nd}\left(\mathcal{C}, \mathbf{d}_u\right)\right] \geq \frac{1}{2}\left(1 - \frac{M}{N}\right)K$ *when* $F \leq \frac{\lceil N/M \rceil}{2K}\left(1 - \frac{1}{\lceil N/M \rceil}\right)\exp\left(2t\left(1 - \frac{t}{K}\right)\left(1 - \frac{1}{K}\right)\right)$ *and* $t = \frac{K}{\lceil N/M \rceil}$.

Proof. We can use the exact identical proof in Theorem 5.3.1 with N/M replaced by $\lceil \frac{N}{M} \rceil$ and we get this result. We refer to the remark after the proof of Theorem 5.3.1 in the Appendix. □

Therefore, there is very little coding gain if we do not have exponential number of file packets in $t = KM/N$. Note that t is the best asymptotic gain possible in previous works [23] with random placement schemes.

5.3.2 Requirements for any Clique Cover Delivery Scheme

Let c_{up} denote a random independent and symmetric placement algorithm that has the following properties:

1. For any packet (n, f), the probability of placing this in a user cache k is independent of placing it in all other caches.

2. Placing of packets belonging to different files in the same cache is independent.

3. The probability of placing a packet equals M/N for a given cache.

Now, we have the following result on any clique cover scheme on the side information graph induced by random caching algorithm c_{up} and a unique set of demands $\mathbf{d}_u$.

Theorem 5.3.3. *When user demands are distinct, for any clique cover algorithm on the side information graph induced by the random cache configuration due to $\mathbf{c}_{up}$, if $\mathbb{E}_{c_{up}}\left(R\left(\mathcal{C}, \mathbf{d}_u\right)\right) \leq \frac{K(1-M/N)}{\frac{4}{3}g}$ for any $g > 2$, then we need the number of file packets $F \geq \frac{g}{2et}\left(\frac{N}{M}\right)^{g-2}$ where $t = KM/N$. Clearly, these bounds apply to both $\mathbf{c}_{op}$ and $\mathbf{c}_{np}$.*

Note: We would like to note that c_{up} represents a broad set of schemes where every file packet is placed in a cache independently of its placement elsewhere and no file packet is given undue importance over other packets belonging to the same file.

5.4 Efficient Achievable Schemes

5.4.1 Deterministic Caching Scheme with User Grouping:

Now, briefly we would like to explore what can be said about the file size requirements of deterministic placement schemes. In this section, we describe a variation of the deterministic caching scheme in [84] that requires a similar file size requirement as the previous section for a target gain of g. However, it is not clear if, for a clique cover scheme at the delivery stage, this is the best one can do with deterministic caching schemes. In other words, a lower bound for deterministic caching scheme similar to the one above is not known.

Now, we give a description of a deterministic caching and delivery scheme that requires $F = \binom{K}{g}$ packets to get a gain of $g + 1$. This follows directly from the deterministic scheme of [84]. For ease of exposition we describe it here: For every file, split the file into $\binom{K}{g}$ packets. For every subset $G \subset [1 : K]$ such that $|G| = g$, we place the corresponding packet in the user caches in the subset G. The total number of files per user cache is $N\frac{\binom{K-1}{g-1}}{\binom{K}{g}} = \frac{gN}{K} \leq M$. This satisfies the memory constraint because the gain $g \leq KM/N$. Following the same arguments in [84], it is easy to show that the peak transmission rate is at most : $\frac{K-g}{g+1}$.

Now, we show a slight modification of the deterministic caching scheme mentioned above which (approximately order wise) matches the lower bound in the previous section. Let us divide the users into groups of size $K' = g\lceil N/M \rceil$ and then apply the caching and delivery scheme for each group separately. The number of file packets required is $F = \binom{K'}{g}$. The memory constraint would be satisfied when $g \leq K'M/N = g\lceil N/M \rceil \left(M/N\right)$ which is true. Now, coded multicasting is done within every user group. The total number of transmissions is: $\frac{K}{K'}\frac{K'-g}{g+1} = \frac{K}{g+1}\left(1 - \frac{1}{\lceil N/M \rceil}\right)$. This requires $\binom{K'}{g} = O\left((\lceil N/M \rceil e)^g\right)$ packets.

5.4.2 New Randomized Delivery scheme

For the deterministic scheme described previously, similar to the one in [84], it is necessary to refresh (possibly) all the caches in a specific way when users leave or join the system that requires coordination among the caches. Now, we show that under an uncoordinated random caching scheme given by the new placement scheme in Algorithm 3 and a new randomized clique cover algorithm, it is possible to have an average peak rate (with respect to all the randomness) of about $\frac{K}{g+1}$ when $F = \Theta\left(g\binom{K}{g}\log K\right)$. In this section, we analyze our main delivery scheme Algorithm 5 to prove the above assertion.

Theorem 5.4.1. *Using the randomized Algorithm 3 for the placement scheme and* *the randomized Algorithm 5 for delivery, for any set of demands* $\mathbf{d}$*, the average peak rate, with respect to all the randomness (randomness in both delivery and placement) is given by* $\mathbb{E}^{md}_{\mathcal{C}_{\text{np}}}(\mathcal{R}^{md}(\mathcal{C}, \mathbf{d})) \leq \frac{4}{3}\frac{K}{g+1}(1 + o(1))$ *and the number of file packets needed is* $F = c\binom{K}{g}(\log(\binom{K}{g}))^2 \lceil N/M \rceil$ *for some constant* $c > 0$ *when* $2 \leq g \leq \frac{K}{3\lceil N/M \rceil}, \lceil N/M \rceil \leq \frac{K}{\frac{27}{4}\log K}, N > K.$

5.4.3 Grouping into smaller user groups approximately achieves the lower bound

We now propose a user grouping scheme similar to the one for the deterministic caching scheme which can achieve the same average number of transmissions as the scheme mentioned in the previous section but with improved file size requirement almost matching the lower bound.

We group users in groups of size $K' = \lceil N/M \rceil 3g(\log(N/M))$ and apply the new placement scheme (Algorithm 3) and delivery scheme of Algorithm 5 to each of the user groups. It can be seen that K' satisfies the conditions: $e \leq \lceil N/M \rceil \leq \frac{K'}{\frac{27}{4}\log K'}$ and $7 \leq g \leq \min\{\frac{K'}{3\lceil N/M \rceil}, \frac{(\frac{N}{M})^2}{3\log(N/M)}\}$. Therefore, Theorem 5.4.1 is applicable. For every group, the average number of transmissions for a particular demand configuration is at most $\frac{4}{3}\frac{K'}{g+1}(1 + o(1))$. Adding over all groups, we have the following theorem

Theorem 5.4.2. *Let the placement scheme be that of Algorithm 3. For any target gain* $7 \leq g \leq \frac{(\frac{N}{M})^2}{3\log(N/M)}$ *and* $\lceil N/M \rceil \geq e$*, let the number of users in the system be such that* K *is a large multiple of* $\lceil \frac{N}{M} \rceil 3g\log(N/M)$*. Consider the case when users are divided into groups of size* $K' = \lceil N/M \rceil 3g\log(N/M)$ *and delivery scheme of Algorithm 5* *is applied to each user group separately. For any demand pattern, the expected total number of transmission required for all users is at most* $\frac{4}{3}\frac{K}{g+1}(1+o(1))$. *The file size needed is* $F = \Theta(\binom{K'}{g}(\log(\binom{K'}{g}))^2\lceil N/M\rceil) \approx \Theta(\left(\frac{N}{M}\right)^{g+1}(3e)^g(\log(N/M))^{g+2}g^2)$.

Proof. Essentially, Theorem 5.4.1 is applied to all groups of size $K' = \lceil\frac{N}{M}\rceil 3g\log(N/M)$ and K' satisfies the conditions of Theorem 5.4.1. Adding up the contributions of various groups, we obtain the result stated in the theorem. □

Note: The constant e in the above requirement for file size comes due to bounding $\binom{n}{k}$ by $\left(\frac{ne}{k}\right)^k$. Other constants in the derivation can be relaxed if (D.8) can be strengthened which we do not do here. If $N/M = \Theta(K^\delta)$ for some $0 < \delta < 1$ and K large, then for a constant gain g, the above result requires $\Theta\left(K^{\delta(g+1)}\right)$ packets whereas the previous best known uncoordinated random caching schemes require a file size of $\Omega(\exp(K^{1-\delta}))$ for obtaining a gain of 2.

5.5 Numerical Results

In this section, we provide some simulations results to demonstrate the effectiveness of Algorithm 5. According to result in the previous section, Algorithm 5 should be applied to groups of size $\frac{cN}{M}\log(N/M)$ for some constant c to get the best file size tradeoffs. We fix the number of users to be $K = 40$ for our numerical simulations. For different ratios of N/M, namely $1/3$ and $1/4$, we plot the number of file transmissions required by various algorithms. $K = 40$ roughly satisfies the condition for the group size for the target gains and ratios considered in the simulations. We compare Algorithm 4 (or equivalently Algorithm 2) with our proposed

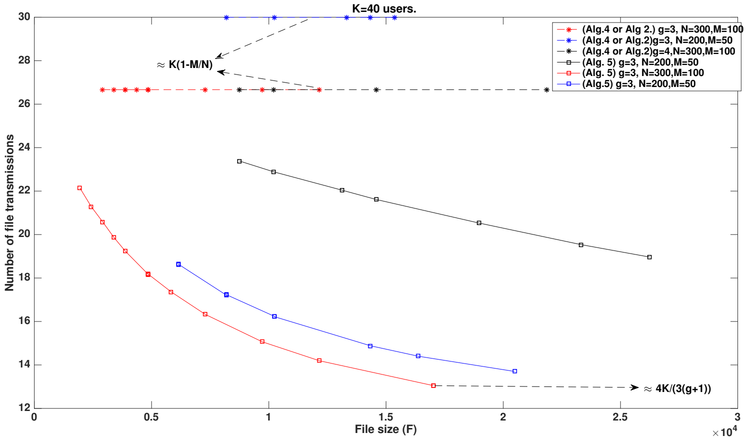


Figure 5.1: The figure compares the existing Algorithm 4 (or equivalently Algorithm 2), marked with dotted lines and star markers, with our proposed Algorithm 5 marked with solid lines and square markers. $K = 40$, target gains of $3, 4$ and $N/M = 3, 4$ are considered. Our proposed algorithm achieves close to what is predicted by Theorem 9 when $F \approx 15000 - 20000$. The existing algorithms lose almost all the coding gain in these file size regimes.

algorithm under the placement algorithm given by Algorithm 3 . We would like to note that implementing Algorithm 3 or Algorithm 1 makes very little difference for the placement and every point concentrates, with respect to the randomness involved, well with even 5 runs. The plot of comparing the algorithms is given in Fig. 5.1.

The figure compares the existing algorithms (Algorithm 4 or equivalently Algorithm 2), marked with dotted lines and star markers, with our proposed Algorithm 5 marked with solid lines and square markers. $K = 40$, target gains of $3, 4$ and $N/M = 3, 4$ are the scenarios considered in the plot. Our proposed algorithm achieves close to $\frac{4K}{3*(g+1)} \approx 13$ transmissions when the target gain is 3 as per Theorem 9 when $F \approx 15000 - 20000$ while the existing algorithms achieve only

$K(1 - M/N)$ transmissions losing almost all the coding gain in the finite file size regimes justifying our lower bounds. It is also instructive to note that the black solid curve is above the red solid curve although the target gain is 4 because the number of file packets is not enough to target a higher gain than 3. Therefore, targeting the right gain during the pull-down phase of Alg. 5 is very crucial.

5.6 Conclusion

We have analyzed random uncoordinated placement schemes along with clique cover based coded delivery schemes in the finite length regime for the caching-aided coded multicasting problem (or coded caching problem). This problem involves designing caches at user devices offline and optimizing broadcast transmissions when requests arise from a known library of popular files for worst case demand. The previous order optimal results on the number of broadcast transmissions for any demand pattern assumed that the number of packets per file is very large (tending to infinity). We showed that existing random placement and coded delivery schemes for achieving order optimal peak broadcast rate do not give any gain even when you have exponential number of packets. Further, we showed that to get a multiplicative gain of g over the naive scheme of transmitting all packets, one needs $\Omega(\frac{g}{K}(N/M)^{g-1})$ packets per file for any clique cover based scheme where N and M are the library size and cache memory size respectively. We also provide an improved random delivery scheme, that achieve this lower bound approximately. We demonstrate the improvements in the file size achieved by our proposed delivery scheme through numerical simulations. Future interesting research directions include designing improved deterministic coordinated caching schemes that have better file size tradeoffs with respect to the coding gain than the uncoordinated random caching schemes in this chapter.

Chapter 6

Learning Ising Graphical Models: Sample Complexity for Learning Random Ensembles

6.1 Introduction

Identifying relationships [1] among many variables from data is an important part of machine learning. Ideally, one would like to infer as much information as possible about a multi-variate distribution from as few samples obtained from them and in a computationally efficient manner. Distributional information gives an idea about how the observed variables are *correlated* with each other. Graphical models provide a compact representation of multivariate distributions using graphs that represent Markov conditional independencies in the distribution. They are thus widely used in a number of machine learning domains where there are a large number of random variables, including natural language processing [1], image processing [2–4], statistical physics [5], and spatial statistics [6], among others. Graphical model is a graph on the set of variables of interest such that the network structure encodes conditional independencies in the multi-variate distribution un-

[1] The material in this chapter is based on the conference paper: R. Tandon*, K. Shanmugam*, A. G. Dimakis, P. Ravikumar, "On the Information Theoretic Limits of Learning Ising Models", *NIPS, 2014* (*-equal contribution). The dissertation author's main contributions are towards the structural characterization with large correlation (Section 6.3.3) and the sample complexity lower bounds for Ising Models drawn randomly from Erdős-Rényi ensembles. This chapter contains results primarily relating to these main contributions.

derlying the observed data. It encodes all relationships of the form - Conditioned on the variables corresponding to the set of indices S, variables corresponding to indices in set A and variables corresponding to indices in set B are independent. An important use of graphical model is that it makes inference procedures (such as maximum likelihood estimation etc.) *computationally efficient* if the graphical model structure is sparse.

In many domains, a key problem of interest is to recover the underlying conditional independencies, represented by the graph, given samples *i.e.* to estimate the graph of conditional independencies given samples drawn from the distribution. A common regime where this graph selection problem is of interest is the *high-dimensional* setting, where the number of samples n is potentially smaller than the number of variables p. Given the importance of this problem, it is instructive to have *lower bounds* on the sample complexity of any estimator: it clarifies the statistical difficulty of the underlying problem, and moreover it could serve as a certificate of optimality in terms of sample complexity for any estimator that actually achieves this lower bound. We are particularly interested in such lower bounds under the structural constraint that the graph lies within a given class of graphs and the graph is *drawn randomly* from it.

In this chapter, we employ graph theoretic techniques combined with standard information theoretic techniques that determine sample complexity of learning graphical models from random samples. In particular, we study ferromagnetic Ising models where the underlying graphical model is drawn from a canonical distribution. We derive sample complexity bounds for average error of graphical model identification (or learning) given a distribution on the set of models. It is a follow-up of previous work which concerned itself with *worst-case* sample complexity requirements for a class of models with no distribution on it. One can consider this work to be the first step to reason about sample complexity considering the average error of learning graphical models drawn from random ensembles in general.

6.1.1 Previous work on worst case bounds

Previous works have mostly dealt with identifying a graphical model drawn from a given class of graphs and the sample complexity bounds are for the worst-case scenario: What is the sample complexity to distinguish *any* graph from the rest of the class? The simplest approach to obtaining such bounds involves counting arguments on the entire graph class, and an application of Fano's lemma. [87, 88], for instance, derive such bounds for the case of degree-bounded and power-law graph classes respectively. This approach however is purely counting-based, and thus fails to capture the interaction of the graphical model parameters with the graph structural constraints, and thus typically provides suboptimal lower bounds. (as also observed in [89]). The other standard approach requires a more complicated argument through Fano's lemma that requires finding a subset of graphs such that (a) the subset is large enough in number, and (b) the graphs in the subset are close enough in a suitable metric, typically the KL-divergence of the corresponding distributions. For the simple class of bounded degree graphs, [89] used the above approach to provide lower bounds for Ising models.

6.1.2 This work: Average error bounds for random graph classes

In modern high-dimensional settings, it is becoming increasingly important to incorporate structural constraints in statistical estimation, and graph classes are a key interpretable structural constraint. This means that pathological scenarios of a small (but large enough for bounding purposes) subset of graphs in a much larger class could determine sample complexity bounds. A more refined approach could discuss distinguishability only among *typical* graphs when a graph is randomly sampled from a graph class. In order to do that, the key ingredient involves finding structural characterizations that *typically differentiates* the most commonly occurring graphs within a graph class. In this work, the one we identify is : Connectivity by short paths between pairs of nodes.

Moreover, using structural arguments allows us to bring out the dependence of the edge-weights, λ, on the sample complexity. We use this to establish lower bound requirements for the class of Erdős-Rényi graphs in a moderately dense setting. Here, we show that under a certain scaling of the edge-weights λ, $\mathcal{G}_{p,c/p}$ requires exponentially many samples, as opposed to a polynomial requirement suggested from earlier bounds[90].

6.1.2.1 Contributions:

This is the main result (stated informally): Let us consider a dense Erdős-Rényi ensemble: $G \sim G(p, c/p)$, $c = \Omega(p^{3/4+\epsilon'})$. When $\lambda = \Omega(\sqrt{p}/c)$, a huge number (exponential in $\frac{\lambda^2 c^2}{p}$) of samples are required. Hence, for any efficient algorithm, we require $\lambda = O\left(\sqrt{p}/c\right)$ and in this regime $O\left(c \log p\right)$ samples are required to learn.

6.2 Preliminaries and Definitions

Notation: $\mathbb{R}$ represents the real line. $[p]$ denotes the set of numbers from 1 to p. Let $\mathbf{1}_S$ denote the vector of ones and zeros where S is the set of coordinates containing 1. Let $A - B$ denote $A \bigcap B^c$ and $A \Delta B$ denote the symmetric difference for two sets A and B.

In this work, we consider the problem of learning the graph structure of an Ising model. Ising models are a class of graphical model distributions over binary random vectors, characterized by the pair $(G(V, E), \bar{\theta})$ where $G(V, E)$ is an undirected graph on p vertices and $\bar{\theta} \in \mathbb{R}^{\binom{p}{2}} : \theta_{i,j} = 0 \ \forall (i, j) \notin E, \ \theta_{i,j} \neq 0 \ \forall \ (i, j) \in E$. Let $\mathcal{X} = \{+1, -1\}$. Then, for the pair $(G, \bar{\theta})$, the distribution on $\mathcal{X}^p$ is given as: $f_{G,\bar{\theta}}(\mathbf{x}) = \frac{1}{Z} \exp \left(\sum\limits_{i,j} \theta_{i,j} x_i x_j \right)$ where $\mathbf{x} \in \mathcal{X}^p$ and Z is the normalization factor, also known as the *partition function*.

Thus, we obtain a family of distributions by considering a set of edge-weighted graphs $\mathcal{G}_\theta$, where each element of $\mathcal{G}_\theta$ is a pair $(G, \bar{\theta})$. In other words, every member of the class $\mathcal{G}_\theta$ is a weighted undirected graph. Let $\mathcal{G}$ denote the set of distinct unweighted graphs in the class $\mathcal{G}_\theta$.

A learning algorithm that learns the graph G (and not the weights $\bar{\theta}$) from n independent samples (each sample is a p-dimensional binary vector) drawn from the distribution $f_{G,\bar{\theta}}(.)$, is an efficiently computable map $\phi : \chi^{np} \rightarrow \mathcal{G}$ which maps the input samples $\{\mathbf{x}_1, \ldots \mathbf{x}_n\}$ to an undirected graph $\hat{G} \in \mathcal{G}$ *i.e.* $\hat{G} = \phi(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

We now discuss two metrics of reliability for such an estimator ϕ. For a given $(G, \bar{\theta})$, the probability of error (over the samples drawn) is given by $p(G, \bar{\theta}) = \Pr\left(\hat{G} \neq G\right)$. Given a graph class $\mathcal{G}_\theta$, one may consider the maximum probability of error for the map ϕ, given as:

$$p_{\max} = \max_{(G,\theta)\in\mathcal{G}_\theta} \Pr\left(\hat{G} \neq G\right). \tag{6.1}$$

The goal of any estimator ϕ would be to achieve as low a $p_{\max}$ as possible. We will concentrate on the next metric in this chapter. Alternatively, there are random graph classes that come naturally endowed with a probability measure $\mu(G, \theta)$ of choosing the graphical model. In this case, the quantity we would want to minimize would be the average probability of error of the map ϕ, given as:

$$p_{\text{avg}} = \mathbb{E}_\mu\left[\Pr\left(\hat{G} \neq G\right)\right] \tag{6.2}$$

In this work, we are interested in answering the following question: For any estimator ϕ, what is the minimum number of samples n, needed to guarantee an asymptotically p_{avg} ? The answer depends on $\mathcal{G}_\theta$ and μ.

For the sake of simplicity[2], we impose the following restrictions. We restrict to the set of *zero-field ferromagnetic* Ising models, where *zero-field* refers to a lack of node weights, and *ferromagnetic* refers to all positive edge weights. Further, we will restrict all the non-zero edge weights $(\theta_{i,j})$ in the graph classes to be the same, set equal to $\lambda > 0$. Therefore, for a given $G(V, E)$, we have $\bar{\theta} = \lambda \mathbf{1}_E$ for

[2] Note that a lower bound for a restricted subset of a class of Ising models will also serve as a lower bound for the class without that restriction.

some $\lambda > 0$. A deterministic graph class is described by a scalar $\lambda > 0$ and the family of graphs $\mathcal{G}$. In the case of a random graph class, we describe it by a scalar $\lambda > 0$ and a probability measure μ, the measure being solely on the structure of the graph G (on $\mathcal{G}$).

Since we have the same weight $\lambda(> 0)$ on all edges, henceforth we will skip the reference to it, i.e. the graph class will simply be denoted $\mathcal{G}$ and for a given $G \in \mathcal{G}$, the distribution will be denoted by $f_G(.)$, with the dependence on λ being implicit. Before proceeding further, we summarize the following additional notation. For any two distributions f_G and $f_{G'}$, corresponding to the graphs G and G' respectively, we denote the Kullback-Liebler divergence (KL-divergence) between them as $D\left(f_G \| f_{G'}\right) = \sum_{x \in \mathcal{X}^p} f_G(x) \log\left(\frac{f_G(x)}{f_{G'}(x)}\right)$. For any subset $\mathcal{T} \subseteq \mathcal{G}$, we let $C_{\mathcal{T}}(\epsilon)$ denote an ϵ-covering w.r.t. the KL-divergence (of the corresponding distributions) *i.e.* $C_{\mathcal{T}}(\epsilon)(\subseteq \mathcal{G})$ is a set of graphs such that for any $G \in \mathcal{T}$, there exists a $G' \in C_{\mathcal{T}}(\epsilon)$ satisfying $D\left(f_G \| f_{G'}\right) \leq \epsilon$. We denote the entropy of any r.v. X by $H(X)$, and the mutual information between any two r.v.s X and Y, by $I(X; Y)$.

6.3 Ideas and Tools used

6.3.1 Conditional Fano's Lemma

Fano's lemma [91] is a primary tool for obtaining bounds on the probability of error. It provides a lower bound on the probability of error of any estimator ϕ in terms of the entropy $H(\cdot)$ of the output space, the cardinality of the output space, and the mutual information $I(\cdot\,,\,\cdot)$ between the input and the output of a channel. Here, G is chosen randomly from an ensemble $\mathcal{G}$ according to measure μ and it is the source of the channel. Then, the channel outputs samples $X^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ which are drawn from f_G (a distribution dependent on the input G chosen).

To obtain sharper lower bound guarantees, it is useful to consider instead a conditional form of Fano's lemma [90, Lemma 9], which allows us to obtain lower bounds on p_{avg}.

The conditional version allows us to focus on potentially harder to learn subsets which occur typically, leading to sharper lower bound guarantees. Also, for a random graph class, the entropy $H(G)$ may be asymptotically much smaller than the log cardinality of the graph class, $\log|\mathcal{G}|$ (e.g. Erdős-Rényi random graphs). The conditional version allows us to circumvent this issue by focusing on a *high-probability* subset of the graph class.

Lemma 6.3.1 (Conditional Fano's Lemma)**.** *Consider a graph class $\mathcal{G}$ with measure μ. Let, $G \sim \mu$, and let $X^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be n independent samples such that $\mathbf{x}_i \sim f_G$, $i \in [n]$. Consider any $\mathcal{T} \subseteq \mathcal{G}$ and let $\mu(\mathcal{T})$ be the measure of this subset* i.e. *$\mu(\mathcal{T}) = \Pr_\mu(G \in \mathcal{T})$. Then, we have*

$$p_{avg} \geq \mu(\mathcal{T}) \frac{H(G|G \in \mathcal{T}) - I(G; X^n|G \in \mathcal{T}) - \log 2}{\log|\mathcal{T}|}$$
$$p_{max} \geq \frac{H(G|G \in \mathcal{T}) - I(G; X^n|G \in \mathcal{T}) - \log 2}{\log|\mathcal{T}|}$$

Here, for the case of p_{max}, the the measure μ is taken to be uniform over the class $\mathcal{G}$.

Given Lemma 6.3.1, it is the sharpness of an upper bound on the mutual information that governs the sharpness of lower bounds on the probability of error (and effectively, the number of samples n).

We illustrate this bounding of the mutual information term in the simpler case of p_{max} and uniform measure. Using [92], mutual information can be upper bounded (upto small additive terms) to the size of covering set $|C_{\mathcal{T}}(\epsilon)|$, as in Section 6.2 in terms of the KL-divergence. Analogous bounds can be obtained for p_{avg} although they have some differing details and a remark is made at the end of this discussion.

Corollary 6.3.1. *Consider a graph class $\mathcal{G}$, and any $\mathcal{T} \subseteq \mathcal{G}$. Recall the definition of $C_{\mathcal{T}}(\epsilon)$ from Section 6.2. For any $\epsilon > 0$, we have $p_{max} \geq \left(1 - \frac{\log|C_{\mathcal{T}}(\epsilon)| + n\epsilon + \log 2}{\log|\mathcal{T}|}\right)$.*

Remark 6.3.1. *From Corollary 6.3.1, we get: If $n \leq \frac{\log|\mathcal{T}|}{\epsilon}\left((1-\delta) - \frac{\log 2}{\log|\mathcal{T}|} - \frac{\log|C_{\mathcal{T}}(\epsilon)|}{\log|\mathcal{T}|}\right)$, then $p_{max} \geq \delta$. ϵ is an upper bound on the radius of the KL-balls in the covering, and usually varies with λ.*

The corollary requires us to specify a subset $\mathcal{T}$ of the overall graph class, an ϵ-covering set $C_{\mathcal{T}}(\epsilon)$ in terms of KL divergence . Further, for the above result to make sense, clearly $|\mathcal{T}|$ must be exponentially much smaller than $|\mathcal{T}|$. Therefore, structural properties of the graph class $\mathcal{G}$ is needed to choose appropriate $\mathcal{T}$.

Remark 6.3.2. *In the case of p_{avg} which is of actual interest, one way to ensure a sensible bound is to choose $\mathcal{T}$ to be a typical set with measure close to 1 such that the covering set $C_{\mathcal{T}}(\epsilon)$ covers most of $\mathcal{T}$ with respect to measure μ and $|C_{\mathcal{T}}(\epsilon)|$ is exponentially much smaller than $|\mathcal{T}|$.*

We note that Fano's lemma and variants described in this section are standard, and have been applied to a number of problems in statistical estimation [89, 90, 92–94].

6.3.2 Structural Conditions governing Correlations

As discussed in the previous section, we want to find subsets $\mathcal{T}$ that are large in size, and yet have a covering set $C_{\mathcal{T}}(\epsilon)$ with small KL-diameter ϵ.

As a first step, we need to get a sense of when two graphs would have corresponding distributions with a small KL-divergence. To do so we need a general upper bound on such a KL divergence between two graphs. A simple strategy is to simply bound it by its symmetric divergence[89]. In this case, a little calculation shows :

$$\begin{aligned} D\left(f_G \| f_{G'}\right) &\leq D\left(f_G \| f_{G'}\right) + D\left(f_{G'} \| f_G\right) \\ &= \sum_{(s,t)\in E\setminus E'} \lambda\left(\mathbb{E}_G\left[x_s x_t\right] - \mathbb{E}_{G'}\left[x_s x_t\right]\right) + \sum_{(s,t)\in E'\setminus E} \lambda\left(\mathbb{E}_{G'}\left[x_s x_t\right] - \mathbb{E}_G\left[x_s x_t\right]\right) \end{aligned} \tag{6.3}$$

where E and E' are the edges in the graphs G and G' respectively, and $\mathbb{E}_G[\cdot]$ denotes the expectation under f_G. Also note that the correlation between x_s and x_t, $\mathbb{E}_G[x_s x_t] = 2P_G(x_s x_t = +1) - 1$.

From Eq. (6.3), we observe that the only pairs, (s,t), contributing to the KL-divergence are the ones that lie in the symmetric difference, $E\Delta E'$. If the number of such pairs is small, and the difference of correlations in G and G' (*i.e.* $\mathbb{E}_G\left[x_s x_t\right] - \mathbb{E}_{G'}\left[x_s x_t\right]$) for such pairs is small, then the KL-divergence would be small. In the subsequent subsection, we provide a general structural characterization that achieves such a small difference of correlations between G and G'.

6.3.3 Structural Characterization with Large Correlation

One scenario when there might be a small difference in correlations is when one of the correlations is very large, specifically arbitrarily close to 1, say $\mathbb{E}_{G'}[x_s x_t] \geq 1 - \epsilon$, for some $\epsilon > 0$. Then, $\mathbb{E}_G[x_s x_t] - \mathbb{E}_{G'}[x_s x_t] \leq \epsilon$, since $\mathbb{E}_G[x_s x_t] \leq 1$. Indeed, when s, t are part of a clique[89], this is achieved since the large number of connections between them force a higher probability of agreement *i.e.* $P_G(x_s x_t = +1)$ is large.

In this work we provide a much more general characterization of when this might happen by relying on the following key lemma that connects the presence of "many" node disjoint "short" paths between a pair of nodes in the graph to high correlation between them. We define the property formally below.

Definition 6.3.1. *Two nodes a and b in an undirected graph G are said to be (ℓ, d) connected if they have d node disjoint paths of length at most ℓ.*

Lemma 6.3.2. *Consider a graph G and a scalar $\lambda > 0$. Consider the distribution $f_G(\mathbf{x})$ induced by the graph. If a pair of nodes a and b are (ℓ, d) connected, then $\mathbb{E}_G\left[x_a x_b\right] \geq 1 - \frac{2}{1 + \frac{\left(1+(\tanh(\lambda))^\ell\right)^d}{\left(1-(\tanh(\lambda))^\ell\right)^d}}$.*

From the above lemma, we can observe that as ℓ gets smaller and d gets larger, $\mathbb{E}_G\left[x_a x_b\right]$ approaches its maximum value of 1. As an example, in a k-clique, any two vertices, s and t, are $(2, k-1)$ connected. In this case, the bound from Lemma 6.3.2 gives us: $\mathbb{E}_G\left[x_a x_b\right] \geq 1 - \frac{2}{1+(\cosh\lambda)^{k-1}}$. Of course, a clique enjoys a lot more connectivity (*i.e.* also $\left(3, \frac{k-1}{2}\right)$ connected etc.) which allows for a somewhat stronger bound of $\sim 1 - \frac{\lambda k e^{3\lambda/2}}{e^{\lambda k}}$ (see [89]). However, the bound from Lemma 6.3.2 has similar asymptotic behaviour (*i.e.* as k grows) in most regimes of λ.

Now, as discussed earlier, a high correlation between a pair of nodes contributes a small term to the KL-divergence. This is stated in the following corollary.

Corollary 6.3.2. *Consider two graphs $G(V, E)$ and $G'(V, E')$ and scalar weight $\lambda > 0$ such that $E - E'$ and $E' - E$ only contain pairs of nodes that are (ℓ, d) connected in graphs G' and G respectively, then the KL-divergence between f_G and $f_{G'}$, $D\left(f_G \| f_{G'}\right) \leq \frac{2\lambda|E \Delta E'|}{1+\frac{\left(1+(\tanh(\lambda))^{\ell}\right)^{d}}{\left(1-(\tanh(\lambda))^{\ell}\right)^{d}}}$.*

6.4 Main result: Sample complexity requirements for Erdős-Rényi random graphs

In this section, we relate the number of samples required to learn $G \sim G(p, c/p)$ for the dense case, for guaranteeing a constant average probability of error p_{avg}. We have the following main result.

Theorem 6.4.1. *Let $G \sim G(p, c/p)$, $c = \Omega(p^{3/4} + \epsilon')$, $\epsilon' > 0$. For this class of random graphs, if $p_{avg} \leq 1/90$, then $n \geq \max\left(n_1, n_2\right)$ where:*

$$n_1 = \frac{H(c/p)(3/80)\left(1 - 80p_{\text{avg}} - O(1/p)\right)}{\left(\frac{4\lambda p}{3}\exp(-\frac{p}{36}) + 4\exp(-\frac{p^{\frac{3}{2}}}{144}) + \frac{4\lambda}{9\left(1+(\cosh(2\lambda))^{\frac{c^2}{6p}}\right)}\right)}$$

$$n_2 = \frac{p}{4}H\left(\frac{c}{p}\right)(1 - 3p_{\text{avg}}) - O\left(\frac{1}{p}\right) \tag{6.4}$$

Corollary 6.4.1. *Let $G \sim G(p, c/p)$, $c = \Omega(p^{3/4+\epsilon'})$ for any $\epsilon' > 0$. Let $p_{\text{avg}} \leq 1/90$, then*

1. *$\lambda = \Omega(\sqrt{p}/c) : \Omega\left(\lambda H(c/p)(\cosh(2\lambda))^{\frac{c^2}{6p}}\right)$ samples are needed.*
2. *$\lambda < O(\sqrt{p}/c) : \Omega(c \log p)$ samples are needed. (This bound is from [90])*

Remark 6.4.1. *When $G \sim G(p, c/p)$, $c = \Omega(p^{3/4+\epsilon'})$ when $\lambda = \Omega(\sqrt{p}/c)$, a huge number (exponential in $\frac{\lambda^2 c^2}{p}$) of samples are required. Hence, for any efficient algorithm, we require $\lambda = O\left(\sqrt{p}/c\right)$ and in this regime $O\left(c \log p\right)$ samples are required to learn.*

6.4.1 Proof Outline

The proof skeleton is based on Lemma 6.3.1 and Remark 6.3.2. The essence of the proof is to cover a set of graphs $\mathcal{T}$, with large measure, by an exponentially small set $C_{\mathcal{T}}(\epsilon)$ where the KL-divergence between any covered and the covering graph is also very small ϵ. For this we use Corollary 6.3.2. The key steps in the proof are outlined below:

1. We identify a subclass of graphs $\mathcal{T}$, as in Lemma 6.3.1, whose measure is close to 1, i.e. $\mu(\mathcal{T}) = 1 - o(1)$. A natural candidate is the 'typical' set $\mathcal{T}_\epsilon^p$ which is defined to be a set of graphs each with $(\frac{cp}{2} - \frac{cp\epsilon}{2}, \frac{cp}{2} + \frac{cp\epsilon}{2})$ edges in the graph.
2. (Path property) We show that most graphs in $\mathcal{T}$ have property $\mathcal{R}$: there are $O(p^2)$ pairs of nodes such that every pair is $(2, O(\frac{c^2}{p}))$ connected (see Defn. 6.3.1) with high probability. The measure $\mu(\mathcal{R} \,|\mathcal{T}) = 1 - \delta_1$.

3. (Covering with low diameter) *Every graph G in $\mathcal{R} \bigcap \mathcal{T}$ is covered by a graph G' from a covering set $C_{\mathcal{R}}(\delta_2)$ such that their edge set differs* ***only*** *in the $O(p^2)$ nodes that are well connected.* Therefore, by Corollary 6.3.2, KL-divergence between G and G' is very small ($\delta_2 = O(\lambda p^2 \cosh(\lambda)^{-c^2/p})$).

4. (Efficient covering in Size) Further, the covering set $C_{\mathcal{R}}$ is exponentially smaller than $\mathcal{T}$.

5. (Uncovered graphs have exponentially low measure) Then we show that the uncovered graphs have large KL-divergence $\left(O(p^2\lambda)\right)$ but their measure $\mu(\mathcal{R}^c|\mathcal{T})$ is exponentially small.

6. Using a similar (but more involved) expression for probability of error as in Corollary 6.3.1, roughly we need $O(\frac{\log|T|}{\delta_1+\delta_2})$ samples.

Remark: The proof is long and technical and the reader can refer to the long arxiv version [95] of the paper [96].

6.5 Conclusion

We derived the first set of sample complexity lower bounds for learning Ising models drawn from moderately dense Erdős-Rényi ensembles. It would be interesting to extend the above treatment of finding sample complexity lower bounds by extending it to sparser Erdős-Rényi ensembles and/or tighten the bounds by considering a more complex structure.

Chapter 7

Learning Causal Graphs with Small Interventions

7.1 Introduction

In the previous chapter [1], we considered the important problem of how to learn a compact underlying network structure, i.e. a graphical model, that captures conditional independencies among different observed variables in data. Information theoretically, passively collecting more samples is sufficient since the graphical model can be obtained from distributional information. There is another important type of relationship that is desired among a set of variables. These are *causal* relationships. *Correlation does not imply causation* is an often heard maxim. The import of the maxim is that causal relationships, in general, cannot be inferred from distributional information alone. For example, consider the case of two jointly gaussian variables, (X, Y) such that $X = Y + N$ where N and Y are independent gaussians. Even full distributional information cannot resolve the question - Does Y cause X ? - in one way or the other. The reason is Y can be written as $X + N'$ for another independent random variable N' with suitable variance. Unless there is any other indirect criterion used, causal relationships cannot

[1] The material in this chapter is based on the conference paper: K. Shanmugam*, M. Kocaoglu*, A.G.Dimakis and S. Vishwanath, "Learning Causal Graphs with Small Interventions", *NIPS 2015* (*-equal contribution). The dissertation author contributed equally to all the results appearing in this chapter along with the second student author of the above paper.

be distinguished from just passively observing data or knowing functional relationships. Causal relationships can be learnt under *interventions* or *experiments* wherein a specific variable or a set of variables is forced to take a specific value with other mechanisms being undisturbed. An example in the real world is A/B testing or drug trials where effectiveness of a drug is tested by randomly forcing half the people to have placebos and half the people to have the drug and then examining the effects. There are mathematical frameworks to capture this notion of causality together with the notion of interventions.

Causality is a fundamental concept in sciences and philosophy. The mathematical formulation of a *theory of causality* in a probabilistic sense has received significant attention recently (e.g. [97–101]). A formulation advocated by Pearl considers the *structural equation models*: In this framework, X is a cause of Y, if Y can be written as $f(X,E)$, for some deterministic function f and some latent random variable E. Given two causally related variables X and Y, it is not possible to infer whether X *causes* Y or Y *causes* X from random samples, unless certain assumptions are made on the distribution of E and/or on f [102,103]. For more than two random variables, *directed acyclic graphs* (DAGs) are the most common tools used for representing causal relations. For a given DAG $D = (V,E)$, the directed edge $(X,Y) \in E$ shows that X is a cause of Y.

If we make no assumptions on the data generating process, the standard way of inferring the causal directions is by performing experiments, the so-called *interventions*. An intervention requires modifying the process that generates the random variables: The experimenter has to enforce values on the random variables. This process is different than conditioning as explained in detail in [97].

The natural problem to consider is therefore minimizing the *number* of interventions required to learn a causal DAG. Hauser *et al.* [98] developed an efficient algorithm that minimizes this number in the worst case. The algorithm is based on the optimal coloring of chordal graphs and requires at most $\log \chi$ interventions to learn any causal graph where χ is the chromatic number of the chordal skeleton.

However, one important open problem appears when one also considers the *size* of the used interventions: Each intervention is an experiment where the scientist must force a set of variables to take random values. Unfortunately, the interventions obtained in [98] can involve up to $n/2$ variables. The simultaneous enforcing of many variables can be quite challenging in many applications: for example in biology, some variables may not be enforceable at all or may require complicated genomic interventions for each parameter.

In this paper, we consider the problem of learning a causal graph when intervention sizes are bounded by some parameter k. The first work we are aware of for this problem is by Eberhardt et al. [99], where he provided an achievable scheme. Furthermore, [104] shows that the set of interventions to fully identify a causal DAG must satisfy a specific set of combinatorial conditions called *a separating system*[2], when the intervention size is not constrained or is 1. In [100], with the assumption that the same holds true for any intervention size, Hyttinen *et al.* draw connections between causality and known separating system constructions.

[2] A separating system is a 0-1 matrix with n distinct columns and each row has at most k ones.

One open problem is: If the learning algorithm is *adaptive* after each intervention, is a separating system still needed or can one do better? It was believed that adaptivity does not help in the worst case [104] and that one still needs a separating system.

Our Contributions: We obtain several novel results for learning causal graphs with interventions bounded by size k. The problem can be separated for the special case where the underlying undirected graph (the skeleton) is the complete graph and the more general case where the underlying undirected graph is chordal.

1. For complete graph skeletons, we show that any adaptive deterministic algorithm needs a (n, k) separating system. This implies that lower bounds for separating systems also hold for adaptive algorithms and resolves the previously mentioned open problem.
2. We present a novel combinatorial construction of a separating system that is close to the previous lower bound. This simple construction may be of more general interest in combinatorics.
3. Recently [101] showed that *randomized* adaptive algorithms need only $\log\log n$ interventions with high probability for the unbounded case. We extend this result and show that $O\left(\frac{n}{k}\log\log k\right)$ interventions of size bounded by k suffice with high probability.
4. We present a more general information theoretic lower bound of $\frac{n}{2k}$ to capture the performance of such randomized algorithms.
5. We extend the lower bound for adaptive algorithms for general chordal graphs. We show that over all orientations, the number of experiments from a $(\chi(G), k)$

separating system is needed where $\chi(G)$ is the chromatic number of the skeleton graph.

6. We show two extremal classes of graphs. For one of them, the interventions through (χ, k) separating system is sufficient. For the other class, we need $\frac{\alpha(\chi-1)}{2k} \approx \frac{n}{2k}$ experiments in the worst case.
7. We exploit the structural properties of chordal graphs to design a new deterministic adaptive algorithm that uses the idea of separating systems together with adaptability to Meek rules. We simulate our new algorithm and empirically observe that it performs quite close to the (χ, k) separating system for random instances. Our algorithm requires much fewer interventions compared to an (n, k) separating system. We also prove some theoretical guarantees regarding the performance of our algorithm in the worst case.

7.2 Background and Terminology

7.2.1 Essential graphs

A *causal DAG* $D = (V, E)$ is a directed acyclic graph where $V = \{x_1, x_2 \ldots x_n\}$ is a set of random variables and $(x, y) \in E$ is a directed edge if and only if x is a direct *cause* of y. We adopt Pearl's *structural equation model with independent errors* (SEM-IE) in this work (see [97] for more details). Variables in $S \subseteq V$ *cause* x_i, if $x_i = f(\{x_j\}_{j \in S}, e_y)$ where e_y is a random variable independent of all other variables.

The causal relations of D imply a set of conditional independence (CI) relations between the variables. A conditional independence relation is of the following form: Given Z, the set X and the set Y are conditionally independent for some disjoint subsets of variables X, Y, Z. Due to this, causal DAGs are also called *causal Bayesian networks.* A set V of variables is Bayesian with respect to a DAG D if the joint probability distribution of V can be factorized as a product of marginals of every variable conditioned on its parents.

All the CI relations that are learned statistically through observations can also be inferred from the Bayesian network using a graphical criterion called the *d-separation* [105] assuming that the distribution is faithful to the graph [3]. Two causal DAGs are said to be *Markov equivalent* if they encode the same set of CIs. Two causal DAGs are Markov equivalent if and only if they have the same skeleton[4] and the same immoralities[5]. The class of causal DAGs that encode the same set of CIs is called the *Markov equivalence class.* We denote the Markov equivalence class of a DAG D by $[D]$. The graph union[6] of all DAGs in $[D]$ is called the *essential graph* of D. It is denoted $\mathcal{E}(D)$. $\mathcal{E}(D)$ is always a chain graph with chordal[7] chain components [8] [107].

[3] Given Bayesian network, any CI relation implied by d-separation holds true. All the CIs implied by the distribution can be found using d-separation if the distribution is faithful. Faithfulness is a widely accepted assumption, since it is known that only a measure zero set of distributions are not faithful [106].

[4] Skeleton of a DAG is the undirected graph obtained when directed edges are converted to undirected edges.

[5] An induced subgraph on X, Y, Z is an immorality if X and Y are disconnected, $X \rightarrow Z$ and $Z \leftarrow Y$.

[6] Graph union of two DAGs $D_1 = (V, E_1)$ and $D_2 = (V, E_2)$ with the same skeleton is a partially directed graph $D = (V, E)$, where $(v_a, v_b) \in E$ is undirected if the edges (v_a, v_b) in E_1 and E_2 have different directions, and directed as $v_a \rightarrow v_b$ if the edges (v_a, v_b) in E_1 and E_2 are both directed as $v_a \rightarrow v_b$.

[7] An undirected graph is chordal if it has no induced cycle of length greater than 3.

[8] This means that $\mathcal{E}(D)$ can be decomposed as a sequence of undirected chordal graphs $G_1, G_2 \ldots G_m$ (chain components) such that there is a directed edge from a vertex in G_i to a vertex in G_j only if $i < j$

The d-separation criterion can be used to identify the skeleton and all the immoralities of the underlying causal DAG [105]. Additional edges can be identified using the fact that the underlying DAG is acyclic and there are no more immoralities. Meek derived 3 local rules (*Meek rules*), introduced in [108], to be recursively applied to identify every such additional edge (see Theorem 3 of [109]). The repeated application of *Meek rules* on this partially directed graph with identified immoralities until they can no longer be used yields the essential graph.

7.2.2 Interventions and Active Learning

Given a set of variables $V = \{x_1, \ldots, x_n\}$, an *intervention* on a set $S \subset X$ of the variables is an experiment where the performer forces each variable $s \in S$ to take the value of another independent (from other variables) variable u, i.e., $s = u$. This operation, and how it affects the joint distribution is formalized by the *do* operator by Pearl [97]. An intervention modifies the causal DAG D as follows: The post intervention DAG $D_{\{S\}}$ is obtained by removing the connections of nodes in S to their parents. The *size of an intervention* S is the number of intervened variables, i.e., $|S|$. Let S^c denote the complement of the set S.

CI-based learning algorithms can be applied to $D_{\{S\}}$ to identify the set of removed edges, i.e. parents of S [105], and the remaining adjacent edges in the original skeleton are declared to be the children. Hence,

(R0) The orientations of the edges of the cut between S and S^c in the original DAG D can be inferred.

Then, 4 local Meek rules (introduced in [108]) are repeatedly applied to the original DAG D with the new directions learnt from the cut to learn more till no more directed edges can be identified. Further application of CI-based algorithms on D will reveal no more information. The Meek rules are given below:

(R1) $(a - b)$ is oriented as $(a \rightarrow b)$ if $\exists c$ s.t. $(c \rightarrow a)$ and $(c, b) \notin E$.

(R2) $(a - b)$ is oriented as $(a \rightarrow b)$ if $\exists c$ s.t. $(a \rightarrow c)$ and $(c \rightarrow b)$.

(R3) $(a - b)$ is oriented as $(a \rightarrow b)$ if $\exists c, d$ s.t. $(a - c)$,$(a - d)$,$(c \rightarrow b)$,$(d \rightarrow b)$ and $(c, d) \notin E$.

(R4) $(a - c)$ is oriented as $(a \rightarrow c)$ if $\exists b, d$ s.t. $(b \rightarrow c)$,$(a - d)$,$(a - b)$,$(d \rightarrow b)$ and $(c, d) \notin E$.

The concepts of essential graphs and Markov equivalence classes are extended in [110] to incorporate the role of interventions: Let $\mathcal{I} = \{I_1, I_2, ..., I_m\}$, be a set of interventions and let the above process be followed after each intervention. Interventional Markov equivalence class ($\mathcal{I}$ equivalence) of a DAG is the set of DAGs that represent the same set of probability distributions obtained when the above process is applied after every intervention in $\mathcal{I}$. It is denoted by $[D]_{\mathcal{I}}$. Similar to the observational case, $\mathcal{I}$ *essential graph* of a DAG D is the graph union of all DAGs in the same $\mathcal{I}$ equivalence class; it is denoted by $\mathcal{E}_{\mathcal{I}}(D)$. We have the following sequence:

$$\begin{aligned} D &\rightarrow \text{CI learning} \rightarrow \text{Meek rules} \rightarrow \mathcal{E}(D) \rightarrow I_1 \xrightarrow{a} \text{learn by R0} \xrightarrow{b} \text{Meek rules} \\ &\rightarrow \mathcal{E}_{\{I_1\}}(D) \rightarrow I_2 \ldots \rightarrow \mathcal{E}_{\{I_1, I_2\}}(D) \ldots \end{aligned} \tag{7.1}$$

Therefore, after a set of interventions $\mathcal{I}$, the essential graph $\mathcal{E}_{\mathcal{I}}(D)$ is a graph with some oriented edges that captures all the causal relations we have discovered so far, using $\mathcal{I}$. Before any interventions happened $\mathcal{E}(D)$ captures the initially known causal directions. It is known that $\mathcal{E}_{\mathcal{I}}(D)$ is a chain graph with chordal chain components. Therefore when all the directed edges are removed, the graph becomes a set of disjoint chordal graphs.

7.2.3 Problem Definition

We are interested in the following question:

Problem 7.2.1. *Given that all interventions in $\mathcal{I}$ are of size at most $k < n/2$ variables, i.e., for each intervention I, $|I| \leq k, \forall I \in \mathcal{I}$, minimize the number of interventions $|\mathcal{I}|$ such that the partially directed graph with all directions learned so far $\mathcal{E}_{\mathcal{I}}(D) = D$.*

The question is the design of an algorithm that computes the small set of interventions $\mathcal{I}$ given $\mathcal{E}(D)$. Note, of course, that the unknown directions of the edges D are not available to the algorithm. One can view the design of $\mathcal{I}$ as an active learning process to find D from the essential graph $\mathcal{E}(D)$. $\mathcal{E}(D)$ is a chain graph with undirected chordal components and it is known that interventions on one chain components do not affect the discovery process of directed edges in the other components [111]. So we will assume that $\mathcal{E}(D)$ is undirected and a chordal graph to start with. Our notion of algorithm does not consider the time complexity (of statistical algorithms involved) of steps a and b in (7.1). Given m interventions, we only consider efficiently computing I_{m+1} using (possibly) the graph $\mathcal{E}_{\{I_1,\ldots I_m\}}$. We consider the following three classes of algorithms:

1. *Non-adaptive algorithm:* The choice of $\mathcal{I}$ is fixed prior to the discovery process.

2. *Adaptive algorithm:* At every step m, the choice of I_{m+1} is a deterministic function of $\mathcal{E}_{\{I_1,\ldots I_m\}}(D)$.
3. *Randomized adaptive algorithm:* At every step m, the choice of I_{m+1} is a random function of $\mathcal{E}_{\{I_1,\ldots I_m\}}(D)$.

The problem is different for complete graphs versus more general chordal graphs since rule R1 becomes applicable when the graph is not complete. Thus we give a separate treatment for each case. First, we provide algorithms for all three cases for learning the directions of complete graphs $\mathcal{E}(D) = K_n$ (undirected complete graph) on n vertices. Then, we generalize to chordal graph skeletons and provide a novel adaptive algorithm with upper and lower bounds on its performance.

The missing proofs of the results that follow can be found in the Appendix.

7.3 Complete Graphs

In this section, we consider the case where the skeleton we start with, i.e. $\mathcal{E}(D)$, is an undirected complete graph (denoted K_n). It is known that at any stage in (7.1) starting from $\mathcal{E}(D)$, rules R1, R3 and R4 do not apply. Further, the underlying DAG D is a directed clique. The directed clique is characterized by an ordering σ on $[1 : n]$ such that, in the subgraph induced by $\sigma(i), \sigma(i+1)\ldots\sigma(n)$, $\sigma(i)$ has no incoming edges. Let D be denoted by $\vec{K}_n(\sigma)$ for some ordering σ. Let $[1 : n]$ denote the set $\{1, 2 \ldots n\}$. We need the following results on a separating system for our first result regarding adaptive and non-adaptive algorithms for a complete graph.

7.3.1 Separating System

Definition 7.3.1. *[112, 113] An (n, k)-separating system on an n element set $[1 : n]$ is a set of subsets $\mathcal{S} = \{S_1, S_2 \ldots S_m\}$ such that $|S_i| \leq k$ and for every pair i, j there is a subset $S \in \mathcal{S}$ such that either $i \in S,\ j \notin S$ or $j \in S,\ i \notin S$. If a pair i, j satisfies the above condition with respect to $\mathcal{S}$, then $\mathcal{S}$ is said to separate the pair i, j. Here, we consider the case when $k < n/2$.*

In [112], Katona gave an (n, k)-separating system together with a lower bound on $|\mathcal{S}|$. In [113], Wegener gave a simpler argument for the lower bound and also provided a tighter upper bound than the one in [112]. In this work, we give a different construction below where the separating system size is at most$\lceil \log_{\lceil n/k \rceil} n \rceil$ larger than the construction of Wegener. However, our construction has a simpler description.

Lemma 7.3.1. *There is a labeling procedure that produces distinct ℓ length labels for all elements in $[1 : n]$ using letters from the integer alphabet $\{0, 1 \ldots a\}$ where $\ell = \lceil \log_a n \rceil$. Further, in every digit (or position), any integer letter is used at most $\lceil n/a \rceil$ times.*

Once we have a set of n string labels as in Lemma 7.3.1, our separating system construction is straightforward.

Theorem 7.3.1. *Consider an alphabet $\mathcal{A} = [0 : \lceil \frac{n}{k} \rceil]$ of size $\lceil \frac{n}{k} \rceil + 1$ where $k < n/2$. Label every element of an n element set using a distinct string of letters from $\mathcal{A}$ of length $\ell = \lceil \log_{\lceil \frac{n}{k} \rceil} n \rceil$ using the procedure in Lemma 7.3.1 with $a = \lceil \frac{n}{k} \rceil$. For every*

$1 \leq i \leq \ell$ and $1 \leq j \leq \lceil \frac{n}{k} \rceil$, choose the subset $S_{i,j}$ of vertices whose string's i-th letter is j. The set of all such subsets $\mathcal{S} = \{S_{i,j}\}$ is a k-separating system on n elements and $|\mathcal{S}| \leq (\lceil \frac{n}{k} \rceil) \lceil \log_{\lceil \frac{n}{k} \rceil} n \rceil$.

7.3.2 Adaptive algorithms: Equivalence to a Separating System

Consider any non-adaptive algorithm that designs a set of interventions $\mathcal{I}$, each of size at most k, to discover $\vec{K}_n(\sigma)$. $\mathcal{I}$ has to be a separating system in the worst case over all σ. This is already known. Now, we prove the necessity of a separating system for deterministic adaptive algorithms in the worst case.

Theorem 7.3.2. *Let there be an adaptive deterministic algorithm A that designs the set of interventions $\mathcal{I}$ such that the final graph learnt $\mathcal{E}_{\mathcal{I}}(D) = \vec{K}_n(\sigma)$ for any ground truth ordering σ starting from the initial skeleton $\mathcal{E}(D) = K_n$. Then, there exists a σ such that A designs an $\mathcal{I}$ which is a separating system.*

The theorem above is independent of the individual intervention sizes. Therefore, we have the following theorem, which is a direct corollary of Theorem 7.3.2:

Theorem 7.3.3. *In the worst case over σ, any adaptive or a non-adaptive deterministic algorithm on the DAG $\vec{K}_n(\sigma)$ has to be such that $\frac{n}{k} \log_{\frac{ne}{k}} n \leq |\mathcal{I}|$. There is a feasible $\mathcal{I}$ with $|\mathcal{I}| \leq \lceil (\frac{n}{k} \rceil - 1) \lceil \log_{\lceil \frac{n}{k} \rceil} n \rceil$*

Proof. By Theorem 7.3.2, we need a separating system in the worst case and the lower and upper bounds are from [112, 113]. □

7.3.3 Randomized Adaptive Algorithms

In this section, we show that that total number of variable accesses to fully identify the complete causal DAG is $\Omega(n)$.

Theorem 7.3.4. *To fully identify a complete causal DAG $\vec{K}_n(\sigma)$ on n variables using size-k interventions, $\frac{n}{2k}$ interventions are necessary. Also, the total number of variables accessed is at least $\frac{n}{2}$.*

The lower bound in Theorem 7.3.4 is information theoretic. We now give a randomized algorithm that requires $O(\frac{n}{k} \log\log k)$ experiments in expectation. We provide a straightforward generalization of [101], where the authors gave a randomized algorithm for unbounded intervention size.

Theorem 7.3.5. *Let $\mathcal{E}(D)$ be K_n and the experiment size $k = n^r$ for some $0 < r < 1$. Then there exists a randomized adaptive algorithm which designs an $\mathcal{I}$ such that $\mathcal{E}_{\mathcal{I}}(D) = D$ with probability polynomial in n, and $|\mathcal{I}| = \mathcal{O}(\frac{n}{k} \log\log(k))$ in expectation.*

7.4 General Chordal Graphs

In this section, we turn to interventions on a general DAG G. After the initial stages in (7.1), $\mathcal{E}(G)$ is a chain graph with chordal chain components. There are no further immoralities throughout the graph. In this work, we focus on one of the chordal chain components. Thus the DAG D we work on is assumed to be a directed graph with no immoralities and whose skeleton $\mathcal{E}(D)$ is chordal. We are interested in recovering D from $\mathcal{E}(D)$ using interventions of size at most k following (7.1).

7.4.1 Bounds for Chordal skeletons

We provide a lower bound for both adaptive and non-adaptive deterministic schemes for a chordal skeleton $\mathcal{E}(D)$. Let $\chi\left(\mathcal{E}(D)\right)$ be the coloring number of the given chordal graph. Since, chordal graphs are perfect, it is the same as the clique number.

Theorem 7.4.1. *Given a chordal $\mathcal{E}(D)$, in the worst case over all DAGs D (which has skeleton $\mathcal{E}(D)$ and no immoralities), if every intervention is of size at most k, then*

$$|\mathcal{I}| \geq \frac{\chi\left(\mathcal{E}(D)\right)}{k} \log_{\frac{\chi(\mathcal{E}(D))e}{k}} \chi\left(\mathcal{E}(D)\right) \tag{7.2}$$

for any adaptive and non-adaptive algorithm with $\mathcal{E}_{\mathcal{I}}(D) = D$.

Upper bound: Clearly, the separating system based algorithm of Section 7.3 can be applied to the vertices in the chordal skeleton $\mathcal{E}(D)$ and it is possible to find all the directions. Thus, $|\mathcal{I}| \leq \frac{n}{k} \log_{\lceil \frac{n}{k} \rceil} n \leq \frac{\alpha(\mathcal{E}(D))\chi(\mathcal{E}(D))}{k} \log_{\lceil \frac{n}{k} \rceil} n$. This with the lower bound implies an α approximation algorithm (since $\log_{\lceil \frac{n}{k} \rceil} n \leq \log_{\frac{\chi(\mathcal{E}(D))e}{k}} \chi\left(\mathcal{E}(D)\right)$, under a mild assumption $\chi(\mathcal{E}(D)) \leq \frac{n}{e}$).

Remark: The separating system on n nodes gives an α approximation. However, the new algorithm in Section 7.4.3 exploits chordality and performs much better empirically. It is possible to show that our heuristic also has an α approximation guarantee but we skip that.

7.4.2 Two extreme counter examples

We provide two classes of chordal skeletons G: One for which the number of interventions close to the lower bound is *sufficient* and the other for which the number of interventions *needed* is very close to the upper bound.

Theorem 7.4.2. *There exists chordal skeletons such that for any algorithm with intervention size constraint k, the number of interventions $|\mathcal{I}|$ required is at least $\alpha \frac{(\chi-1)}{2k}$ where α and χ are the independence number and chromatic numbers respectively. There exists chordal graph classes such that $|\mathcal{I}| = \lceil \frac{\chi}{k} \rceil \lceil \log_{\lceil \frac{\chi}{k} \rceil} \chi \rceil$ is sufficient.*

7.4.3 An Improved Algorithm using Meek Rules

In this section, we design an adaptive deterministic algorithm that *anticipates* Meek rule R1 usage along with the idea of a separating system. We evaluate this experimentally on random chordal graphs. First, we make a few observations on learning connected directed trees T from the skeleton $\mathcal{E}(T)$ (undirected trees are chordal) that do not have immoralities using Meek rule R1 where every intervention is of size $k = 1$. Because the tree has no cycle, Meek rules R2-R4 do not apply.

Lemma 7.4.1. *Every node in a directed tree with no immoralities has at most one incoming edge. There is a root node with no incoming edges and intervening on that node alone identifies the whole tree using repeated application of rule $R1$.*

Lemma 7.4.2. *If every intervention in $\mathcal{I}$ is of size at most 1, learning all directions on a directed tree T with no immoralities can be done adaptively with at most $|\mathcal{I}| \leq$*

$O(\log_2 n)$ where n is the number of vertices in the tree. The algorithm runs in time $\mathrm{poly}(n)$.

Lemma 7.4.3. *Given any chordal graph and a valid coloring, the graph induced by any two color classes is a forest.*

In the next section, we combine the above single intervention adaptive algorithm on directed trees which uses Meek rules, with that of the non-adaptive separating system approach.

7.4.3.1 Description of the algorithm

The key motivation behind the algorithm is that, a pair of color classes is a forest (Lemma 7.4.3). Choosing the right node to intervene leaves only a small subtree unlearnt as in the proof of Lemma 7.4.2. In subsequent steps, suitable nodes in the remaining subtrees could be chosen until all edges are learnt. We give a brief description of the algorithm below.

Let G denote the initial undirected chordal skeleton $\mathcal{E}(D)$ and let χ be its coloring number. Consider a (χ, k) separating system $\mathcal{S} = \{S_i\}$. To intervene on the actual graph, an intervention set I_i corresponding to S_i is chosen. We would like to intervene on a node of color $c \in S_i$.

Consider a node v of color c. Now, we attach a score $P(v, c)$ as follows. For any color $c' \notin S_i$, consider the induced forest $F(c, c')$ on the color classes c and c' in G. Consider the tree $T(v, c, c')$ containing node v in F. Let $d(v)$ be the degree of v in T. Let $T_1, T_2, \ldots T_{d(v)}$ be the resulting disjoint trees after node v is removed from T. If v is intervened on, according to the proof of Lemma 7.4.2: a) All edge directions in all trees T_i except one of them would be learnt when applying Meek Rules and rule R0. b) All the directions from v to all its neighbors would be found.

The score is taken to be the total number of edge directions guaranteed to be learnt in the worst case. Therefore, the score $P(v)$ is:

$$P(v) = \sum_{c' \in S_i^c} \left(|T(c, c')| - \max_{1 \leq j \leq d(v)} |T_j| \right). \tag{7.3}$$

The node with the highest score among the color class c is used for the intervention I_i. After intervening on I_i, all the edges whose directions are known through Meek Rules (by repeated application till nothing more can be learnt) and R0 are deleted from G. Once $\mathcal{S}$ is processed, we recolor the sparser graph G. We find a new $\mathcal{S}$ with the new chromatic number on G and the above procedure is repeated. The exact hybrid algorithm is described in Algorithm 6.

Theorem 7.4.3. *Given an undirected choral skeleton G of an underlying directed graph with no immoralities, Algorithm 6 ends in time polynomial in n and it returns the correct underlying directed graph.*

We prove an approximation guarantee on the performance of a mild variation of Algorithm 6. We need the definition a completely separating system.

Definition 7.4.1. *[114] An (n, k)-completely separating system on an n element set $[1 : n]$ is a set of subsets $\mathcal{S} = \{S_1, S_2 \ldots S_m\}$ such that $|S_i| \leq k$ and for every pair i, j, there is a subset $S_1 \in \mathcal{S} : i \in S_1,\ j \notin S_1$ and there is a subset $S_2 \in \mathcal{S} : j \in S_2,\ i \notin S_2$. If a pair i, j satisfies the above condition with respect to $\mathcal{S}$, then $\mathcal{S}$ is said to separate the pair i, j. Here, we consider the case when $k < n/2$.*

Theorem 7.4.4. *Let G be the undirected chordal skeleton of an underlying directed graph with no immoralities. Consider an optimal coloring* $\psi : V \to [1 : \chi]$ *of the chordal skeleton G where the total number of colors in the coloring is* χ*.* χ *is the coloring number of the chordal skeleton G. Let C be the maximum number of components and T be the size of the maximum component in any induced sub-graph between two color classes in the coloring* ψ*. Then, Algorithm 6, initialized with coloring* ψ *and a* $(\chi, \min(k, \lceil \chi/2 \rceil))$ *completely separating system of size* $R\left(\chi, \min(k, \lceil \chi/2 \rceil)\right)$*, needs* $R\left(\chi, \min(k, \lceil \chi/2 \rceil)\right) * \chi * C \log T$ *size-k interventions to learn the underlying directed causal graph with skeleton G.*

We now show that a small modification of the construction in Theorem 7.3.1 gives an (n, k) completely separating system of size $\left(\lceil \frac{n}{k} \rceil + 1\right) \lceil \log_{\lceil \frac{n}{k} \rceil} n \rceil$.

Theorem 7.4.5. *Consider an alphabet* $\mathcal{A} = [0 : \lceil \frac{n}{k} \rceil]$ *of size* $\lceil \frac{n}{k} \rceil + 1$ *where* $k < n/2$*. Label every element of an n element set using a distinct string of letters from* $\mathcal{A}$ *of length* $\ell = \lceil \log_{\lceil \frac{n}{k} \rceil} n \rceil$ *using the procedure in Lemma 7.3.1 with* $a = \lceil \frac{n}{k} \rceil$*. For every* $1 \leq i \leq \ell$ *and* $0 \leq j \leq \lceil \frac{n}{k} \rceil$*, choose the subset* $S_{i,j}$ *of vertices whose string's i-th letter is j. The set of all such subsets* $\mathcal{S} = \{S_{i,j}\}$ *is a k-separating system on n elements and* $|\mathcal{S}| \leq (\lceil \frac{n}{k} \rceil + 1) \lceil \log_{\lceil \frac{n}{k} \rceil} n \rceil$*.*

Remark: Since an (n, k) completely separating system's size is close to the optimal separating system's size (upto additive logarithmic factors), Theorem 7.4.4 guarantees that algorithm 6 provides a $C\chi \log T$ approximation to the optimum number of interventions. The dependence on C is optimal since in the classes of extreme examples that requires at least $\alpha\frac{(\chi-1)}{2k}$ interventions in Theorem 7.4.2, $C = \alpha$ and $T = 2$.

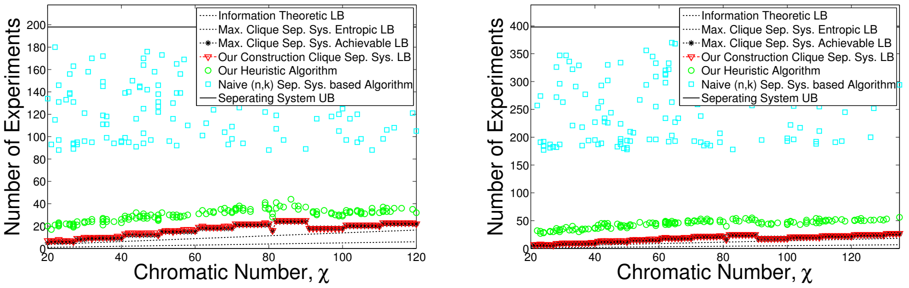
Algorithm 6 Hybrid Algorithm using Meek rules with separating system

```
Input: Chordal Graph skeleton $G = (V, E)$ with no Immoralities.
Initialize $\vec{G}(V, E_d = \emptyset)$ with $n$ nodes and no directed edges. Initialize time $t = 1$.
while $E \neq \emptyset$ do
    Color the chordal graph $G$ with $\chi$ colors.    ▷ Standard algorithms exist to do it in linear time
    Initialize color set $\mathcal{C} = \{1, 2 \dots \chi\}$. Form a $(\chi, \min(k, \lceil \chi/2 \rceil))$ separating system $\mathcal{S}$ such that $|S| \leq k, \ \forall S \in \mathcal{S}$.
    for $i = 1$ until $|\mathcal{S}|$ do
        Initialize Intervention $I_t = \emptyset$.
        for $c \in S_i$ and every node $v$ in color class $c$ do
            Consider $F(c, c')$, $T(c, c', v)$ and $\{T_j\}_1^{d(i)}$(as per definitions in Sec. 7.4.3.1).
            Compute: $P(v, c) = \sum\limits_{c' \in \mathcal{C} \bigcap S_i^c} |T(c, c', v)| - \max\limits_{1 \leq j \leq d(i)} |T_j|$.
        end for
        if $k \leq \chi/2$ then
            $I_t = I_t \bigcup\limits_{c \in S_i} \{\underset{v:P(v,c) \neq 0}{\text{argmax}} \, P(v, c)\}$.
        else
            $I_t = I_t \cup_{c \in S_i} \{\text{First } \frac{k}{\lceil \chi/2 \rceil} \text{ nodes } v \text{ with largest nonzero } P(v, c)\}$.
        end if
        $t = t + 1$
        Apply R0 and Meek rules using $E_d$ and $E$ after intervention $I_t$. Add newly learnt directed edges to $E_d$ and delete them from $E$.
    end for
    Remove all nodes which have degree 0 in G.
end while
return $\vec{G}$.
```

7.5 Simulations

Although the theoretical results in the previous section is for Algorithm 6 supplied with a completely separating system, we present numerical results for the exact algorithm 6 provided. We also notice that numerically the algorithm performs better than the approximation guarantee, which seems to have an extra factor of χ that does not show up in our numerical simulations. We simulate our new heuristic, namely Algorithm 6, on randomly generated chordal graphs and compare it with a naive algorithm that follows the intervention sets given by our (n, k) separating system as in Theorem 7.3.1. Both algorithms apply R0 and Meek rules after each intervention according to (7.1). We plot the following lower bounds: a) *Information Theoretic LB* of $\frac{\chi}{2k}$ b) *Max. Clique Sep. Sys. Entropic LB* which is the chromatic number based lower bound of Theorem 7.4.1. Moreover, we use two known (χ, k) separating system constructions for the maximum clique size as "references": The best known (χ, k) separating system is shown by the label *Max. Clique Sep. Sys. Achievable LB* and our new simpler separating system construction (Theorem 7.3.1) is shown by *Our Construction Clique Sep. Sys. LB.* As an upper bound, we use the size of the best known (n, k) separating system (without any Meek rules) and is denoted *Separating System UB.*

Random generation of chordal graphs: Start with a random ordering σ on the vertices. Consider every vertex starting from $\sigma(n)$. For each vertex i, $(j, i) \in E$ with probability inversely proportional to $\sigma(i)$ for every $j \in S_i$ where $S_i = \{v : \sigma^{-1}(v) < \sigma^{-1}(i)\}$. The proportionality constant is changed to adjust sparsity of the graph. After all such j are considered, make $S_i \cap \text{ne}(i)$ a clique by adding edges respecting the ordering σ, where ne(i) is the neighborhood of i. The resultant graph is a DAG and the corresponding skeleton is chordal. Also, σ is a perfect elimination ordering.



(a) $n = 1000, k = 10$

(b) $n = 2000, k = 10$

Figure 7.1: n: no. of vertices, k: Intervention size bound. The number of experiments is compared between our heuristic and the naive algorithm based on the (n, k) separating system on random chordal graphs. The red markers represent the sizes of (χ, k) separating system. Green circle markers and the cyan square markers for the same χ value correspond to the number of experiments required by our heuristic and the algorithm based on an (n, k) separating system(Theorem 7.3.1), respectively, on the same set of chordal graphs. Note that, when $n = 1000$ and $n = 2000$, the naive algorithm requires on average about 130 and 260 (close to n/k) experiments respectively, while our algorithm requires at most ~ 40 (orderwise close to $\chi/k = 10$) when $\chi = 100$.

Results: We are interested in comparing our algorithm and the naive one which depends on the (n, k) separating system to the size of the (χ, k) separating system. The size of the (χ, k) separating system is roughly $\tilde{O}(\chi/k)$. Consider values around $\chi = 100$ on the x-axis for the plots with $n = 1000, k = 10$ and $n = 2000, k = 10$. Note that, our algorithm performs very close to the size of the (χ, k) separating system, i.e. $\tilde{O}(\chi/k)$. In fact, it is always < 40 in both cases while the average performance of naive algorithm goes from 130 (close to $n/k = 100$)

to 260 (close to $n/k = 200$). The result points to this: For random chordal graphs, the structured tree search allows us to learn the edges in a number of experiments quite close to the lower bound based only on the maximum clique size and not n. The plots for $(n, k) = (500, 10)$ and $(n, k) = (2000, 20)$ are given in Appendix.

7.6 Conclusion

We considered the problem of adaptively designing the minimum number of size-k interventions to learn a causal network on several variables under the Pearl's structural equation model for causality. We showed that non-adaptive separating system based constructions provide optimal number of interventions upto logarithmic factors for a complete graph skeleton. For general graphs, we derive lower bounds for any adaptive algorithm. Further, we also characterize extreme examples where the number of interventions are very close and very far from the lower bound. We also propose a new algorithm for general graphs that works empirically very well on random examples. Further, we also provide some theoretical guarantees for such an algorithm. An interesting future direction would be to tighten the approximation guarantees for the algorithm given in this chapter for general graphs or design a better algorithm.

Appendices

Appendix A

Proofs for Chapter 2

A.1 Algorithm: Pipage Rounding

A.2 Basic Definitions

Matroids: Matroids are structures that generalize the concept of independence from linear algebra, to general sets. Informally, we need a finite ground set S and a matroid is a way to label some subsets of S as "independent". In vector spaces, the ground set is a set of vectors, and subsets are called independent if their vectors are linearly independent, in the usual linear algebraic sense. Formally, we have [115]:

Definition A.2.1. *A matroid $\mathcal{M}$ is a tuple $\mathcal{M} = (S, \mathcal{I})$, where S is a finite ground set and $\mathcal{I} \subseteq 2^S$ (the power set of S) is a collection of independent sets, such that:*

1. $\mathcal{I}$ is nonempty, in particular, $\emptyset \in \mathcal{I}$,

2. $\mathcal{I}$ is downward closed; i.e., if $Y \in \mathcal{I}$ and $X \subseteq Y$, then $X \in \mathcal{I}$,

3. If $X, Y \in \mathcal{I}$, and $|X| < |Y|$, then $\exists y \in Y \backslash X$ such that $X \cup \{y\} \in \mathcal{I}$. □

One example is the partition matroid. In a partition matroid, the ground set S is partitioned into (disjoint) sets $S_1; S_2; ...; S_l$ and

$$\mathcal{I} = \{X \subseteq S : |X \cap S_i| \leq k_i \text{ for all } i = 1 \dots l\}, \tag{A.1}$$

Algorithm 7 PipageRounding($G, \phi, \{p(y)\}, \mathbf{R}$)

```
Input: $\mathbf{R}$. Output: $\bar{\mathbf{X}}$.
Initialize: $\hat{\mathbf{X}} = \mathbf{R}$
while $\hat{\mathbf{X}}$ is not integral do
    Construct the subgraph $H_{\hat{x}}$ of $G$ which contains all vertices but the edge set $E_{\hat{x}}$ is such that $(a,b) \in E_{\hat{x}}$ only if $\hat{x}_{a,b}$ is not integral
    if $H_{\hat{x}}$ has cycles then
        Set $\alpha$ to be one such simple cycle (each node occurs at most once in a simple cycle except the first node)
    else
        Set $\alpha$ to be a simple path with end points being nodes of degree 1 (In a bi-partite cycle-free graph, such a path exists).
    end if
    Decompose $\alpha = M_1 \bigcup M_2$ (union of two disjoint matchings) where $M_1$ and $M_2$ are matchings.
    Define parameterized solution $\mathbf{X}(\epsilon, \alpha)$ as follows:
    If $(a,b) \notin \alpha$, $x_{a,b}(\epsilon, \alpha) = \hat{x}_{a,b}$.
    Let $\epsilon_1 = \min\{\min_{(a,b) \in M_1} \hat{x}_{a,b}, \min_{(a,b) \in M_2} (1 - \hat{x}_{a,b})\}$.
    Let $\epsilon_2 = \min\{\min_{(a,b) \in M_2} \hat{x}_{a,b}, \min_{(a,b) \in M_1} (1 - \hat{x}_{a,b})\}$.
    $x_{a,b}(\epsilon, \alpha) = \hat{x}_{a,b} + \epsilon$, $\forall (a,b) \in M_1$ and $x_{a,b}(\epsilon, \alpha) = \hat{x}_{a,b} - \epsilon$, $\forall (a,b) \in M_2$. $\epsilon \in [-\epsilon_1, \epsilon_2]$.
    if $\phi(\mathbf{X}(\epsilon_2, \alpha)) > \phi(\mathbf{X}(-\epsilon_1, \alpha))$ then
        $\hat{\mathbf{X}} = \mathbf{X}(\epsilon_2, \alpha)$
    else
        $\hat{\mathbf{X}} = \mathbf{X}(-\epsilon_1, \alpha)$
    end if
end while
return $\bar{\mathbf{X}} = \hat{\mathbf{X}}$.
```

for some given parameters $k_1, k_2, ..., k_l$.

Submodular functions: Let S be a finite ground set. A set function $f : 2^S \rightarrow \mathbb{R}$ is submodular if for all sets $A, B \subseteq S$,

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B). \tag{A.2}$$

Equivalently, submodularity can be defined by the following condition. Let $f_A(i) = f(A + i) - f(A)$ denote the marginal value of an element $i \in S$ with respect to a subset $A \subseteq S$. Then, f is submodular if for all $A \subseteq B \subseteq S$ and for all $i \in S \backslash B$ we have:

$$f_A(i) \geq f_B(i). \tag{A.3}$$

Intuitively, submodular functions capture the concept of diminishing returns: as the set becomes larger the benefit of adding a new element to the set will decrease. Submodular functions can also be regarded as functions on the boolean hypercube $\{0, 1\}^{|S|} \rightarrow \mathbb{R}$. Every set has an equivalent boolean representation by assigning 1 to the elements in the set and 0 to other ones. We denote the boolean representation of a set X by a vector $X^b \in \{0, 1\}^{|S|}$.

The function f is monotone if for $A \subseteq B \subseteq S$, we have $f(A) \leq f(B)$.

A.3 Proof of Theorem 2.3.1

Consider an oracle that can solve any instance HLP$(\mathcal{G}, \mathcal{F}, P, \Omega, M, Q)$ in unit time. Then solving an instance of 2DSC(G) is equivalent to solving HLP$(\mathcal{G}, \mathcal{F}, \mathcal{P},$

$\Omega, M, Q)$ for $\mathcal{G} = G$, $\mathcal{F} = \{1, 2\}$, $\mathcal{P} = \left(\frac{1}{1+\epsilon}, \frac{\epsilon}{1+\epsilon}\right)$, $\Omega = \{1, 1, \dots, 1\}$, $M = 1$ and $Q = U = |\mathcal{U}|$.

In order to see this, notice that for any user u we have that $\mathcal{A}_u$ can be $\emptyset, \{1\}, \{2\}, \{1, 2\}$. Notice also that the value of any user u is $\sum_{f \in \mathcal{A}_u} P_f \leq 1$ and it is exactly equal to 1 only if $\mathcal{A}_u = \{1, 2\}$. However, since the cache constraint for all helpers is equal to 1, any helper can cache either file 1 or file 2 or none. It follows that in order to have the objective function value equal to U the value of all users must be equal to 1, i.e., $\mathcal{A}_u = \{1, 2\}$ for all u, which implies that each user u has at least one neighboring helper containing file 1 and another neighboring helper containing file 2. Letting B_1 and B_2 denote the (disjoint) sets of helpers containing file 1 and file 2, respectively, we conclude that determining whether the objective function (2.4) is equal to U is equivalent to determining the existence of B_1 and B_2 forming a 2-disjoint cover (see Fig. A.1).

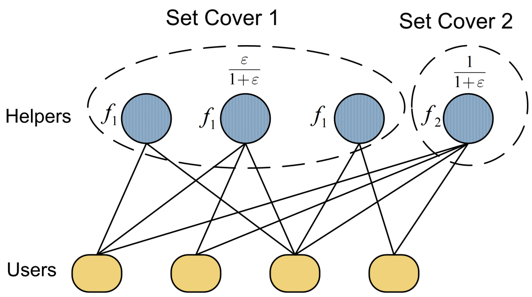

Set Cover 1
Set Cover 2
$\frac{\varepsilon}{1+\varepsilon}$
$\frac{1}{1+\varepsilon}$
Helpers
f_1
f_1
f_1
f_2
Users

Figure A.1: Figure illustrating the reduction from 2-Disjoint Set Cover Problem.

A.4 Proof of Lemma 2.3.1

Notice that the non-zero elements of the h-th column of $\mathbf{X}$ correspond to the elements in $X \cap S_h$. Hence, the constraints on the cache capacity of helpers can be expressed as $X \subseteq \mathcal{I}$, where

$$\mathcal{I} = \{X \subseteq S : |X \cap S_h| \leq M, \ \ \forall\, h = 1, \ldots, H\}. \tag{A.4}$$

Comparing $\mathcal{I}$ in (A.4) and the definition of the partition matroid in (A.1), we can see that our constraints form a partition matroid with $l = H$ and $k_i = M$, for $i = 1, ..., H$. The partition matroid is denoted by $\mathcal{M} = (S, \mathcal{I})$.

A.5 Proof of Lemma 2.3.2

Monotonicity is obvious since any new placement of a file cannot decrease the value of the objective function. In order to show submodularity, we observe that since the sum of submodular functions is submodular, it is enough to prove that for a user u the set function $G_u(X) \triangleq \omega_{0,u} - \bar{D}_u$ is submodular. We show that the marginal value of adding a new file to an arbitrary helper $h \in \mathcal{H}(u)$ decreases as the placement set X becomes larger. The marginal value of adding a new element to a placement set X is the amount of increase in $G_u(X)$ due to the addition.

Let's consider two placement sets X and X' where $X \subset X' \subset S$. For some $1 \leq i \leq |\mathcal{H}(u)| - 1$, consider adding the element $s_f^{(i)_u} \in S \backslash X'$ to both placement sets. This corresponds to adding file f in the cache of helper $(i)_u$, where such file is not placed anywhere neither in placement X nor in placement X'. We distinguish the following cases.

1) According to placement X', user u gets file f from helper $(j')_u$ with $j' < i$, i.e., $s_f^{(j')_u} \in X'$. In this case, it is immediate to see that $G_u(X' \cup \{s_f^{(i)_u}\}) - G_u(X') = 0$ (the marginal value is zero). According to placement X, user u gets file f from some helper $(j)_u$ with $j \geq j'$. If $j < i$, again the marginal value will be zero. However, when $j > i$, the marginal value is given by $G_u(X \cup \{s_f^{(i)_u}\}) - G_u(X) = P_f(\omega_{(j)_u,u} - \omega_{(i)_u,u}) > 0$.

2) According to placement X', user u gets file f through helper $(j')_u$ with $j' > i$. Hence, the marginal value is given by $G_u(X' \cup \{s_f^{(i)_u}\}) - G_u(X') = P_f(\omega_{(j')_u,u} - \omega_{(i)_u,u})$. Since in the placement set X, user u downloads the file from helper $(j)_u$ with $j \geq j'$, the resulting marginal value is $G_u(X \cup \{s_f^{(i)_u}\}) - G_u(X) = P_f(\omega_{(j)_u,u} - \omega_{(i)_u,u})$. The difference of marginal values is given by:

$$G_u(X \cup \{s_f^{(i)_u}\}) - G_u(X) - \Big(G_u(X' \cup \{s_f^{(i)_u}\})$$
$$-G_u(X')) = P_f(\omega_{(j)_u,u} - \omega_{(j')_u,u}) \geq 0.$$

Hence, the lemma is proved.

A.6 Proof of Theorem 2.3.2

We prove the main theorem by recalling some results about pipage rounding and carefully applying it to this current problem. We recall the definition of a *matching* used in Algorithm 7. A matching in an undirected graph is a subset of edges such that no two edges in the subset have a common vertex. Algorithm 7 runs in at most $|E|$ steps (a step being the outer while loop). Each step runs in time polynomial in $|E|$. The final solution $\bar{\mathbf{X}}$ is integral and it satisfies the constraints in program (2.7)–(2.10). We refer the reader to [34] for the proof of correctness and the time complexity analysis for pipage rounding. Now, we have following sufficient conditions for a $C-$ approximate polynomial time algorithm that uses pipage rounding.

Theorem A.6.1. *[34] Consider the problem (2.7) – (2.10) and suppose that $\phi(\cdot)$ satisfies the following conditions:*

1. *There exists another objective function $L(\cdot)$ such that $\forall$ $\mathbf{X} \in \{0,1\}^{|A|\times|B|}$, $L(\mathbf{X}) = \phi(\mathbf{X})$.*

2. *(Lower Bound condition) For $\mathbf{R} \in \mathbb{R}_+^{|A|\times|B|}$, $\phi(\mathbf{R}) \geq CL(\mathbf{R})$ for some constant $C < 1$.*

3. *(ϵ-convexity condition) For all feasible $\mathbf{R}$ in (2.7) – (2.10) and for all possible cycles and paths α that occur in pipage rounding , $\phi(\mathbf{X}(\epsilon,\alpha))$ is convex with respect to ϵ in the range $\epsilon \in [-\epsilon_1, \epsilon_2]$, where $\mathbf{X}(\epsilon,\alpha), \epsilon_1, \epsilon_2$ are intermediate values that occur in every iteration as defined in Algorithm 7.*

Let the optimum of the maximization $\max\ L(\mathbf{R})$ *subject to (2.8), (2.9), (2.10), be $\mathbf{R}_{opt}$. Let $\mathbf{X}_{int}$ be the integral output of* $\text{PipageRounding}\left(G, \phi, \{p(u) : u \in \mathcal{A} \cup \mathcal{B}\}, \mathbf{R}_{opt}\right)$ *(see Algorithm 7). Then, $\phi(\mathbf{X}_{int}) \geq C\phi\left(\mathbf{X}_{\text{opt}}\right)$ where $\mathbf{X}_{\text{opt}}$ is the optimum solution to the integer version of program (2.7) – (2.10), obtained by replacing $\mathbf{R}$ with the binary matrix $\mathbf{X}$.*

Proof. At the end of the inner while loop of Algorithm 7, one of the end points of the curve $\mathbf{X}(\epsilon,\alpha)$ is chosen as the improved solution. The next iteration proceeds with this improved solution. For $\epsilon = 0 \in [-\epsilon_1, \epsilon_2]$, $\mathbf{X}(0, \alpha) = \hat{\mathbf{X}}$ where $\hat{\mathbf{X}}$ is the solution from the previous iteration. If $\phi(\mathbf{X}(\epsilon, \alpha))$ is convex in ϵ (ϵ- convexity condition), then the maximum is attained at the end points. Therefore, $\max\{\phi(\mathbf{X}(-\epsilon_1, \alpha)), \phi(\mathbf{X}(\epsilon_2, \alpha))\} \geq \phi\left(\hat{\mathbf{X}}\right)$. Hence, the solution at the end of the inner while loop is no worse than the solution at the beginning. Therefore, if $\mathbf{R}_{opt}$ is the input to the pipage algorithm and $\mathbf{X}_{int}$ is the output, then we have the following chain of inequalities:

$$\begin{aligned} \phi(\mathbf{X}_{int}) &\overset{(a)}{\geq} \phi\left(\mathbf{R}_{opt}\right) \\ &\overset{(b)}{\geq} CL\left(\mathbf{R}_{opt}\right) \overset{(c)}{\geq} CL\left(\mathbf{X}_{opt}\right) \overset{(d)}{\geq} C\phi\left(\mathbf{X}_{opt}\right) \end{aligned}$$

Justification for the above inequalities are: (a) pipage rounding with ϵ-convexity condition ;(b) Lower Bound condition ;(c) $\mathbf{R}_{opt}$ is obtained when $L(\cdot)$ is optimized over the reals (relaxed version of program (2.7)–(2.10) with L as the objective) ;(d) Condition 1 in the theorem. □

In our case, it is easy to particularize the general template program (2.7) – (2.10) to the program at hand (2.3), by letting $\phi(\cdot) = g(\cdot)$, defined in (2.6), by identifying the graph G with the complete bipartite graph $\mathcal{K}_{\mathcal{F},\mathcal{H}}$ formed by the vertices $\mathcal{F}$, $\mathcal{H}$ and all possible edges connecting the elements of $\mathcal{F}$ (files) with the elements of $\mathcal{H}$ (helpers), and the edge node constraints as $p(h) = M$ for all $h \in \mathcal{H}$ and $p(f) = H$ for all $f \in \mathcal{F}$. Notice that, any feasible placement graph $\widetilde{\mathcal{G}}$ is a subgraph of this complete bipartite graph, and that letting $p(f) = H$ makes the set of constraints (2.9) irrelevant.

Now, we design a suitable $L(.)$ function such that $\phi(\mathbf{X}) = L(\mathbf{X})\ \forall\ \mathbf{X} \in \{0,1\}^{|\mathcal{F}|\times|\mathcal{H}|}$. Let $L = \sum_{f,u} P_f \widetilde{\omega}_u L_{f,u}(\mathbf{X})$ and $L_{f,u}(\mathbf{X}) = \min\{1, \sum_{h\in\mathcal{H}(u):h\neq 0} x_{f,h}\}$. This establishes condition 1 of Theorem A.6.1.

Also, the following is true from results in [34].

Lemma A.6.1. *$g(\mathbf{X}) \geq \left(1-(1-1/d)^d\right) L(\mathbf{X})$ where $d = \max_u |\mathcal{H}(u)| - 1$.*

Proof. It has been shown in [34] that for $0 \leq y_i \leq 1$,

$$1 - \prod_{k=1}^{d} (1 - y_k) \geq (1-1/d)^d \min\{1, \sum_k y_k\}. \tag{A.5}$$

Applying this to all $g_{f,u}(\mathbf{X})$ and observing that $(1-(1-1/d)^d)$ is decreasing in d, we prove the result in the above lemma. □

This establishes the lower bound condition in Theorem A.6.1. Finally, we show the ϵ-convexity condition for $g(\cdot)$.

Lemma A.6.2. *$g(\mathbf{X}(\epsilon,\alpha))$ is convex for intermediate $\mathbf{X}(\epsilon,\alpha)$, ϵ and all possible paths/cycles α that occur in Algorithm 7.*

Proof. It is sufficient to show that $g_{f,u}(\cdot)$ is convex in ϵ since $\widetilde{\omega}_u$ and P_f are non negative. Observe that only the variables $x_{f,h}$, for a particular f, are involved in the expression for $g_{f,u}(\cdot)$. The edges (f,h) are all incident on a particular vertex f. We refer the reader to Algorithm 7 for the definition of the variables used in the proof with (f,h) replacing the edges (a,b). Briefly, $\hat{\mathbf{X}}$ is the current solution at the beginning of any iteration and $\mathbf{X}(\epsilon,\alpha)$ is the parametrized solution associated with $\hat{\mathbf{X}}$ during the iteration. Only the variables $\hat{x}_{f,h}$, corresponding to edges participating in α, are changed in the iteration. Since α is either a simple cycle or a simple path, at most two of the variables $x_{f,h}(\epsilon,\alpha)$ are different from $\hat{x}_{f,h}$(by either adding or subtracting ϵ) in any iteration for a given f. Also, variables corresponding to one matching are increased and the ones corresponding to the other are decreased. Therefore, if only one variable is changed, then the expression is linear in ϵ and hence it is convex. If two variables are changed (say $\hat{x}_{f,h_1}$ and $\hat{x}_{f,h_2}$), then either they are changed to $\hat{x}_{f,h_1}+\epsilon$, $\hat{x}_{f,h_2}-\epsilon$ or to $x_{f,h_1}-\epsilon$, $x_{f,h_2}+\epsilon$. Without loss of generality, assuming one of the cases we have, $g_{f,u}\left(\mathbf{X}(\epsilon,\alpha)\right) = 1-\left(1-\hat{x}_{f,h_1}-\epsilon\right)\left(1-\hat{x}_{f,h_2}+\epsilon\right)\prod_{h\in\mathcal{H}(u):h\neq h_1,h_2,0}\left(1-\hat{x}_{f,h}\right)$. This expression is quadratic in ϵ with upward concavity, and hence it is convex. This proves the theorem. □

Now, we apply Theorem A.6.1. Consider $\mathbf{R}_{opt}$ to be the optimal solution obtained by maximizing $L=\sum_{f,u}P_f\widetilde{\omega}_u L_{f,u}(\mathbf{R})$ subject to the constraints in program (2.3) where $x_{f,h}$ is replaced by relaxed variables $\rho_{f,h}\in[0,1]$ as follows:

$$\begin{aligned}
\text{maximize}\quad & \sum_{f=1}^{F}P_f\sum_{u=1}^{U}\widetilde{\omega}_u\min\{1,\sum_{h\in\mathcal{H}(u):h\neq 0}\rho_{f,h}\}\\
\text{subject to}\quad & \sum_{f=1}^{F}\rho_{f,h}\leq M,\quad\forall\ h,\\
& \mathbf{R}\in[0,1]^{F\times H}
\end{aligned}\tag{A.6}$$

Let $\mathbf{X}_{int}$ be the solution obtained by running Pipage Rounding($\mathcal{K}_{\mathcal{F},\mathcal{H}}$, F

$, M, \mathbf{R}_{opt})$. By Theorem A.6.1, $g(\mathbf{X}_{int}) \geq \left(1 - (1 - 1/d)^d\right) g(\mathbf{X}_{opt})$ where $\mathbf{X}_{opt}$ is the optimum to problem (2.3) and $d = \max_u\{|\mathcal{H}(u)| - 1\}$.

Appendix B

Proofs for Chapter 3

B.1 Proof of Theorem 3.4.1

The proof is analogous to the ones for the UIC problem. The need for outlining an additional proof is because of additional technicalities due to the fact that user requests overlap. First, we consider the case of $\psi_\ell(\mathcal{H})$. Consider the optimal integral solution $(t, \{y_C\})$ to program (3.3). Every vertex is in exactly one hyperclique which is chosen. Consider the set $\mathcal{C}^{\text{opt}}$ of hypercliques C for which $y_C = 1$ (hyperclique chosen) in the optimal solution. Let $|\mathcal{C}^{\text{opt}}| = s$. Consider a $t \times s$ generator $\mathbf{G}$ of an (s, t) MDS code over a field Σ of size greater than s. Let the ith column be $\mathbf{g}_i$. Assign each column to a hyperclique in $\mathcal{C}^{\text{opt}}$. Let $C(u) \in \mathcal{C}^{\text{opt}}$ be the unique hyperclique to which user u belongs. Let packet p be denoted by $x_p \in \Sigma$, $\forall p \in P$. Define an equivalence relation $\sim$ between two users $u, v \in U$ such that $u \sim v$ iff $C(u) = C(v)$ and $R(u) = R(v)$. Let A_u be the equivalence class of u under $\sim$. Note that $u \in A_u$. Let us assume that U partitions into q equivalence classes, i.e. $U = \bigcup\limits_{i=1}^{q} A_{u_q}$. Then the transmission scheme is given by:

$$\mathbf{y} = \sum_{i \in [q]} \mathbf{g}_{C(u_i)} x_{R(u_i)}. \tag{B.1}$$

In other words, if there are two users who request the same packet and belong to the same clique, they belong to the same equivalence class and their terms can be merged into one. The broadcast rate is given by t.

We need to show that a user u can decode $\mathbf{x}_{R(u)}$ from this. All the terms with $\mathbf{x}_p$, $p \in S(u)$ can be cancelled due to the side information of u. This means that, in $\mathbf{y}$, all summands corresponding to users $W(S(u))$ do not affect decoding. Note that,

$$W\left(R(u) \bigcup (S(u))^c\right) = W\left(R(u)\right) \bigcup W\left((S(u))^c\right). \tag{B.2}$$

The first constraint in program (3.3) ensures that the number of distinct hypercliques from $\mathcal{C}^{\text{opt}}$ in $W\left(R(u)\right) \bigcup W\left((S(u))^c\right)$ is at most t. Let A_u be the equivalence class to which u belongs. Observe that $A_u \bigcap W\left((S(u))^c\right) = \emptyset$ because A_u only has users requesting $R(u)$ and $R(u) \notin (S(u))^c$ by definition. The number of hyperlcliques from $\mathcal{C}^{\text{opt}}$ in A_u is 1, by definition of A_u. Then, $\left(W\left(R(u)\right) - A_u\right) \bigcup W\left((S(u))^c\right)$ has at most $t - 1$ hyper cliques.

The terms corresponding to users in $\left(W\left(R(u)\right) - A_u\right) \bigcup W\left((S(u))^c\right)$ constitute the interference terms. Therefore, at most $t - 1$ distinct columns from $\mathbf{G}$ interferes with $\mathbf{g}_{C(u)}$. Since any t columns in $\mathbf{G}$ are linear independent, the interference can be cancelled. The difference from this and the UIC case is that user requests overlap leading to a more technical analysis.

Now, we move to specifying an achievable scheme for the LP relaxation of (3.3). Let $t, \{y_C\}_{C \in \mathcal{C}}$ constitute the optimum solution. Since, the constraints on the variables involve only integers, the optimal solution involves only rationals. Let r denote the least common multiple of denominators of y_C and t. Define the new variable $\hat{t} = rt$ and $\hat{y}_C = ry_C$. Now the new variables carry integral weights.

Every hyperclique (y_C) is assigned an integer weight in the set $\{0, 1, 2, \ldots r\}$. By the covering constraints on every vertex, every vertex is covered by exactly r hypercliques. Assume that every packet p is a super packet containing r subpackets $\mathbf{x}_{p_1} \ldots \mathbf{x}_{p_r} \in \Sigma$. Let $\sum\limits_{C} \hat{y}_C = s$.

If a hyperclique has weight $1 \leq q \leq r$, then consider q different copies of the same hyperclique. Denote the resulting multiset of cliques by $\mathcal{C}^{\text{opt}}$. Every hyperclique in $\mathcal{C}^{\text{opt}}$ has weight 1 with possible repetitions among cliques. Every user u is covered by at most r hypercliques from $\mathcal{C}^{\text{opt}}$. Now assign these r hypercliques to r different indices of the form (u, i), $1 \leq i \leq r$. Hence, every index pair (u, i) is assigned a subpacket $x_{(R(u))_i}$ and a hyperclique $C(u, i)$.

If two user requests overlap, i.e. $R(u) = R(v)$, then $(R(u))_i = (R(v))_i$ (all respective subpackets are identical). As before in the scalar case, define $\sim$ to be an equivalence relation such that $(u, i) \sim (v, j)$ iff $R(u) = R(v)$ and $C(u, i) = C(v, i)$ and $i = j$, $\forall v, u \in U$ $\forall i, j \in [r]$. Let $\mathcal{A}_{u,i}$ be an equivalence class, of all the index pairs which denote the same subpacket as the index pair (u, i) and are assigned the same hyperclique, under the relation $\sim$. Note that, $(u, i) \in \mathcal{A}_{u,i}$. Let the set $U \times [r]$ of all index pairs be partitioned into b equivalence classes, i.e. $U \times [r] = \bigcup\limits_{i=1}^{b} \mathcal{A}_{u_i,k_i}$ where $1 \leq k_i \leq r$ and $u_i \in U$. Now, consider an $\left(s, \hat{t}\right)$ MDS code over Σ with generator $\mathbf{G}$ with columns i denoted by $\mathbf{g}_i$. Assign every column to a distinct hyperclique in $\mathcal{C}^{\text{opt}}$ such that a column is denoted by $\mathbf{g}_C$ after the hyperclique C assigned to it. The transmission scheme is given by:

$$\mathbf{y} = \sum_{i=1}^{b} x_{(R(u_i))_{k_i}} \mathbf{g}_{C(u_i,k_i)} \tag{B.3}$$

If any two index pairs (u, i) and (v, i) are such that $R(u) = R(v)$ and if they are assigned the same hyperclique there is only one term corresponding to both of them.

We need to show that every user decodes all the subpackets $\mathbf{x}_{(R(u))_i}$, $\forall i \in [r]$. We define a modified W function tht produces index pairs instead of just users. Define $\tilde{W}(A) = \{(u, i) : R(u) \in A, i \in [r]\}$ for any subset $A \subseteq P$. Let us consider the decoding of subpacket $(R(u))_i$.

Now, we use arguments very similar to the scalar case. Rephrasing the first constraint in program (3.3), the number of distinct hypercliques in $\tilde{W}(R(u)) \bigcup \tilde{W}((S(u))^c)$ is at most $\hat{t}$ different hypercliques from $\mathcal{C}^{\text{opt}}$. Let $\mathcal{A}_{u,i}$ be the equivalence class to which (u, i) belongs.

Observe that $\mathcal{A}_{u,i} \bigcap \tilde{W}((S(u))^c) = \emptyset$. Hence, $\left(\tilde{W}(R(u)) - \mathcal{A}_{u,i}\right) \bigcup \tilde{W}((S(u))^c)$ has at most $\hat{t} - 1$ different hyper cliques from $\mathcal{C}^{\text{opt}}$. The summands in (B.3), corresponding to users in $\left(\tilde{W}(R(u)) - \mathcal{A}_{u,i}\right) \bigcup \tilde{W}((S(u))^c)$, constitute the interference terms. Therefore, at most $t-1$ distinct columns from $\mathbf{G}$ interferes with $\mathbf{g}_{C(u,i)}$. Since any $\hat{t}$ columns in $\mathbf{G}$ are linear independent, the interference can be cancelled and therefore user u can decode $\mathbf{x}_{(R(u))_i}$. Note that, the terms involving $\mathbf{x}_{(R(u))_j}$ constitute interference when $j \neq i$. Since, every user receives r subpackets and the total number of transmissions is $\hat{t}$, rate is given by t. This concludes the proof.

B.2 Proof of Theorem 3.4.2

We begin with the proof for the fractional partition multicast number, denoted by ψ_f^p, given by the LP relaxation of program (3.2). Let us first consider the integer version given by (3.2) before moving onto the LP relaxation. In an optimal solution, the set of users is partitioned into multicast groups, i.e. $U = \bigcup_{i=1}^{k} M_i$. Every user belongs to one multicast group. In a multicast group, the minimum size of the side information set is found. It is given by $\min\limits_{u \in M} |R(M) \bigcap S(M)|$. Note that, once a multicast group is considered, the problem is to satisfy only the users in the multicast group and only their packets participate in the transmission. Hence, the relevant induced bipartite graph is $\mathcal{H}\left(M, R(M)\right)$.

The transmission scheme is given by an $(R(M), d_M)$ MDS code. Since, every user has $R(M) - d_M$ number of distinct packets as side information, by the MDS property every user in the multicast can decode his request. The overall scheme is given by time sharing the different multicast groups, i.e. $\{M_1, M_2 \ldots M_k\}$ in the partition.

For the LP relaxation, every nonzero subset $M \in 2^U$ is a multicast group. The transmission scheme for each group is the same as the scalar case. The only difference is in time sharing. Since, the program in (3.2) has only integer coefficients, the real optimal solutions a_M are rational. As in other proofs, let r be the least common multiple of denominators of a_M. Let $\hat{a}_M = r a_M$. With the new variables, the first constraint in the LP relaxation of (3.2) implies that every user is in exactly r multicast groups. Hence, every user packet consists of r subpackets and each subpacket is transmitted using the scalar scheme corresponding to one of the

r multicast groups. Hence, ψ_f^p is achievable because of rate normalization by r.

Now, we provide an achievable scheme for ψ_ℓ^p (integer program (3.4)). Given a multicast group M, the variables $t_M, \{y_C\}$ constitute a scalar achievable scheme with t_M transmissions identical to the one used to achieve ψ_ℓ, as in Theorem 3.4.1, for the GIC problem (defined on $\mathcal{H}(M, R(M))$) induced by the multicast group. And the various disjoint multicast groups are timeshared. This provides an achievable scheme for ψ_ℓ^p.

Now, consider $\psi_{f\ell}^p$ (LP relaxation of (3.4)). Let the optimal solution be given by real values $\{t_M\}, \{y_C\}, \{a_M\}$. Since, the program has only integral constraints, all variables are rational. Now, consider r_1 to the least common multiple of denominators of $\{y_C\}, \{t_M\}$. Now, define $\hat{y}_C = r_1 y_C$ and $\hat{t}_M = r_1 t_M$. All the new variables are integral. For a particular multicast group M, apply the vector coding scheme of $\psi_{f\ell}$ on the GIC problem (defined on $\mathcal{H}\ (M, R(M))$) induced by the group M. This needs $r_1 t_M$ transmissions and every user in M gets r_1 subpackets. Call this scheme $\mathcal{S}_M$. Now, let r_2 be the least common multiple of $\{a_M\}$. Now, consider $r_1 r_2$ sub packets for every user. Now use the scheme S_M, which transmits r_1 subpackets, $a_M r_2$ times. Since, every user is exactly in r_2 subgroups (by constraint 2 in the LP relaxation of (3.4)), every user gets $r_1 r_2$ subpackets. The total number of subpackets transmitted is $\sum\limits_M a_M r_2 (t_M r_1)$. Dividing by $r_1 r_2$, we get the same broadcast rate as the objective in (3.4).

B.3 Proof of Theorem 3.5.1

We prove first prove the following result which upper bounds $\chi_f(\bar{G}_d)/\chi_\ell(\bar{G}_d)$ and expressions in its analysis will then be used to prove Theorem 3.5.1.

Theorem B.3.1. $\chi_f(\bar{G}_d)/\chi_\ell(\bar{G}_d) \leq \frac{5}{4}e^2$

Before proving Theorem B.3.1, we would like to recall notations and present a few technical lemmas. We always work on the interference graph $\bar{G}_d$ which is the directed complement of the directed side information graph G_d. G_u is obtained by throwing away uni-directed edges $(i,j) \in E_d,\ (j,i) \notin E_d$ and replacing bi-directed edges $(i,j),(j,i) \in E_d$ by an undirected edge $\{i,j\}$. $\bar{G}_u$ is the complement of G_u. Alternatively, $\bar{G}_u$ is obtained by ignoring orientations of the directed edges in the interference graph $\bar{G}_d$. If there is a bi-directed edge in $\bar{G}_d$, only one undirected edge is used to replace them in $\bar{G}_u$.

From discussions in the previous sections, $\bar{\chi}_f(G_d) = \bar{\chi}_f(G_u) = \chi_f(\bar{G}_u) = \chi_f(\bar{G}_d)$. We intend to upper bound the ratio $\chi_f(\bar{G}_d)/\chi_\ell(\bar{G}_d)$ for any G_d. We use ideas of graph homomorphisms and universal graphs already defined in [53]. Let us review those concepts. A directed graph G_d is homomorphic to another directed graph H_d if there is a function $f : V(G_d) \to V(H_d)$ such that if $(i,j) \in E(G_d)$, then $(f(i), f(j)) \in E(H_d)$. In other words, the directed edges are preserved under the mapping f. Similarly, an undirected graph G is homomorphic to another undirected graph H if there is a function $f : V(G) \to V(H)$ such that $\{i,j\} \in E(G)$ implies $\{f(i), f(j)\} \in E(H)$.

Let G_u be the undirected graph obtained by ignoring orientation of the directed edges in G_d (and replacing any bi-directed edge by a single undirected edge). Let H_u be obtained similarly from H_d. Then,

Lemma B.3.1. *If G_d is homomorphic to H_d, then G_u is homomorphic to H_u.*

Proof. If $\{i, j\} \in G_u$, then either $(i, j) \in E(G_d)$ or $(j, i) \in E(G_d)$ or both. This implies either $(f(i), f(j)) \in E(H_d)$ or $(f(j), f(i)) \in E(H_d)$. This means that $\{f(i), f(j)\} \in E(H_u)$. Hence G_u is homomorphic to H_u. □

Let us recall the definitions of the universal graph $U_d(m, k)$ from [53]. Let $[m]$ denote $\{1, 2, 3 \ldots m\}$.

Definition B.3.1. $V(U_d(m, k)) = \{(x, A) : x \in [m], A \subseteq [m], |A| = k - 1, x \notin A\}$. $E(U_d(m, k)) = \{((x, A), (y, B)) : y \in A\}$. *In other words, there is a directed edge from (x, A) to (y, B) if $y \in A$.* ◊

With some different notation from [53] , let us define the undirected graph $U(m, k)$ obtained by disregarding the orientation of the edges in $U_d(m, k)$. It is defined as:

Definition B.3.2. $V(U(m, k)) = \{(x, A) : x \in [m], A \subseteq [m], |A| = k - 1, x \notin A\}$. $E(U(m, k)) = \{((x, A), (y, B)) : y \in A \text{ or } x \in B\}$. *In other words, there is an undirected edge between (x, A) and (y, B) if $y \in A$ or $x \in B$ or both.* ◊

It has been shown in [53] that if $\chi_\ell(\bar{G}_d) = k$, then there is an integer $m \geq k$ such that $\bar{G}_d$ is homomorphic to $U_d(m, k)$ and $\chi_\ell(U_d(m, k)) = k$. Since $\bar{G}_d$

is homomorphic to $U_d(m,k)$, from Lemma B.3.1, $\bar{G_u}$ is homomorphic to $U(m,k)$. It is known from [116], that if undirected G is homomorphic to undirected H, then $\chi_f(G) \leq \chi_f(H)$. Then clearly, $\chi_f(\bar{G_d})/\chi_\ell(\bar{G_d}) \leq \chi_f(U(m,k))/k$ for all $\bar{G_d}$: $\chi_\ell(\bar{G_d}) = k$. We will show that $\chi_f(U(m,k))/k$ is upper bounded by e^2 for all parameters.

Proof of Theorem B.3.1. We show that $\chi_f\left(U(m,k)\right)/k \leq \frac{5}{4}\mathrm{e}^2, k \geq 2$. It is easy to observe that $U(m,k)$ is a vertex transitive graph. For vertex transitive graphs, it is known that $\chi_f\left(U(m,k)\right) = \frac{|V(U(m,k))|}{\alpha(U(m,k))}$, where $\alpha(.)$ is the independence number of the graph. We analyze the structure of maximal independent sets of the graph $U(m,k)$. Let $I = \{(x_1,A_1),(x_2,A_2)\ldots(x_n,A_n)\}$ be an independent set. Let $X = \{x_1,x_2\ldots x_n\}$ and $A = \bigcup\limits_i A_i$. Then clearly, from the definition of $U(m,k)$, $A \cap X = \emptyset$, otherwise there is an edge between two vertices in the set I. If $X \cup A \neq [m]$, then one can add elements (x_i,A_i) to I s.t. $x_i \in X$ and $A_i \in A' = [m]-X$. Let $|A'| = m-p$ and $|X| = p$. Choose the set I' containing all possible vertices (x,B) such that $x \in X$ and $B \subseteq A'$, $|B| = k-1$. I' is independent and $I \subseteq I'$. $|I'| = p\binom{m-p}{k-1}$.

Hence, every independent set is contained in a suitable independent set I' parametrized by p. Therefore, $\alpha\left(U(m,k)\right) = \max\limits_{1\leq p\leq m-k+1} p\binom{m-p}{k-1}$. $|V\left(U(m,k)\right)| = m\binom{m-1}{k-1}$.

Let $m' = \lfloor m/k \rfloor * k$. Clearly, $m \leq m'+k$. If $m \geq 4k$, then $m' \geq 4k$. For $m \geq 4k$, we have:

$$\chi_f\left(U(m,k)\right)/k = \min_{1\leq p\leq m-k+1} \frac{m\binom{m-1}{k-1}}{kp\binom{m-p}{k-1}} \tag{B.4}$$

$$\leq \min_{1\leq p\leq m-k+1} \frac{(m'+k)\binom{m-1}{k-1}}{kp\binom{m-p}{k-1}} \tag{B.5}$$

$$\leq \min_{1\leq p\leq m-k+1} \frac{(m'+k)\binom{m'-1}{k-1}}{kp\binom{m'-p}{k-1}} \tag{B.6}$$

$$\leq \left.\frac{(m'+k)\binom{m'-1}{k-1}}{kp\binom{m'-p}{k-1}}\right|_{p=m'/k} \tag{B.7}$$

$$\leq \frac{\left(1+\frac{k}{m'}\right)(m'-1)(m'-2)\ldots(m'-k+1)}{\left(m'\left(1-1/k\right)\right)\left(m'\left(1-1/k\right)-1\right)\left(m'\left(1-1/k\right)-2\right)\ldots\left(m'\left(1-1/k\right)-k+2\right)} \tag{B.8}$$

$$\leq \frac{5}{4}\frac{m'-k+1}{\left(m'\left(1-1/k\right)\right)}\frac{(m'-1)(m'-2)\ldots(m'-(k-2))}{\left(m'\left(1-1/k\right)-1\right)\left(m'\left(1-1/k\right)-2\right)\ldots\left(m'\left(1-1/k\right)-(k-2)\right)} \tag{B.9}$$

$$\leq \frac{5}{4}\frac{1}{\left(1-1/k\right)}\frac{(m'-1)(m'-2)\ldots(m'-(k-2))}{\left(m'\left(1-1/k\right)-1\right)\left(m'\left(1-1/k\right)-2\right)\ldots\left(m'\left(1-1/k\right)-(k-2)\right)} \tag{B.10}$$

$$\leq \frac{5}{4}\frac{1}{\left(1-1/k\right)}\frac{1}{\left(1-1/k\right)^{2(k-2)}} \tag{B.11}$$

$$\leq \frac{5}{4}\frac{1}{\left(1-1/k\right)^{2(k-1)}} \tag{B.12}$$

$$\leq \frac{5}{4}\mathrm{e}^{2} \tag{B.13}$$

(B.5) is because $m \leq m'+k$. (B.6) is because $\frac{\binom{m-1}{k-1}}{\binom{m-p}{k-1}}$ is a decreasing function with respect to the variable m. To see that, $\frac{\binom{m-1}{k-1}}{\binom{m-p}{k-1}} = \prod_{i=1}^{k-1} \frac{m-i}{m-(p-1)-i}$. But, $\frac{m-i}{m-(p-1)-i} = 1 + \frac{p-1}{m-(p-1)-i}$ which is decreasing in m. (B.7) is because substituting any value for p gives an upper bound. Here, we choose $p = m'/k$. (B.10) is because $m-k+1 \leq m,\ \forall k \geq 1$. (B.11) is valid under the assumption that $m/k \geq 4, k \geq 2$ and Lemma B.3.2 given below. When $m \geq 4k$, $m' \geq 4k$. Therefore, (B.9) is valid. (B.13) is because $\left(1+\frac{1}{k-1}\right)^{k-1} \leq \mathrm{e},\ \forall k \geq 1$. We now prove Lemma B.3.2 needed to justify

(B.11) below.

If $m/k \leq 4$ then taking $p = 1$ in (B.4), we can show that the ratio is upper bounded by $m/k \leq 4$. □

We note that $k \geq 2$ is not a restriction, since, $\chi_\ell \geq 2$ for a digraph with at least one directed edge .

Lemma B.3.2. $\frac{m-i}{m(1-1/k)-i} \leq \frac{1}{(1-1/k)^2},\ \forall 1 \leq i \leq k,\ m \geq 4k,\ k \geq 2.$

Proof. The following equivalence hold for $x \geq y$:

$$\frac{x-i}{y-i} \leq \frac{x^2}{y^2} \Leftrightarrow i(x+y) \leq xy \tag{B.14}$$

It is sufficient to show that $i \leq xy/(x+y)$ when $i \leq k$, $x = m,\ y = m\left(1-1/k\right)$. This means it is enough to show $k \leq m\frac{(1-1/k)}{2}$. Since $k \geq 2$, $(1-1/k) \geq 1/2$, for $m \geq 4k$, $k \leq m\frac{(1-1/k)}{2}$ holds. Hence, the statement in the lemma is proved. □

Now, we move to proving Theorem3.5.1. Since we work on the interference graph, $\bar{G}_d$ will be used throughout. The proof of the theorem follows a similar recipe as the proof of the bound for the ratio between fractional chromatic and the local chromatic number, i.e. the ratio $\chi_f\left(\bar{G}_d\right)/\chi_l\left(\bar{G}_d\right)$. Apart from the *LP* characterization of the fractional local chromatic number, there is another characterization using $r-$fold local colorings of a directed graph $\bar{G}_d$ [53]. We will review the results regarding $r-$ fold colorings from [53].

Definition B.3.3. *(r-fold local coloring number of a digraph) A proper r-fold coloring of $\bar{G}_d$ is a coloring of $\bar{G}_u$ (the undirected graph obtained by removing edge* *orientations in $\bar{G}_d$) where r distinct colors are assigned to each vertex such that color set of two adjacent vertices are disjoint. The r-fold local coloring number of a graph, denoted by $\chi_\ell(\bar{G}_d, r)$, is less than or equal to k if there is a proper r-fold coloring of $\bar{G}_d$ such that the total number of distinct colors in the closed out-neighborhood of any vertex is at most k.* ◊

$\chi_\ell\left(\bar{G}_d, r\right)$ is the minimum possible maximum number of colors in any closed out neighborhood over all possible proper r-fold colorings of graph $\bar{G}_d$. It is known that [53]:

$$\chi_{f\ell}(\bar{G}_d) = \inf_r \chi_\ell(\bar{G}_d, r)/r \tag{B.15}$$

Let us define a directed universal graph $U_d(r, m, k)$ as follows:

Definition B.3.4. $V\left(U_d(r, m, k)\right) = \{(X, A) : x \subseteq [m],\ A \subseteq [m],\ |A| = k - r,\ |X| = r,\ X \cap A = \emptyset\}$. $E\left(U_d(r, m, k)\right) = \{((X, A), (Y, B)) : Y \subseteq A\}$. *In other words, there is a directed edge from* (X, A) *to* (Y, B) *if* $Y \subseteq A$. ◊

Consider the undirected graph $U(r, m, k)$ obtained by ignoring the orientation of the directed edges in $U_d(r, m, k)$ and bi-directed edges replaced by a single undirected edge. It is formally defined as:

Definition B.3.5. $V\left(U(r, m, k)\right) = \{(X, A) : x \subseteq [m],\ A \subseteq [m],\ |A| = k - r,\ |X| = r,\ X \cap A = \emptyset\}$. $E\left(U(r, m, k)\right) = \{((X, A), (Y, B)) : Y \subseteq A \text{ or } X \subseteq B\}$. *In other words, there is a directed edge from* (X, A) *to* (Y, B) *if* $Y \subseteq A$ *or* $X \subseteq A$ *or both.* ◊

Replacing all undirected edges by directed edges and out-neighborhoods by closed out-neighborhoods in the argument of Lemma 3 in [53], we have the following lemma:

Lemma B.3.3. *Given $\bar{G}_d$, if $\chi_\ell\left(\bar{G}_d, r\right) \leq k$ then there is an m such that $\bar{G}_d$ is homomorphic to $U_d(r,m,k)$.*

This means, in the undirected sense $\bar{G}_u$ is homomorphic to $U(r,m,k)$, by Lemma B.3.1, if $\bar{G}_d$ is homomorphic to $U_d(r,m,k)$ in the directed sense. Also, $\chi_f(\bar{G}_d) \leq \chi_f\left(U(r,m,k)\right)$ as fractional chromatic numbers cannot decrease under homomorphism.

Consider $\bar{G}_d$ such that $\chi_\ell(\bar{G}_d, r) = k$ for some r and k. By Lemma B.3.3, $\bar{G}_d$ is homomorphic to $U_d(r,m,k)$ for some m. Hence, $\chi_f(\bar{G}_d) = \chi_f(\bar{G}_u) \leq \chi_f\left(U(r,m,k)\right)$. Here, $k \geq 2$ as it is a r-fold local chromatic number of a directed graph with one directed edge. Now, we have the following lemma:

Lemma B.3.4. $\chi_f(\bar{G}_d)/\frac{k}{r} \leq \chi_f\left(U(r,m,k)\right)/\frac{k}{r} \leq \frac{5}{4}e^2$.

Proof. The first part of the inequality follows from the fact that $\bar{G}_d$ is homomorphic to $U_d(r,m,k)$ as discussed before. It is easy to see that $U_d(r,m,k)$ is vertex transitive. Therefore, $\chi_f\left(U(r,m,k)\right) = \frac{|V(U(r,m,k))|}{\alpha(U(r,m,k))}$ where $\alpha(.)$ is the independence number of the graph. To upper bound the ratio $|V|/\alpha$, we provide a lower bound for the independence number by constructing a suitably large independent set.

Consider p elements from the set $[m]$ and denote the set by Z. Let $1 \leq p \leq m-k+1$. Pick one out of the p elements in Z, say z. Now create vertices (X,A) as follows: $z \in X$ and choose $k-1$ elements out of remaining $m-p$ elements. Then choose $r-1$ elements out of this and put in X apart from z. The remaining $k-r$ elements form A . Let I_z denote the set of (X,A) vertices created by the above method. Since $z \in X$, $\forall (X, A) \in I_z$, I_z is an independent set by the definition of $U(r, m, k)$. We now argue that $\bigcup_{x \in Z} I_x = I$ is an independent set. We need to argue that there cannot be an edge between $(X, A) \in I_x$ and $(Y, B) \in I_y$ for $x \neq y$. $x \in X$ but $x \notin B$ because when I_y (and in particular $(Y, B) \in I_y$) is formed x is not considered at all. Similarly, $y \in Y$ but $y \notin A$. Hence, I is an independent set. $|I_z| = \binom{m-p}{k-1}\binom{k-1}{r-1}$.

Hence , $\alpha\left(U(r, m, k)\right) \geq \max_{1 \leq p \leq m-k+1} p\binom{m-p}{k-1}\binom{k-1}{r-1}$. Substituting the lower bound , we have the following:

$$\chi_f\left(U(r, m, k)\right)/\frac{k}{r} \leq \frac{|V|}{\alpha\frac{k}{r}} \leq \frac{\binom{m}{k}\binom{k}{r}}{\max_{1 \leq p \leq m-k+1} p\binom{m-p}{k-1}\binom{k-1}{r-1}\frac{k}{r}} \tag{B.16}$$

$$= \frac{\binom{m}{k}\binom{k}{r}}{\max_{1 \leq p \leq m-k+1} p\binom{m-p}{k-1}\binom{k}{r}} \tag{B.17}$$

$$= \frac{\binom{m}{k}}{\max_{1 \leq p \leq m-k+1} p\binom{m-p}{k-1}} = \frac{m\binom{m-1}{k-1}}{k \max_{1 \leq p \leq m-k+1} p\binom{m-p}{k-1}} \tag{B.18}$$

The last expression is identical to the ratio of χ_f to χ_ℓ of $U(m, k)$ for the directed local chromatic number in (B.4) in Theorem B.3.1. We have seen the ratio is upper bounded by $\frac{5}{4}\mathrm{e}^2$ when $k \geq 2$ in Theorem B.3.1 . Hence, we conclude our proof. □

Proof of Theorem 3.5.1. From the lemma above, $\chi_f(\bar{G}_d)/\left(\chi_\ell(\bar{G}_d, r)/r\right) \leq \frac{5}{4}\mathrm{e}^2$. We have the following chain to prove the main result.

$$\begin{aligned} \chi_f(\bar{G}_d)/\left(\chi_{f\ell}(\bar{G}_d)\right) &= \chi_f(\bar{G}_d)/\left(\inf_r \chi_\ell(\bar{G}_d, r)/r\right). \\ &= \sup_r \chi_f(\bar{G}_d)/\left(\chi_\ell(\bar{G}_d, r)/r\right) \leq \frac{5}{4}\mathrm{e}^2. \end{aligned}$$

□

We provide a numerical example where the ratio is larger than 2.5. Consider $U_d(281, 9)$. Computer calculations show that $\chi_f\left(U_d(289, 9)\right) / \chi_\ell\left(U_d(289, 9)\right) = 2.5244$.

B.4 Proof of Theorem 3.5.2

The left inequality is obvious because $\psi_{f\ell}$ is the LP relaxation of ψ_ℓ. To prove the right inequality, given any GIC problem on $\mathcal{H}$, we come up with a UIC problem on a side information graph G_d such that $\psi_f(\mathcal{H}) = \chi_f\left(\bar{G}_d\right)$ and $\psi_{f\ell}(\mathcal{H}) \geq \chi_{f\ell}\left(\bar{G}_d\right)$. This will imply that $\frac{\psi_f(\mathcal{H})}{\psi_{f\ell}(\mathcal{H})} \leq \frac{\chi_f(\bar{G}_d)}{\chi_{f\ell}(\bar{G}_d)}$.

In the GIC problem on $\mathcal{H}$, the number of packets is less than that of the number of users. To convert it into a UIC problem with side information graph G_d on the user set U, we introduce a packet for every user such that user u requests packet x_u. The user set is identical to both problems. For user u, if $v \in W(S(u))$ in the GIC problem, then $(u, v) \in G_d$. In other words, if user u has a packet requested by user v in GIC problem, packet x_v is present as side information with user u in the UIC version. Further, if requests of users v and u are identical in the GIC problem, then $(u, v), (v, u) \in E_d$.

From the above construction, it is clear that if C is a hyperclique in $\mathcal{H}$, then C is a clique in the side information graph G_d and vice versa. A clique C in a directed graph is defined to be the complete graph on the vertices of C. Any two vertices in a complete graph have edges in both directions. ψ_f is an efficient covering of all users by hypercliques. $\chi_f\left(\bar{G}_d\right)$ is also an efficient covering of all users in G_d by cliques in G_d. Since, the user set is identical and the set of cliques is identical to the set of hypercliques, $\psi_f(\mathcal{H}) = \chi_f\left(\bar{G}_d\right)$.

Now, we show that $\chi_{f\ell}(G_d) \leq \psi_{f\ell}(\mathcal{H})$. Consider a hyperclique C in $\mathcal{H}$. Then C is a clique in G_d. Consider some arbitrary weights y_C for all C and consider a scalar t that satisfies the constraints in LP relaxation of program (3.3). Since, the user set is identical if the weights y_C assigned cover every user u in $\mathcal{H}$, the covering constraint holds for G_d too. Let $N(u)$ represent the out-neighborhood of u in G_d. Let $(N(u))^c = U - N(u) - u$. The equivalent of constraint 1 of the LP relaxation of program (3.3) for $\chi_{f\ell}\left(\bar{G}_d\right)$ is: $\sum\limits_{C \bigcap ((N(u))^c \bigcup u)} y_C \leq t$. It is enough to show that this equivalent constraint is satisfied for t. Observe that $(N(u))^c \subseteq W\left((S(u))^c\right)$ and $u \in W(R(u))$. Therefore, $C \bigcap ((N(u))^c \bigcup u) \subseteq C \bigcap W\left(R(u)\right) \bigcup W\left((S(u))^c\right)$. Therefore, the equivalent constraint holds. From the above two results, we obtain $\frac{\psi_f(\mathcal{H})}{\psi_{f\ell}(\mathcal{H})} \leq \frac{\chi_f\left(\bar{G}_d\right)}{\chi_\ell\left(\bar{G}_d\right)}$. Hence, the result in the theorem follows.

B.5 Proof of Theorem 3.5.2

Setting $a_U = 1$ and $a_M = 0$, $\forall M : |M| < |U|$ in the LP relaxation of program (3.4) and optimizing for the rest of the variables one gets $\psi_{f\ell}$ which is less than or equal to ψ_f by definition. Hence, the first chain of inequalities is proved.

Now, we show that the fractional partition multicast number $\psi_f^p \geq \psi_{f\ell}^p$. To see this, consider the optimal solution of the relaxation of program (3.2) given by a_M. Now, it is enough to show that there exists a set of feasible variables y_C

and $t_M = d_M$ such that the constraints of the LP relaxation of program (3.4) are satisfied. If $C = W(p)$ for some packet $p \in P$, then $y_C = 1$, otherwise $y_C = 0$. In other words, assign a weight of 1 to those hypercliques that comprise the set of users requesting the same packet p for some $p \in P$ and all other hypercliques are assigned a weight 0. Since, every user is contained in a unique hyperclique characterized by the packet the user requests, constraint 3 in the LP relaxation of program (3.4) is satisfied. Constraint 2 is satisfied because the set of variables a_M form a feasible solution to the LP relaxation of (3.2). We need to show that the first constraint is satisfied with $t_M = d_M$ for all M and $u \in M$. Consider a particular M. By the assignment of variables y_C, we have the following chain of inequalities:

$$\sum_{C:W(R(u)\bigcup(S(u))^c)\bigcap C\bigcap M\neq\emptyset} y_C = R(M) - |R(M) - S(u)| \leq d_M \tag{B.19}$$

This is because, the assignment of values imply that a hyperclique with non-zero weight is 'synonymous' with the packet and the number of hypercliques in $W\left(R(u)\bigcup(S(u))^c\right)\bigcap M$ is exactly the number of packets $R(M) - S(u)$. This completes the proof.

B.6 Proof of Theorem 3.5.4

Let the set of variables $\{t_M, y_C, a_M\}$ be the optimal solution to the LP relaxation of program (3.4). Let $\mathcal{C}_M$ be the set of hypercliques for the induced GIC problem on $\mathcal{H}(M, R(M))$. Let $\hat{y}_{C_M} = \sum\limits_{C:C\bigcap M=C_M} y_C$, $\forall C_M \in \mathcal{C}_M$. Observe that, since variables $\{t_M\}$ and $\{y_C\}$ satisfy the first constraint of the LP relaxation of program (3.4), the variables $\hat{y}_C$ and the variable $\{t_M\}$ satisfy the constraints of the

LP relaxation of program (3.3) on the induced GIC problem given by $\mathcal{H}\left(M, R(M)\right)$. Therefore, $t_M \geq \psi_{f\ell}\left(\mathcal{H}\left(M, R(M)\right)\right)$. Now, we have the following chain of inequalities:

$$\begin{aligned}
\frac{\psi_f\left(\mathcal{H}\right)}{\psi^{p}_{f\ell}\left(\mathcal{H}\right)} &\leq \frac{\psi_f\left(\mathcal{H}\right)}{\sum\limits_{M \in 2^U - \{0\}} a_M t_M} \\
&\overset{a}{\leq} \frac{\psi_f\left(\mathcal{H}\right)}{\sum\limits_{M \in 2^U - \{0\}} a_M \psi_{f\ell}\left(\mathcal{H}\left(M, R(M)\right)\right)} \\
&\overset{b}{\leq} \frac{\sum\limits_{M \in 2^U - \{0\}} a_M \psi_f\left(\mathcal{H}\left(M, R(M)\right)\right)}{\sum\limits_{M \in 2^U - \{0\}} a_M \psi_{f\ell}\left(\mathcal{H}\left(M, R(M)\right)\right)} \\
&\overset{c}{\leq} \frac{\sum\limits_{M \in 2^U - \{0\}} a_M \psi_f\left(\mathcal{H}\left(M, R(M)\right)\right)}{\sum\limits_{M \in 2^U - \{0\}} a_M \frac{1}{\max\limits_{G_d} \frac{\chi_f(\bar{G}_d)}{\chi_{f\ell}(\bar{G}_d)}} \psi_f\left(\mathcal{H}\left(M, R(M)\right)\right)} \leq \max_{G_d} \frac{\chi_f(\bar{G}_d)}{\chi_{f\ell}(\bar{G}_d)}
\end{aligned} \tag{B.20}$$

Justifications for the above chain are: a) $t_M \geq \psi_{f\ell}\left(\mathcal{H}\left(M, R(M)\right)\right)$ b) Partitioning the set of users and adding up ψ_f over all partitions can not increase ψ_f for the original GIC problem. c) Theorem 3.5.2.

Appendix C

Proofs for Chapter 4

C.1 Proof of Theorem 4.3.1

We first show that if $\mathcal{C}$ is a valid index code on $\bar{G}$, then the dual code $\mathcal{C}^{\perp}$ with its generator $\mathbf{G}$ is a valid GLRC code for $\bar{G}$. Consider any user i in the index coding problem. Let the side information set be S_i. If $\mathcal{C}$ is a valid index code, then there exists a vector linear (linear in all the subsymbols) decoding function $\phi_i : \phi_i\left(\mathbf{y}, \{\mathbf{x}_j\}_{j \in S_i}\right) = \mathbf{x}_i$. This is true for all message vectors $\mathbf{x} : \mathbf{y} = \mathbf{V}\mathbf{x}$. Let $\mathbf{w}$ be a vector such that $\mathbf{y} = \mathbf{V}\mathbf{w}$. Let $\mathbf{x}$ represent the actual message vector (of all n messages). Let the encoded transmission be $\mathbf{y}$. Then, $\mathbf{x} = \mathbf{w} + \mathbf{z}$ for some $\mathbf{z} \in \mathcal{C}^{\perp}$ because $\mathcal{C}^{\perp}$ is the right null space of $\mathbf{V}$.

Given $\mathbf{y}$, the uncertainty about message vector $\mathbf{x}$ is because of the unknown $\mathbf{z}$ in the null space. In that sense, given the generator $\mathbf{V}$ of the code, one can fix a candidate $\mathbf{w}$ for a given $\mathbf{y}$. Because ϕ_i is linear in all the arguments, we have the following chain of inequalities:

$$\begin{aligned}
&\phi_i(\mathbf{y}, \{\mathbf{x}_j\}_{j \in S_i}) = \mathbf{x}_i \\
\Rightarrow &\phi_i(\mathbf{y}, \{\mathbf{w}_j + \mathbf{z}_j\}_{j \in S_i}) = \mathbf{w}_i + \mathbf{z}_i \\
\Rightarrow &\phi_i(\mathbf{y}, \{\mathbf{w}_j\}_{j \in S_i}) + \phi_i(\mathbf{0}, \{\mathbf{z}_j\}_{j \in S_i}) = \mathbf{w}_i + \mathbf{z}_i
\end{aligned} \tag{C.1}$$

The last step uses linearity of ϕ_i. The decoding should work even when $\mathbf{w}$ is the actual message vector. Hence, $\phi_i(\mathbf{y}, \{\mathbf{w}_j\}_{j \in S_i}) = \mathbf{w}_i$. With ($C.1$), we have:

$$\phi_i\left(\mathbf{0}, \{\mathbf{z}_j\}_{j \in S_i}\right) = \mathbf{z}_i \tag{C.2}$$

Since ϕ_i is linear, this implies that every subsymbol of the ith code supersymbol is linearly dependent on all the code subsymbols in the set S_j for the dual code $\mathcal{C}^{\perp}$ since $\mathbf{z} \in \mathcal{C}^{\perp}$. Hence, the dual code is a valid GLRC proving one direction.

To prove the other direction, let us assume that for every $i : 1 \leq i \leq n$, there exist functions $\tilde{\phi_i}$ such that :

$$\tilde{\phi_i}\left(\{\mathbf{z}_j\}_{j \in S_i}\right) = \mathbf{z}_i, \ \forall \mathbf{z} \in \mathcal{C}^{\perp} \tag{C.3}$$

Here, $\mathbf{z}$ is a vector of all supersymbols $\mathbf{z}_i$. This means that every supersymbol i of the GLRC code $\mathcal{C}^{\perp}$ is recoverable from the set S_i of codeword supersymbols. For the index coding problem, let $\mathbf{x}$ be the message vector not known to the users prior to receiving the encoded transmission. Let $\mathbf{y} = \mathbf{V}\mathbf{x}$. Given $\mathbf{y}$, from the previous part of the proof, we know that $\mathbf{y} = \mathbf{w} + \mathbf{z}$ for some $\mathbf{z} \in \mathcal{C}^{\perp}$. $\mathbf{w}$ is known to all users from just $\mathbf{y}$ because the code $\mathbf{V}$ employed is known to all the users.

Since $\mathbf{z}$ satisfies the recoverability conditions in (C.3), $\mathbf{w}_i + \tilde{\phi_i}\left(\{\mathbf{x}_j - \mathbf{w}_j\}_{j \in S_i}\right) = \mathbf{x}_i$. $\mathbf{w}$ is a function of just $\mathbf{y}$ and $\mathbf{V}$. Hence, user i can recover $\mathbf{x}_i$ from supersymbols from the side information set S_i and the encoded transmission $\mathbf{y}$ for all message vectors $\mathbf{x}$.

We again note that the choice of $\mathbf{w}$ is arbitrary. For every $\mathbf{y}$, users have to pick some $\mathbf{w}$ such that $\mathbf{y} = \mathbf{V}\mathbf{w}$. Since the forward map is linear, the inverse one-to-one map $\mathcal{V}^{-1}(\mathbf{y})$ determining $\mathbf{w}$ can be made linear by fixing $\mathcal{V}^{-1}(\mathbf{e}_i)$ for all unit vectors $\mathbf{e}_1, \ldots \mathbf{e}_k$. Then, linearity of the forward map determines a candidate pre-image for all vectors $\mathbf{y}$, i.e. $\mathcal{V}^{-1}(\mathbf{y}) = \sum_{i=1}^{k} y_i \mathcal{V}^{-1}(\mathbf{e}_i)$. Therefore, if $\tilde{\phi}_i$ are all linear in all the subsymbol arguments, then the decoding functions for the index coding problems are also linear. This completes the proof.

C.2 Proof of Theorem 4.3.2

We show that a correlated unicast code on the network $(\mathcal{G})$ is identical to a GLRC on a digraph $\bar{G}(V, E)$ which we construct as follows: There is a node for every edge in the network, i.e. $V = \mathcal{L}$. If the edge e is not a source edge, define the recoverability set $S_e = \{e' \in \mathcal{L} : h(e') = t(e)\}$. If $e \in \mathcal{E}_i$ (a source edge feeding into source i) for some i, then $S_e = \{e' \in \mathcal{L} : h(e') = d_i\}$. Recoverability set S_e forms the directed out-neighborhood of vertex e in $\bar{G}$. In other words, $(e, e') \in E$ iff $e' \in S_e$. It is easy to see that a GLRC code for $\bar{G}$ of dimension k is exactly the same as a correlated unicast code for $\mathcal{N}$ of dimension k and vice versa. This is because the decodability conditions at the destinations and local encoding conditions translate to recoverability conditions for the GLRC and vice versa. Hence, $R^{CO}(\mathcal{G}) = R^{GLRC}\left(\bar{G}\right)$.

C.3 Proof of Theorem 4.3.4

We first prove a statement comparing $m - \text{MAIS}(G)$ and $\text{GNSCUT}(\mathcal{G})$ before proving the main theorem.

Theorem C.3.1. *Consider a multiple-unicasts network $\mathcal{G}$ with k sources and m links. Let G be a digraph on m vertices constructed based on $\mathcal{G}$ as described in Section 4.3.2.*

Then,

$$m - \textsf{MAIS}(G) \;\leq\; \textsf{GNSCUT}(\mathcal{G}).$$

Proof. Recall that $\mathcal{G}'$ is the directed network obtained from $\mathcal{G}$ by setting the destination node t_i to be the tail of each source link of source s_i, $i = 1, \ldots, k$. (Section 4.3). Further, G is the (reversed) line digraph of $\mathcal{G}'$. Any set of vertices lying on a cyclic path in G corresponds to a set of edges forming a cycle in $\mathcal{G}'$. Hence, $m - \textsf{MAIS}(G)$ equals the cardinality of the minimum feedback edge set of the cyclic network $\mathcal{G}'$, *i.e.*, the smallest set of (unit-capacity) edges that need to be removed from $\mathcal{G}'$ to obtain an acyclic network. To show the desired result, it suffices to show any GNS cut in $\mathcal{G}$ is a feedback edge set in $\mathcal{G}'$.

Any cycle in $\mathcal{G}'$ must contain at least one of the edges connecting a destination node t_i to its source node s_i: these are the only links modified to obtain $\mathcal{G}'$ from $\mathcal{G}$, and the latter is an acyclic network. It turn, all cycles in $\mathcal{G}'$ are of the form $t_i, s_i, \ldots, t_i$.

Let S be a GNS cut in $\mathcal{G}$. By definition, there exists a permutation $\pi : [k] \rightarrow [k]$ such that if $\pi(i) \geq \pi(j)$, no path exists from s_i to t_j in $\mathcal{G} - S$. We want to show that $\mathcal{G}' - S$ is acyclic. Assume, for the sake of contradiction, that this is not the case, and let $\mathcal{C} \subseteq \{1, \ldots, k\}$ be the set of indices such that a source edge from t_i to s_i, $\forall i \in C$ lies on a cycle. Let $i^\star = \max_{i \in \mathcal{C}} \pi(i)$. Consider a cycle in $\mathcal{G}' - S$ going through an edge from $t_{i^\star}$ to $s_{i^\star}$; it must be of the form $t_{i^\star}, s_{i^\star}, \ldots, t_{i^\star}$. Without loss of generality, only one of the source edges from $t_{i^\star}$ to $s_{i^\star}$ occurs in this cycle. Since S is a GNS cut of $\mathcal{G}$, no path exists in $\mathcal{G}$ from $s_{i^\star}$ to $t_{i^\star}$. We conclude that a path from $s_{i^\star}$ to $t_{i^\star}$ in $\mathcal{G}'$ must use edges that are not available in $\mathcal{G}$, that is, edges from t_j to s_j for some $j \in \{1, \dots, k\}$. Let $j^\star$ be the source node such that a source edge from $t_{j^\star}$ to $s_{j^\star}$ is the first source edge appearing in the path from $s_{i^\star}$ to $t_{i^\star}$. Then, the path from $s_{i^\star}$ to $t_{j^\star}$ uses only edges in $\mathcal{G} - S$ (otherwise it would go through another source edge contradicting the fact that edge from $t_{j^\star}$ to $s_{j^\star}$ is the first source edge in the cycle after $s_{i^\star}$). Hence, there is a path in $\mathcal{G} - S$ from $s_{i^\star}$ to $t_{j^\star}$, with $\pi(i^\star) > \pi(j^\star)$ which is a contradiction. □

C.3.1 Proof of Theorem 4.3.4

Recall that G is the (reversed) line graph of $\mathcal{G}'$, the cyclic network obtained from $\mathcal{G}$ by connecting each destination node t_i to the source links of the source node s_i, $i = 1 \dots, k$, as described in Section 4.3.2.

The quantity $m-\textsf{MAIS}(G)$ is the cardinality of the minimum feedback vertex set in G, which in turn equals the cardinality of the minimum feedback edge set (FES) in $\mathcal{G}'$, *i.e.*,

$$m - \textsf{MAIS}(G) \quad = \quad \min_{\mathcal{F}' \text{ is a FES in } \mathcal{G}'} |\mathcal{F}'|. \tag{C.4}$$

We will show that the right hand side of (C.4) is equal to $\textsf{GNSCUT}(\widetilde{\mathcal{G}})$, *i.e.*, the cardinality of the smallest GNS cut in $\widetilde{\mathcal{G}}$. In fact, we will show that for every FES $\mathcal{F}'$ in $\mathcal{G}'$, there exists a GNS cut-set $\widetilde{\mathcal{F}}$ in $\widetilde{\mathcal{G}}$ with $|\widetilde{\mathcal{F}}| = |\mathcal{F}'|$, and vice versa.

Ignoring the source links in $\widetilde{\mathcal{G}}$ (links with head vertex $\widetilde{s_i}$ for some $i \in [k]$ and no tail vertex), we consider the following trivial one-to-one mapping $\mathcal{M}$ between the links of $\widetilde{\mathcal{G}}$ and those of $\mathcal{G}'$:

- Each of the $\textsf{MINCUT}(s_i, t_i)$ links from t_i to s_i in $\mathcal{G}'$ is mapped to a link from $\widetilde{s_i}$ to s_i in $\widetilde{\mathcal{G}}$.

- All remaining links are common in both networks.

Consider an arbitrary FES $\mathcal{F}'$ in $\mathcal{G}'$. Let $\widetilde{\mathcal{F}}$ be the image of $\mathcal{F}'$ under the mapping $\mathcal{M}$. We will show that $\widetilde{\mathcal{F}}$ is a GNS cut-set in $\widetilde{\mathcal{G}}$.

Claim C.3.1. *Consider a k-unicast network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and a subset of links $\mathcal{F} \subset \mathcal{E}$. If $\mathcal{F}$ is* not *a GNS cut-set in $\mathcal{G}$, then there exists a source-destination pair s_i, t_i with a path from s_i to t_i in $\mathcal{G} - \mathcal{F}$, or a sequence of $r \geq 2$ distinct indices $i_1, \ldots, i_r \in \{1, \ldots, k\}$ such source s_{i_j} has a path to destination t_{i_j+1}, for $j = 1, \ldots, r-1$, and s_{i_r} has a path to t_{i_1} in $\mathcal{G} - \mathcal{F}$.*

The proof of Claim C.3.1 is deferred to the end of this section. It follows from Claim C.3.1 that if $\widetilde{\mathcal{F}}$ is not a GNS cut-set of $\widetilde{\mathcal{G}}$, then $\mathcal{G}' - \mathcal{F}'$ contains a cycle, contradicting the fact that $\mathcal{F}'$ is a FES of $\mathcal{G}'$. We conclude that $\widetilde{\mathcal{F}}$ is a GNS cut-set in $\widetilde{\mathcal{G}}$. Note that $|\widetilde{\mathcal{F}}| = |\mathcal{F}'|$. Finally, the above implies that

$$\textsf{GNSCUT}(\widetilde{\mathcal{G}}) \quad \leq \quad \min_{\mathcal{F}' \text{ is a FES in } \mathcal{G}'} |\mathcal{F}'|. \tag{C.5}$$

Conversely, consider an arbitrary GNS cut-set $\widetilde{\mathcal{F}}$ in $\widetilde{\mathcal{G}}$. Let $\mathcal{F}'$ be the (inverse) image of $\widetilde{\mathcal{F}}$ according to the mapping $\mathcal{M}$. We will show that $\mathcal{F}'$ is an FES in $\mathcal{G}'$.

$\widetilde{\mathcal{F}}$ is a GNS cut-set. Hence, there exists a permutation $\pi : [k] \rightarrow [k]$ such that if $\pi(i) \geq \pi(j)$, no path exists from $\widetilde{s_i}$ to t_j in $\widetilde{\mathcal{G}} - \widetilde{\mathcal{F}}$, $\forall i, j \in [k]$. Assume, for the sake of contradiction, that $\mathcal{F}'$ is not a FES in $\mathcal{G}'$, *i.e.*, $\mathcal{G}' - \mathcal{F}'$ contains a cycle. Any cycle in $\mathcal{G}'$ has to include a link from the destination node t_i to the source node s_i, for some $i \in \{1, \ldots, k\}$, *i.e.*, it is of the form $t_i, s_i, \ldots, t_i$ including one or several source nodes. Using an argument identical to that in the proof of Theorem C.3.1: either *(i)* the cycle contains a path from s_i to t_i, which is a path in $\mathcal{G}$, or *(ii)* $\exists j$: $\pi(i) > \pi(j)$ and the cycle contains a path from s_i to t_j. This in turn implies (under the mapping $\mathcal{M}$) that $\widetilde{\mathcal{G}} - \widetilde{\mathcal{F}}$ contains either a path from $\widetilde{s_i}$ to t_i, or a path from $\widetilde{s_i}$ to t_j contradicting that $\widetilde{\mathcal{F}}$ is a GNS cut-set in $\widetilde{\mathcal{G}}$. We conclude that $\mathcal{F}'$ is a FES of $\mathcal{G}'$, while by construction $|\mathcal{F}'| = |\widetilde{\mathcal{F}}|$. Finally, the above imply that

$$\min_{\mathcal{F}' \text{ is a FES in } \mathcal{G}'} |\mathcal{F}'| \quad \leq \quad \textsc{GNSCUT}(\widetilde{\mathcal{G}}). \tag{C.6}$$

The theorem follows from (C.4), (C.5) and (C.6).

C.3.1.1 Proof of Claim C.3.1

We prove the contrapositive statement; if $\mathcal{G}-\mathcal{F}$ contains no source-destination pair s_i, t_i such that s_i has a path to t_i, nor a sequence of $r \geq 2$ distinct indices $i_1, \ldots, i_r \in \{1, \ldots, k\}$ with the properties described in the claim, then $\mathcal{F}$ is a GNS cut.

Consider a directed graph H on k vertices labeled $1, \ldots, k$, with vertex i corresponding to the ith source-terminal pair s_i, t_i of $\mathcal{G} - \mathcal{F}$. A directed edge (i, j) from vertex i to vertex j exists in H if and only if a path from s_i to t_j exists in $\mathcal{G} - \mathcal{F}$.

By assumption, $\mathcal{G} - \mathcal{F}$ contains no source-destination pair s_i, t_i such that s_i has a path to t_i. Hence, H contains no self-loops. Further, it is straightforward to verify that H contains a cyclic path $i_1, \ldots, i_r, i_1$, $r \geq 2$ if and only if the sequence of indices $i_1, \ldots, i_r \in \{1, \ldots, k\}$ satisfies the property described in the claim. By assumption, no such sequence exists in $\mathcal{G} - \mathcal{F}$ either. Hence, H is acyclic and has a topological ordering, *i.e.*, a permutation $\pi : [k] \rightarrow [k]$ of the k vertices such that for all $i, j \in [k]$, if $\pi(i) \geq \pi(j)$, then no edge exists from i to j . In turn, if $\pi(i) \geq \pi(j)$, no path exists from source s_i to destination t_j in $\mathcal{G} - \mathcal{F}$. The existence of such a permutation implies that $\mathcal{F}$ is a GNS cut of $\mathcal{G}$ (see Def. 4.3.2). □

C.4 Proof of Theorem 4.4.1

Consider a multiple-unicasts network $\mathcal{G}$ with k sources and m (unit-capacity) edges in the set $\mathcal{E}$. Recall that $\mathcal{G}'$ is the network formed by setting the destination t_i as the tail of all source links of s_i in $\mathcal{G}$, $\forall i \in \{1, \ldots, k\}$. We define $\mathcal{L}$ to be the set of *capacitated links* of $\mathcal{G}'$: $(a, b) \in \mathcal{L}$ if and only if there exists a link from a to b in $\mathcal{G}'$ and the capacity of (a, b), denoted by $c_{a,b}$, is equal to the number of unit links from a to b in $\mathcal{G}'$.

First, we observe that to find the smallest feedback edge set in $\mathcal{G}'$, it suffices to consider the set of capacitated links $\mathcal{L}$. Consider a set $\mathcal{F}'_E \subseteq \mathcal{E}$ be a minimal feedback edge set of $\mathcal{G}'$, *i.e.*, a minimal subset of *unit-capacity* edges whose removal from $\mathcal{G}'$ yields an acyclic network. If there exist multiple links from node u to node v in $\mathcal{G}'$, then either all or none of them is included in $\mathcal{F}'_E$. To verify that, let e, $\widehat{e}$ be unit links from u to v, and $e \in \mathcal{F}'_E$, while $\widehat{e} \notin \mathcal{F}'_E$. By construction, $\mathcal{G}' - \mathcal{F}'_E$ is an acyclic network. The minimality of $\mathcal{F}'_E$ implies that $\mathcal{G}'$ contains a cycle $u', \ldots, u \xrightarrow{e} v, \ldots, u'$ whose only edge contained in $\mathcal{F}'_E$ is e. But, $\widehat{e} \notin \mathcal{F}'_E$ and hence $u', \ldots, u \overset{\widehat{e}}{\rightarrow} v, \ldots, u'$ forms a cycle in $\mathcal{G}' - \mathcal{F}'_{E}$, contradicting the fact that $\mathcal{F}'_{E}$ is a feedback edge set.

The second key observation is that since $\mathcal{G}$ is acyclic, every cycle in $\mathcal{G}'$ must include an edge from a destination node to a source node. Equivalently, all cycles in $\mathcal{G}'$ go through the set of nodes $\mathcal{S} = \{s_1, \ldots, s_k\}$, *i.e.*, the set of k source nodes. The *minimum weight subset-feedback edge set problem* (see [82]) is the problem of finding a set of minimum weight edges that cuts all cycles passing through a *specific* set of nodes. Finding a feedback edge set in $\mathcal{G}'$ is equivalent to solving the minimum weight subset-feedback edge set problem for the set of source nodes $\mathcal{S}$ (using the capacitated links $\mathcal{L}$ with weights coinciding with the corresponding capacity). The modified sphere growing approximation algorithm of [82] with input $\mathcal{G}'$ and $\mathcal{S}$ constructs a feedback edge set of weight within a $O(\log^2 |\mathcal{S}|)$ factor from that of the minimum *fractional* weighted feedback edge set of $\mathcal{G}'$, in time polynomial in $|\mathcal{L}|$. The weight of the minimum *fractional* weighted feedback edge set coincides with the *fractional cycle packing number* of G, $r_{\text{CP}}(G)$, where G is the (reversed) line graph of $\mathcal{G}'$. In other words, the aforementioned algorithm yields a feedback edge set of G with weight at most $r_{\text{CP}}(G) \cdot O(\log^2 k)$.

It is known [117] that $m - r_{\text{CP}}(G)$ equals the broadcast rate of a vector-linear index code C for index coding instance with side-information graph G. In other words, there exists an index code C, which can be defined over any field $\mathbb{F}$, that achieves broadcast rate $\beta^{\mathbb{F}}_{\text{VL}}(\mathcal{C}; G) = m - r_{\text{CP}}(G)$. The basis for the coding scheme is that every cycle in G saves a transmission. This implies $m - \beta_{\text{VL}}(G) \geq r_{\text{CP}}(G)$. Further, by duality in Theorem 4.3.1, there is a feasible code for the relaxed-correlated sources problem on G whose joint entropy rate is $r_{\text{CP}}(G)$.

Appendix D

Proofs for Chapter 5

D.1 Proof of Theorem 5.2.1

Consider Algorithm 4. It is a greedy algorithm that determines the best transmission for every packet demanded by at least one user. According to the 'if' statement in the inner 'for' loop in Algorithm 4, packets requested by user k but stored in user caches $\mathcal{S} - k$ gets transmitted with only the packets which are stored in all but one user cache in the set $\mathcal{S}$. Therefore, we just need to show that the number of transmissions for a particular subset $\mathcal{S}$ in Algorithm 2 is same as the number of transmissions when the main for loop processes only packets corresponding to set $\mathcal{S}$ in Algorithm 2. From now on, let us only consider packets stored in all caches in $\mathcal{S}$ except the requesting user $k \in \mathcal{S}$.

For every user k, we will sort the packets in every vector $V_{k,\mathcal{S}-k}$ in Algorithm 2 according to the order in which packets, requested by user k but stored in exactly $\mathcal{S} - k$, are added to the set C of transmitted packets in Algorithm 4. Note, that this alignment of the ordering does not impact the number of transmissions of Algorithm 2.

Now, we show that the sequence of transmissions of Algorithm 2 for set $\mathcal{S}$ is exactly the same as the sequence of transmissions of Algorithm 4. Suppose, the sequence is not the same. Suppose the difference occurs in the ith transmission, $i \geq 1$, and not until $i-1$. Then, it implies that there exists users k and k' such that, (d_k, f) was combined with $(d_{k'}, f')$ in Algorithm 4 in transmission i while (d_k, f) was not combined with $(d_{k'}, f')$ in Algorithm 2. Since, the first $i-1$ transmissions happened exactly identically and $V_{k,\mathcal{S}-k}$'s are sorted according to accesses in 4, the ith element in $V_{k,\mathcal{S}-k}$ must be (d_k, f) and the ith element in $V_{k',\mathcal{S}-k'}$ must be $d_{k'}, f'$ and therefore those will be combined in Algorithm 2 during the ith transmission leading to a contradiction. Therefore, through a proof by contradiction, the theorem holds.

D.2 Proof of Theorem 5.3.1

Since the placement scheme in Algorithm 3 differs slightly with respect to Algorithm 1, we have the following technical result which will be useful in deriving a common proof:

Theorem D.2.1. *For placements schemes in Algorithms 3 and 1, we have:*

$$\Pr\left(\mathbf{1}_{d_i,f_1}^{\mathcal{S}-i} > 0 \bigcap \mathbf{1}_{d_j,f_2}^{\mathcal{S}-j} > 0\right) \leq \left(\Pr\left(\mathbf{1}_{d_i,f_1}^{\mathcal{S}-i} > 0\right)\right)^2 \tag{D.1}$$

whenever $(d_i, f_1) \neq (d_j, f_2)$.

Proof. We first observe that $\Pr\left(\mathbf{1}_{d_i,f_1}^{\mathcal{S}-i} > 0\right)$ is independent of the packet d_i, f_1 and hence symmetric for all packets and for all caches. In the case when $d_1 \neq d_2$, for both placement schemes, the placement across two different caches are independent of each other and probability of placing a particular file packet is symmetric across all file packets. Hence, the result holds.

In the case when $d_1 = d_2$, let us first consider the placement scheme in Algorithm 3. Every file of size F is divided into F' groups each of size $\lceil N/M \rceil$. Across the groups again the placement of file packets is independent. Therefore, if f_1 and f_2 belong to different groups, the result again holds. The interesting case is when f_1 and f_2 are in the same group. In this case, if packet f_1 is placed, then packets f_2 cannot be placed according to the constraints of Algorithm 3. Therefore, here the product is 0 and the inequality in the result still holds.

Consider the case when $d_1 = d_2$ and let the placement scheme be Algorithm 1. Let $|\mathcal{S}| = s$. When $f_1 \neq f_2$, we have:

$$\Pr\left(\mathbf{1}^{\mathcal{S}-i}_{d_i,f_1} > 0 \bigcap \mathbf{1}^{\mathcal{S}-j}_{d_j,f_2} > 0\right) = \left(\frac{1}{\frac{N}{M}}\right)^{s} \left(1 - \frac{1}{N/M}\right)^{K-s} \left(\frac{\frac{M}{N} - \frac{1}{F}}{1 - \frac{1}{F}}\right)^{s-1} \left(1 - \frac{\frac{MF}{N} - 1}{F}\right)^{K-s+1}$$
$$\overset{a}{\leq} \left(\left(\frac{1}{\frac{N}{M}}\right)^{s} \left(1 - \frac{1}{N/M}\right)^{K-s}\right)^{2} = \left(\Pr\left(\mathbf{1}^{\mathcal{S}-i}_{d_i,f_1} > 0\right)\right)^{2} \quad \text{(D.2)}$$

(a)-Once the file part f_1 of a file is chosen, according to Algorithm 1, the probability another file packet f_2 will be chosen is $\frac{\frac{MF}{N}-1}{F-1} \leq \frac{M}{N}$ whenever $M \leq N$. The probability that f_2 is not chosen given f_1 is also not chosen: $1 - \frac{\frac{MF}{N}}{F-1} = 1 - \frac{M}{N(1-\frac{1}{F})} \leq 1 - \frac{M}{N}$. □

We now show that, the coding gain is at most 2 even when the file size is exponential in the targeted gain $t = \frac{K}{N/M}$ when the existing delivery scheme (Algorithm 2 is applied with the existing placement scheme of Algorithm 1.

Under the placement scheme of Algorithm 1, for a set $\mathcal{S}$ of size s that includes k, define:

$$\mu(s) = \Pr\left(\mathbf{1}^{\mathcal{S}-k}_{d_k,f} > 0\right) = \left(\frac{1}{\frac{N}{M}}\right)^{s-1}\left(1-\frac{1}{N/M}\right)^{K-s+1} \tag{D.3}$$

for any user k and packet f of file d_k. We have the following chain:

$$\begin{aligned}
&\mathbb{E}_{c_{op}}\left[R^{nd}\left(\mathcal{C},\mathbf{d}_u\right)\right] = \sum_{\mathcal{S}\neq\emptyset}\frac{\mathbb{E}\left[\max\limits_{k\in\mathcal{S}}|V_{k,\mathcal{S}-k}|\right]}{F}\\
&= \sum_{\mathcal{S}\neq\emptyset}\mathbb{E}\left[\frac{\max\limits_{k\in\mathcal{S}}\sum\limits_{f=1}^{F}\mathbf{1}^{\mathcal{S}-k}_{d_k,f}}{F}\right]\\
&\geq \sum_{\mathcal{S}\neq\emptyset}\frac{\Pr\left(\bigcup\limits_{k\in\mathcal{S},f\in[1:F]}\mathbf{1}^{\mathcal{S}-k}_{d_k,f} > 0\right)}{F}\\
&\geq \sum_{\mathcal{S}\neq\emptyset}\frac{\left[\sum\limits_{k\in\mathcal{S},f\in[1:F]}\Pr\left(\mathbf{1}^{\mathcal{S}-k}_{d_k,f} > 0\right) - \sum\limits_{i,j\in\mathcal{S},f_1,f_2\in[1:F],(i,f_1)\neq(j,f_2)}\Pr\left(\mathbf{1}^{\mathcal{S}-i}_{d_i,f_1} > 0\bigcap\mathbf{1}^{\mathcal{S}-j}_{d_j,f_2} > 0\right)\right]}{F}\\
&\overset{a}{\geq} \frac{\sum\limits_{s=1}^{K}\binom{K}{s}\left(sF\mu(s) - \frac{1}{2}s^2(F)^2\left(\mu(s)\right)^2\right)}{F}\\
&= \sum_{s=1}^{K}\binom{K}{s}s\left(\frac{1}{\frac{N}{M}}\right)^{s-1}\left(1-\frac{1}{\frac{N}{M}}\right)^{K-s+1} - \frac{1}{2}\sum_{s=1}^{K}\binom{K}{s}s^2F\left(\mu(s)\right)^2\\
&\overset{b}{=} K\left(1-\frac{1}{N/M}\right) - \frac{1}{2}\sum_{s=1}^{K}\binom{K}{s}s^2F\left(\mu(s)\right)^2\\
&= K\left(1-\frac{1}{\frac{N}{M}}\right) - \frac{1}{2}\left(1-\frac{1}{\frac{N}{M}}\right)^{2K}F\sum_{s=1}^{K}\binom{K}{s}s^2\left(\frac{\frac{1}{\frac{N}{M}}}{1-\frac{1}{\frac{N}{M}}}\right)^{2(s-1)}\\
&\overset{c}{\geq} K\left(1-\frac{M}{N}\right)
\end{aligned}$$

$$
\begin{aligned}
&-\frac{1}{2}F\left(K\left(1-\frac{M}{N}\right)^2+K(K-1)\left(\frac{M}{N}\right)^2\right)\left(\left(1-\frac{M}{N}\right)^2+\left(\frac{M}{N}\right)^2\right)^{(K-1)}\\
&\overset{d}{\geq} K\left(1-\frac{1}{\frac{N}{M}}\right)-\frac{1}{2}F\left(K+t^2\right)\exp\left(-2t\left(1-\frac{t}{K}\right)\left(1-\frac{1}{K}\right)\right)\\
&\overset{e}{\geq} K\left(1-\frac{1}{\frac{N}{M}}\right)-FKt\exp\left(-2t\left(1-\frac{t}{K}\right)\left(1-\frac{1}{K}\right)\right)
\end{aligned}
\tag{D.4}
$$

Justifications are :

(a)- This is from the definition of $\mu(s)$, Theorem D.2.1 and the inequality $\binom{s}{2} \leq \frac{s^2}{2}$.

(b)- This is because: $\sum_{s=1}^{K} sp^{s-1}(1-p)^{K-s+1} = \frac{1-p}{p}\text{mean}(\text{Bi}(K,p)) = K(1-p)$, where $\text{Bi}(n,p)$ is a binomial random variable with n random Bernoulli trials with probability $p < 1$.

(c)- We use the following chain:

$$
\begin{aligned}
&\sum_{s\geq 1}\binom{K}{s}s^2p^{s-1} = \frac{d}{dp}\left[p\frac{d}{dp}\left(1+p\right)^k\right] = \frac{d}{dp}[pk(1+p)^{k-1}] = \frac{d}{dp}[pk(1+p)^{k-1}]\\
&= k(1+p)^{k-1} + kp(k-1)(1+p)^{k-2} \leq (1+p)^{k-1}(k+k(k-1)p)
\end{aligned}
\tag{D.5}
$$

In the above, $p = \left(\frac{\frac{1}{\frac{N}{M}}}{1-\frac{1}{\frac{N}{M}}}\right)$ gives the justification.

(d)- We use the fact that $t = KM/N$ and use the algebraic manipulation $(1-x) \leq \exp(-x)$ with $x = (2t(1-\frac{t}{K}))$.

(e)- This follows from: $K + t^2 \leq 2Kt$.

This implies that when $F \leq \frac{1}{2t}\left(1-\frac{M}{N}\right)\exp\left(2t\left(1-\frac{t}{K}\right)\left(1-\frac{1}{K}\right)\right)$, the expected number of normalized transmissions for Algorithm 2 for distinct requests under the old placement scheme given by Algorithm 1 is at least $\frac{1}{2}\left(1-\frac{M}{N}\right)K$. This implies that there is very little coding gain ($t = K\frac{M}{N}$) even when we have file size exponential in t.

Remark: We can use the exact identical proof in Theorem 5.3.1 with N/M replaced by $\lceil \frac{N}{M} \rceil$ and we get the result in Theorem 5.3.2.

D.3 Proof of Theorem 5.3.3

We show this by contradiction. Let us assume that $\mathbb{E}_{c_{up}}\left(R\left(\mathcal{C}, \mathbf{d}_u\right)\right) \leq \frac{K(1-M/N)}{\frac{4}{3}g}$. This implies (by Markov's Inequality):

$$\Pr_{c_{up}}\left(R\left(\mathcal{C}, \mathbf{d}_u\right) \leq \frac{K(1-M/N)}{g}\right) \geq \frac{1}{4} \tag{D.6}$$

The number of transmissions $R\left(\mathcal{C}, \mathbf{d}_u\right) \leq \frac{K(1-M/N)}{g}$ implies that there is at least one clique of size g in the side information graph G induced by $\mathcal{C}$ and $\mathbf{d}_u$. Given cache configuration $\mathcal{C}$ and distinct demands $\mathbf{d}_u$, let n_g denote the number of distinct cliques of size g. So we have the following chain of inequalities:

$$\begin{aligned}
\Pr_{c_{up}}\left(R\left(\mathcal{C}, \mathbf{d}_u\right) \leq \frac{K(1-M/N)}{g}\right) &\leq \Pr_{c_{up}}\left(\text{there is one clique of size } g\right) \\
&\overset{a}{\leq} \mathbb{E}_{c_{up}}\left(n_g\right) \\
&\overset{b}{\leq} \binom{K}{g} F^g \left(\frac{M}{N}\right)^{g(g-1)} \\
&\overset{c}{\leq} \left(\frac{Ke}{g}\right)^g F^g \left(\frac{M}{N}\right)^{g(g-1)} \\
&\leq \left(\frac{Ket}{gt}\right)^g F^g \left(\frac{t}{K}\right)^{g(g-1)} \\
&\leq \left(\frac{etF}{g}\right)^g \left(\frac{t}{K}\right)^{g(g-2)}
\end{aligned} \tag{D.7}$$

When $F < \frac{g}{2et}\left(\frac{N}{M}\right)^{g-2}$ and $g > 2$, then probability given by (D.7) is strictly less than 1/4 contradicting the assumption. Therefore, the desired implication follows. Justifications are: (a) $Pr\left(X \geq 1\right) \leq \mathbb{E}[X]$. (b) There are $\binom{K}{g}$ ways of choosing g users caches. Since all demands are distinct, there are F^g ways of choosing g file packets belonging to the files requested by the chosen users. $(M/N)^{g-1}$ is the probability that a file packet wanted by one of the users is present in $g-1$ other user caches. Since the demands are distinct and placement of packets belonging to different files are different, the probability of forming a g-clique is given by $(M/N)^{g(g-1)}$. (c) $\binom{K}{g} \leq \left(\frac{Ke}{g}\right)^g$.

Note: We would like to note that c_{up} represents a broad set of schemes where every file packet is placed in a cache independently of its placement elsewhere and no file packet is given undue importance over other packets belonging to the same file.

D.4 Proof of Theorem 5.4.1

We need the following lemma from [118] (see proof of Theorem 1).

Lemma D.4.1. *[118] Consider m balls being thrown randomly uniformly and independently into n bins. When $m = r(n)n\log n$ where $r(n)$ is $\Theta((log(n))^p)$ for some positive integer p, then maximum number of balls in any bin is at most $r(n)\log n(1+2\frac{\sqrt{2}}{r(n)})$ with probability at least $1-\frac{1}{n^2}$.*

According to the placement scheme given by Algorithm 3, every file is made up of F' groups of file packets. Each group has size $\lceil N/M \rceil$. Let us consider the j-th packet of every group. There are F' such file packets. We will first analyze assuming that algorithm 5 uses only the F' file packets formed by considering only the j-th packet from every group. We will finally add up the number of transmissions for every set of F' packets formed using the differently numbered packet (for all $j \in [1 : \lceil N/M \rceil]$) from every group. Clearly, this is suboptimal. Therefore, this upper bounds the performance of Algorithm 5.

Consider a file n. Let G_j^n be the set of F' packets, each of which is the j-th packet from every group of file n according to the groups formed during placement algorithm 3. Let $S_{n,f,j} \subseteq [1 : K]$ be the subset of user caches where the f-th packet in G_j^n is stored. Here, $1 \leq f \leq F'$ indicates the position among F' packets formed by taking the jth packet from very group. Given a user cache k, the placement of packets from the set G_j^n are mutually independent of each other. The marginal probability of placing it is given by $\frac{1}{\lceil N/M \rceil}$. The placement is also independent across caches. Therefore, the number of user caches in which a particular packet in G_j^n is placed is a binomial random variable $\mathrm{Bi}\left(K, \frac{1}{\lceil N/M \rceil}\right)$ where $\mathrm{Bi}\,(m, p)$ is a binomial distribution with m independent trails each with probability p. Therefore, by chernoff bounds (see Pg. 276 [119]), $\Pr\left(|S_{n,f,j}| < g\right) \leq \exp\left(-\frac{K}{\lceil N/M \rceil}\left(1 - \frac{g\lceil N/M \rceil}{K}\right)^2\right) \leq \exp\left(-\frac{4K}{9\lceil N/M \rceil}\right)$. Here, we have used the fact that $g \leq \frac{K}{3\lceil N/M \rceil}$. Therefore, for any j (by Markov's Inequality),

$$\Pr\left(\sum_{f=1}^{F'} \mathbf{1}_{|S_{n,f,j}|<g} > 3F'(g+1)K\lceil N/M \rceil \exp\left(-\frac{4K}{9\lceil N/M \rceil}\right)\right) \leq \frac{1}{3(g+1)K\lceil N/M \rceil} \tag{D.8}$$

$\lceil N/M \rceil \leq \frac{K}{\frac{27}{4}\log K}$ and $g \leq \frac{K}{3\lceil N/M \rceil}$ implies the following condition (which can be verified by algebra):

$$(g+1)K^2\lceil N/M \rceil < \exp\left(\frac{4K}{9\lceil N/M \rceil}\right). \tag{D.9}$$

If a file bit is stored in p caches, then the file packet is said to be on level p. This implies, that with high probability, $\left(1 - 3(g+1)K\lceil N/M \rceil \exp\left(-\frac{4K}{9\lceil N/M \rceil}\right)\right)F'$ file packets belonging to file n from G_j^n is stored at a level above or equal to g. We will first compute the number of transmissions due to applying Algorithm 5 only on the file packets in $\{d_k, f, j\}_{1 \leq k \leq K, f \in [1:F']}$ for a particular j.

We start by considering a fixed demand pattern $\mathbf{d} = \{d_1, d_2 \dots d_K\}$. Applying union bound with (D.8) over at most K files in the demand $\mathbf{d}$, we have:

$$\Pr\left(\exists k \in [1:K] : \sum_{f=1}^{F'} \mathbf{1}_{|S_{d_k,f,j}| < g} > 3(g+1)F'K\lceil N/M \rceil \exp\left(-\frac{4K}{9\lceil N/M \rceil}\right)\right) \\ \leq \frac{1}{3(g+1)\lceil N/M \rceil} \tag{D.10}$$

Now, consider Algorithm 5. The first few steps of the algorithm, denoted henceforth as ‘pull down’ phase, brings every file packet stored above level g to level g. Consider a file packet (d_k, f, j) before the beginning of Algorithm 5. Given that the packet (d_k, f, j) is at a level above g, after the ‘pull down’ phase, the probability that it occupies any of the $\binom{K}{g}$ subsets is equal. This is because prior to the pull down phase, the probability that the file packet being stored in a particular cache is independent and equal to $\frac{1}{\lceil N/M \rceil}$. Consider the F' file packets $\{(d_k, f, j)\}$, $1 \leq f \leq F'$. Clearly, the probability of any one of them (say (d_k, f, j)) occupying a given set of g caches, after the pull down phase, is independent of the occupancy of all other file packets $\{(d_k, f', j)\}_{f' \neq f}$. Let $S^a_{d_k,f,g}$ denote the occupancy after the pull down phase. Therefore after the pull down phase, which is applied only to the files in the demand vector $\mathbf{d}$,

$$\begin{aligned} \Pr\left(S^a_{d_k,f,j} = B \mid \ |S^a_{d_k,f,j}| > g, \{S^a_{d_k,f',g}\}_{f \neq f'}\right) &= \frac{1}{\binom{K}{g}} \\ \forall B \subseteq \binom{[1:K]}{g},\ k \in [1:K],\ 1 \leq j \leq \lceil N/M \rceil & \end{aligned} \tag{D.11}$$

After the pull down phase in Algorithm 5, we compute the number of transmissions of Algorithm 4 using the modified $S^a_{d_k,f,j}$ after the pull down phase. It has been observed that Algorithm 4 is equivalent to Algorithm 2. After the pull down phase, all the files packets are present at file level g or below. Let us set $F' = c\binom{K}{g}\left(\log(\binom{K}{g})\right)^2$ for some constant $c > 0$. After the pull down phase, let $V^j_{k,\mathcal{S}-k}$ be the set of file packets in $G^{d_k}_j$ requested by user k but stored exactly in the cache of users specified by $\mathcal{S} - k$. With respect to only the file packets $\bigcup\limits_{k \in [1:K]} G^{d_k}_j$, the number of transmissions of Algorithm 4 is given by:

$$\begin{aligned} \text{No. of trans}(j) &= \sum_{\mathcal{S} \neq \emptyset} \frac{\max\limits_{k \in \mathcal{S}} |V^j_{k,\mathcal{S}-k}|}{F'} \\ &\overset{a}{=} \sum_{\mathcal{S} \neq \emptyset, |\mathcal{S}| \leq g+1} \frac{\max\limits_{k \in \mathcal{S}} |V^j_{k,\mathcal{S}-k}|}{F'} = \sum_{|\mathcal{S}| = g+1} \frac{\max\limits_{k \in \mathcal{S}} |V^j_{k,\mathcal{S}-k}|}{F'} + \sum_{|\mathcal{S}| \leq g} \frac{\max\limits_{k \in \mathcal{S}} |V^j_{k,\mathcal{S}-k}|}{F'} \end{aligned} \tag{D.12}$$

(a)- This is because after the pull down phase, all the relevant file packets are at a level at most g. Consider the event E that $b = \left(1 - 3(g+1)K\lceil \frac{N}{M} \rceil \exp\left(-\frac{4K}{9\lceil N/M \rceil}\right)\right) F'$ bits of $G^{d_i}_j$ for all i are stored at a level above g before the beginning of Algorithm 5. Conditioned on this event being true, by (D.11), the pull down phase is equivalent to throwing b balls independently and uniformly randomly into $\binom{K}{g}$ bins. Using (D.9) and the fact that $F' = c\binom{K}{g}(\log(\binom{K}{g}))^2$, the pull down phase is akin to throwing $m = (1 - 3(g+1)K\lceil\frac{N}{M}\rceil \exp\left(-\frac{4K}{9\lceil N/M\rceil}\right))F' \geq c\left(1 - \frac{3(g+1)}{K}\right)(\log n)(n\log n)$ balls into $n = \binom{K}{g}$ bins. In fact, the m balls of file d_k are being thrown independently and uniformly randomly into bins satisfying $\mathcal{S} - k : |\mathcal{S}| = g+1,\ k \in \mathcal{S}$. We apply, Lemma D.4.1 for a particular user k to obtain:

$$\Pr\left(\max_{\mathcal{S}:|\mathcal{S}|=g+1,\ k\in\mathcal{S}} \frac{|V^j_{k,\mathcal{S}-k}|}{F'} \geq \frac{\frac{m}{n}\left(1 + O\left(\frac{1}{\log K}\right)\right)}{F'}|E\right) \leq \frac{1}{\binom{K}{g}^2} \tag{D.13}$$

Please note that $r(n)$ as in Lemma D.4.1 is $O(\log K)$. Now, applying a union bound over all users k to (D.13), we have:

$$\Pr\left(\exists k \in [1:K] : \max_{\mathcal{S}:|\mathcal{S}|=g+1,\ k\in\mathcal{S}} \frac{|V^j_{k,\mathcal{S}-k}|}{F'} \geq \frac{\frac{m}{n}\left(1 + O\left(\frac{1}{\log K}\right)\right)}{F'}|E\right) \leq \frac{K}{\binom{K}{g}^2} \tag{D.14}$$

This implies that all $V_{k,\mathcal{S}-k}$ are bounded in size. Therefore, we have the following:

$$\begin{aligned} 1 - \frac{K}{\binom{K}{g}^2} &\leq \Pr\left(\sum_{|\mathcal{S}|=g+1} \frac{\max_{k\in\mathcal{S}}|V^j_{k,\mathcal{S}-k}|}{F'} \leq \binom{K}{g+1}\frac{\frac{m}{n}\left(1 + O\left(\frac{1}{\log K}\right)\right)}{F'}|E\right) \\ &\overset{a}{=} \Pr\left(\sum_{|\mathcal{S}|=g+1} \frac{\max_{k\in\mathcal{S}}|V^j_{k,\mathcal{S}-k}|}{F'} \leq \frac{K-g}{g+1}\left(1 + O\left(\frac{1}{\log K}\right)\right)|E\right). \end{aligned} \tag{D.15}$$

(a) is because: $1 \geq \frac{m}{F'} \geq 1 - \frac{3(g+1)}{K}$ implying $\frac{m}{F'}(1 + \frac{1}{O(\log K)}) = (1 + \frac{1}{O(\log K)})$. Putting together (D.15), (D.12) and (D.10), we have:

$$\begin{aligned} \Pr\Bigg(\text{No. of trans}(j) &\leq \frac{K-g}{g+1}\left(1 + O\left(\frac{1}{\log K}\right)\right) + 2K^2\lceil\frac{N}{M}\rceil e^{-\frac{4K}{9\lceil N/M\rceil}}\Bigg) \\ &\geq \left(1 - \frac{1}{2(g+1)\lceil N/M\rceil}\right)\left(1 - \frac{1}{\binom{K}{g}^2}\right) \end{aligned} \tag{D.16}$$

Union bounding over all $1 \leq j \leq \lceil N/M \rceil$, we have:

$$\Pr\left(\exists j : \text{No. of trans}(j) > \frac{K-g}{g+1}\left(1 + O\left(\frac{1}{\log K}\right)\right) + 3(g+1)K^2\lceil\frac{N}{M}\rceil e^{-\frac{4K}{9\lceil N/M\rceil}}\right) \leq \frac{1}{3(g+1)} + \frac{\lceil N/M \rceil}{\binom{K}{g}^2} \tag{D.17}$$

From (D.9), we have $3(g+1)K^2\lceil\frac{N}{M}\rceil e^{-\frac{4K}{9\lceil N/M\rceil}} < 3$. Now combining transmissions for different j and normalizing by $\lceil N/M \rceil$, we have:

$$\Pr^{md}_{\mathcal{C}_{\mathrm{np}}}\left(R^{md}\left(\mathfrak{C}, \mathbf{d}\right) > \frac{K-g}{g+1}(1+o(1))\right) \leq \frac{1}{3(g+1)} + \frac{\lceil N/M \rceil}{\binom{K}{g}^2} = \frac{1}{3(g+1)} + O(1/K) \tag{D.18}$$

In the above bad event, the number of transmissions (normalized) needed is at most K. Therefore, we have:

$$\begin{aligned}\mathbb{E}^{md}_{\mathcal{C}_{\mathrm{np}}}\left[R^{md}\left(\mathfrak{C}, \mathbf{d}\right)\right] &\leq \frac{K-g}{g+1}(1+o(1))(1 - \frac{1}{3(g+1)} - O\left(\frac{1}{K}\right)) + \left(\frac{1}{3(g+1)} + O(1/K)\right)K \\ &\leq \frac{4}{3}\frac{K}{g+1}(1+o(1))\end{aligned} \tag{D.19}$$

D.5 Proof of Theorem 5.4.2

Essentially, Theorem 5.4.1 is applied to all groups of size $K' = \lceil N/M \rceil 3g \log(N/M)$ and K' satisfies the conditions of Theorem 5.4.1. Adding up the contributions of various groups, we obtain the result stated in the theorem.

Appendix E

Proofs for Chapter 6

E.1 Proof of Lemma 6.3.1

The conditional version of Fano's lemma (see [90, Lemma 9]) yields:

$$\mathbb{E}_{\mu}\left[\Pr\left(\hat{G} \neq G\right)\middle| G \in \mathcal{T}\right] \geq \frac{H(G|G \in \mathcal{T}) - I(G;X^n|G \in \mathcal{T}) - \log 2}{\log|\mathcal{T}|} \tag{E.1}$$

Now,

$$\begin{aligned} p_{avg} &= \mathbb{E}_{\mu}\left[\Pr\left(\hat{G} \neq G\right)\right] \\ &= \Pr_{\mu}\left(G \in \mathcal{T}\right)\mathbb{E}_{\mu}\left[\Pr\left(\hat{G} \neq G\right)\middle| G \in \mathcal{T}\right] + \Pr_{\mu}\left(G \notin \mathcal{T}\right)\mathbb{E}_{\mu}\left[\Pr\left(\hat{G} \neq G\right)\middle| G \notin \mathcal{T}\right] \\ &\overset{a}{\geq} \Pr_{\mu}\left(G \in \mathcal{T}\right)\mathbb{E}_{\mu}\left[\Pr\left(\hat{G} \neq G\right)\middle| G \in \mathcal{T}\right] \\ &\overset{b}{\geq} \mu\left(\mathcal{T}\right)\frac{H(G|G \in \mathcal{T}) - I(G;X^n|G \in \mathcal{T}) - \log 2}{\log|\mathcal{T}|} \end{aligned} \tag{E.2}$$

Here we have: (a) since both terms in the equation before are positive. (b) by using the conditional Fano's lemma.

Also, note that:

$$\begin{aligned} \mathbb{E}_{\mu}\left[\Pr\left(\hat{G} \neq G\right)\middle| G \in \mathcal{T}\right] &= \sum_{G \in \mathcal{T}} \Pr_{\mu}\left(G|G \in \mathcal{T}\right).\Pr\left(\hat{G} \neq G\right) \\ &\leq \max_{G \in \mathcal{T}} \Pr\left(\hat{G} \neq G\right) \\ &\leq \max_{G \in \mathcal{G}} \Pr\left(\hat{G} \neq G\right) = p_{\max} \end{aligned} \tag{E.3}$$

E.2 Proof of Corollary 6.3.1

We pick μ to be a uniform measure and use $H(G) = \log|\mathcal{G}|$. In addition, we upper bound the mutual information through an approach in [92] which relates it to coverings in terms of the KL-divergence as follows:

$$\begin{aligned}
I(G; X^n | G \in \mathcal{T}) &\overset{a}{=} \sum_{G \in \mathcal{T}} P_\mu(G|G \in \mathcal{T}) D\left(f_G(\mathbf{x}^n) \| f_X(\mathbf{x}^n)\right) \\
&\overset{b}{\leq} \sum_{G \in \mathcal{T}} P_\mu(G|G \in \mathcal{T}) D\left(f_G(\mathbf{x}^n) \| Q(\mathbf{x}^n)\right) \\
&\overset{c}{=} \sum_{G \in \mathcal{T}} P_\mu(G|G \in \mathcal{T}) D\left(f_G(\mathbf{x}^n) \middle\| \sum_{G' \in C_{\mathcal{T}}(\epsilon)} \frac{1}{|C_{\mathcal{T}}(\epsilon)|} f_{G'}(\mathbf{x}^n)\right) \\
&= \sum_{G \in \mathcal{T}} P_\mu(G|G \in \mathcal{T}) \sum_{\mathbf{x}^n} f_G(\mathbf{x}^n) \log\left(\frac{f_G(\mathbf{x}^n)}{\sum\limits_{G' \in C_{\mathcal{T}}(\epsilon)} \frac{1}{|C_{\mathcal{T}}(\epsilon)|} f_{G'}(\mathbf{x}^n))}\right) \\
&\overset{d}{\leq} \log|C_{\mathcal{T}}(\epsilon)| + n\epsilon
\end{aligned} \tag{E.4}$$

Here we have: (a) $f_X(\cdot) = \sum_{G \in \mathcal{T}} P_\mu(G|G \in \mathcal{T}) f_G(\cdot)$. (b) $Q(\cdot)$ is any distribution on $\{-1, 1\}^{np}$ (see [92, Section 2.1]). (c) by picking $Q(\cdot)$ to be the average of the set of distributions $\{f_G(\cdot), G \in C_{\mathcal{T}}(\epsilon)\}$. (d) by lower bounding the denominator sum inside the log by only the covering element term for each $G \in \mathcal{T}$. Also using $D\left(f_G(\mathbf{x}^n) \| f_{G'}(\mathbf{x}^n)\right) = nD\left(f_G \| f'_G\right)$ $(\leq n\epsilon)$, since the samples are drawn i.i.d.

Plugging these estimates in Lemma 6.3.1 gives the corollary.

E.3 Proof of Lemma 6.3.2

Consider a graph $G(V, E)$ with two nodes a and b such that there are at least d node disjoint paths of length at most ℓ between a and b. Consider another graph $G'(V, E')$ with edge set $E' \subseteq E$ such that E' contains only edges belonging to the d node disjoint paths of length ℓ between a and b. All other edges are absent in E'. Let $\mathcal{P}$ denote the set of node disjoint paths. By Griffith's inequality (see [120, Theorem 3.1]),

$$\begin{aligned} \mathbb{E}_{f_G}\left[x_a x_b\right] &\geq \mathbb{E}_{f_{G'}}\left[x_a x_b\right] \\ &= 2P_{G'}\left(x_a x_b = +1\right) - 1 \end{aligned} \tag{E.5}$$

Here, $P_{G'}(.)$ denotes the probability of an event under the distribution $f_{G'}$.

We will calculate the ratio $P_{G'}\left(x_a x_b = +1\right) / P_{G'}\left(x_a x_b = -1\right)$. Since we have a *zero-field* ising model (*i.e.* no weight on the nodes), $f_{G'}(\mathbf{x}) = f_{G'}(-\mathbf{x})$. Therefore, we have:

$$\frac{P_{G'}\left(x_a x_b = +1\right)}{P_{G'}\left(x_a x_b = -1\right)} = \frac{2P_{G'}\left(x_a = +1, x_b = +1\right)}{2P_{G'}\left(x_a = -1, x_b = +1\right)} \tag{E.6}$$

Now consider a path $p \in \mathcal{P}$ of length ℓ_p whose end points are a and b. Consider an edge (i, j) in the path p. We say i, j disagree if x_i and x_j are of opposite signs. Otherwise, we say they agree. When $x_b = +1$, x_a is $+1$ iff there are even number of disagreements in the path p. Odd number of disagreements would correspond to $x_a = -1$, when $x_b = +1$. The location of the disagreements exactly specifies the signs on the remaining variables, when $x_b = +1$. Let $d(p)$ denote the number of disagreements in path p. Every agreement contributes a term $\exp(\lambda)$ and every disagreement contributes a term $\exp(-\lambda)$. Now, we use this to bound (E.6) as follows:

$$\frac{P_{G'}\left(x_a x_b = +1\right)}{P_{G'}\left(x_a x_b = -1\right)} \overset{a}{=} \frac{\prod_{p \in \mathcal{P}} \left(\sum_{d(p) \text{ even}} e^{\lambda \ell_p} e^{-2\lambda d(p)} \right)}{\prod_{p \in \mathcal{P}} \left(\sum_{d(p) \text{ odd}} e^{\lambda \ell_p} e^{-2\lambda d(p)} \right)}$$

$$\overset{b}{=} \frac{\prod_{p \in \mathcal{P}} \left((1 + e^{-2\lambda})^{\ell_p} + (1 - e^{-2\lambda})^{\ell_p} \right)}{\prod_{p \in \mathcal{P}} \left((1 + e^{-2\lambda})^{\ell_p} - (1 - e^{-2\lambda})^{\ell_p} \right)} \overset{c}{=} \frac{\prod_{p \in \mathcal{P}} \left(1 + (\tanh(\lambda))^{\ell_p} \right)}{\prod_{p \in \mathcal{P}} \left(1 - (\tanh(\lambda))^{\ell_p} \right)} \tag{E.7}$$

$$\overset{d}{\geq} \frac{\left(1 + (\tanh(\lambda))^{\ell}\right)^d}{\left(1 - (\tanh(\lambda))^{\ell}\right)^d} \tag{E.8}$$

Here we have: (a) by the discussion above regarding even and odd disagreements. Further, the partition function Z (of $f_{G'}$) cancels in the ratio and since the paths are disjoint, the marginal splits as a product of marginals over each path. (b) using the binomial theorem to add up the even and odd terms separately. (c) $\ell_p \leq \ell$, $\forall p \in \mathcal{P}$. (d) there are d paths in $\mathcal{P}$.

Substituting in (E.5), we get:

$$\mathbb{E}_{f_G}\left[x_a x_b\right] \geq 1 - \frac{2}{1 + \frac{\left(1+(\tanh(\lambda))^{\ell}\right)^d}{\left(1-(\tanh(\lambda))^{\ell}\right)^d}}. \tag{E.9}$$

E.4 Proof of Corollary 6.3.2

From Eq. (6.3), we get:

$$\begin{aligned}
D\left(f_G \| f_{G'}\right) &\leq \sum_{(s,t)\in E-E'} \lambda\left(\mathbb{E}_G\left[x_s x_t\right] - \mathbb{E}_{G'}\left[x_s x_t\right]\right) + \sum_{(s,t)\in E'-E} \lambda\left(\mathbb{E}_{G'}\left[x_s x_t\right] - \mathbb{E}_G\left[x_s x_t\right]\right) \\
&\overset{a}{\leq} \sum_{(s,t)\in E-E'} \lambda\left(1 - \mathbb{E}_{G'}\left[x_s x_t\right]\right) + \sum_{(s,t)\in E'-E} \lambda\left(1 - \mathbb{E}_G\left[x_s x_t\right]\right) \\
&\overset{b}{\leq} \frac{2\lambda|E-E'|}{1 + \frac{\left(1+(\tanh(\lambda))^{\ell}\right)^d}{\left(1-(\tanh(\lambda))^{\ell}\right)^d}} + \frac{2\lambda|E'-E|}{1 + \frac{\left(1+(\tanh(\lambda))^{\ell}\right)^d}{\left(1-(\tanh(\lambda))^{\ell}\right)^d}} \qquad \text{(E.10)}
\end{aligned}$$

Here we have: (a) $\mathbb{E}_G\left[x_s x_t\right] \leq 1$ and $\mathbb{E}_{G'}\left[x_s x_t\right] \leq 1$ (b) for any $(s,t) \in E - E'$, the pair of nodes are (ℓ, d) connected. Therefore, bound on $\mathbb{E}_{G'}\left[x_s x_t\right]$ from Lemma 6.3.2 applies. Similar bound holds for $\mathbb{E}_G\left[x_s x_t\right]$ for $(s,t) \in E' - E$.

Appendix F

Proofs for Chapter 7

F.1 Proof of Lemma 7.3.1

We describe a string labeling procedure as follows to label elements of the set $[1 : n]$.

String Labelling: Let $a > 1$ be a positive integer. Let x be the integer such that $a^x < n \leq a^{x+1}$. $x + 1 = \lceil \log_a n \rceil$. Every element $j \in [1 : n]$ is given a label $L(j)$ which is a string of integers of length $x + 1$ drawn from the alphabet $\{0, 1, 2 \ldots a\}$ of size $a + 1$. Let $n = p_d a^d + r_d$ and $n = p_{d-1} a^{d-1} + r_{d-1}$ for some integers $p_d, p_{d-1}, r_d, r_{d-1}$, where $r_d < a^d$ and $r_{d-1} < a^{d-1}$. Now, we describe the sequence of the d-th digit across the string labels of all elements from 1 to n:

1. Repeat 0 a^{d-1} times, repeat the next integer 1 a^{d-1} times and so on circularly [1] from $\{0, 1 \ldots a - 1\}$ till $p_d a^d$.

2. After that, repeat 0 $\lceil r_d / a \rceil$ times followed by 1 $\lceil r_d / a \rceil$ times till we reach the nth position. Clearly, n-th integer in the sequence would not exceed $a - 1$.

3. Every integer occurring *after* the position $a^{d-1} p_{d-1}$ is increased by 1.

[1] Circular means that after $a - 1$ is completed, we start with 0 again.

From the three steps used to generate every digit, a straightforward calculation shows that every integer letter is repeated at most $\lceil n/a \rceil$ times in every digit i in the string. Now, we would like to prove inductively that the labels are distinct for all n elements. Let us assume the induction hypothesis: For all $n < a^{q+1}$, the labels are distinct. The base case of $q = 0$ is easy to see. Then, we would like to show that for $a^{q+1} \leq n < a^{q+2}$, the labels are distinct.

Another way of looking at the labeling procedure is as follows. Let $n = a^{q+1}p + r$ with $r < a^{q+1}$. Divide the label matrix L (of dimensions $(q+2) \times n$) into two parts, one L_1 consisting of the first pa^{q+1} columns and the other L_2 consisting of the remaining columns. The first $q+1$ rows of L_1 is nothing but the string labels for all numbers from 0 to pa^{q+1} expressed in base a. For any row $i \leq \lceil \log_a r \rceil$ in the original matrix L of labels, till the end of first pa^{q+1} columns, the labeling procedure would be still in Step 1. After that, one can take r to be the new size of the set of elements to be labelled and then restart the procedure with this r. Therefore we have the following *key* observation: $L_2(1 : \lceil \log_a r \rceil, :)$ (the matrix with first $\lceil \log_a r \rceil$ rows of L_2) is nothing but the label matrix for r distinct elements from the above labeling procedure.

Since, $r < a^{q+1}$, by the induction hypothesis, the columns are distinct. Hence, any two columns in L_2 are distinct. Suppose the first $q + 1$ rows of two columns b and c of L_1 are identical. These correspond to base a expansion of $b - 1$ and $c - 1$. They are separated by at least $a^{q+1} + 1$ columns. But the last row of columns b and p in L_1 has to be distinct because according to Step 2 and Step 3 of the labeling procedure, in the $q + 2^{th}$ row, every integer is repeated at most

$\lceil n/a \rceil \leq a^{q+1}$ times continuously, and only once. Therefore, any two columns in L_1 are distinct. The last row entries in L_1 are different from L_2 because of the addition in Step 3. Therefore, all columns of L are distinct. Hence, by induction, the result is shown.

F.2 Proof of Theorem 7.3.1

By Lemma 7.3.1, ith place has at most $\lceil \frac{n}{\lceil n/k \rceil} \rceil \leq k$ occurrences of symbol j. Therefore, $|S_{i,j}| \leq k$. Now, consider the pair of distinct elements $p, q \in [1:n]$. Since they are labelled distinctly (Lemma 7.3.1), there is at least one place i in their string labels where they differ. Suppose the distinct ith letters are $a, b \in \mathcal{A}$, $a \neq b$ and let us say $a \neq 0$ without loss of generality. Then, clearly the separation criterion is met by the subset $S_{i,a}$. This proves the claim.

F.2.1 Proof of Theorem 7.3.2

We construct a worst case σ inductively. Before every step m, the adaptive algorithm deterministically chooses I_m based on $\mathcal{E}_{\{I_1, I_2 \ldots I_{m-1}\}}(K_n)$. Therefore, we will reveal a partial order $\sigma^{(m-1)}$ to satisfy the observations so far. Inductively for every m, we will make sure that after I_m is chosen by the algorithm, further details about σ can be revealed to form $\sigma^{(m)}$ such that after intervening on I_2 and then applying R0, we will make sure there is no opportunity to apply the rule $R2$. This would make sure that $\mathcal{I}$ is a separating system on n elements.

Before intervention at any step m, let us 'tag' every vertex i using a subset $C_i^{(m-1)} \subseteq [1:m]$ such that $C_i^{(m-1)} = \{p : i \in I_p,\ p \leq m-1\}$. $C_i^{(m-1)}$ contains indices of all those interventions that contain vertex i before step m. Let $\mathcal{C}^{(m-1)}$ contain distinct elements of the multi-set $\{C_i^{(m-1)}\}$ .We will construct σ partially such that it satisfies the following criterion always:

Inductive Hypothesis: The partial order $\sigma^{(m-1)}$ is such that for any two elements i, j with C_i and C_j, i and j are incomparable if $C_i = C_j$ and comparable otherwise. This means the edges between the elements tagged with the same tag C has not been revealed, and thus the relevant directed edges are not known by the algorithm.

Now, we briefly digress to argue that if we could construct $\sigma^{(1)}, \sigma^{(2)} \ldots$ satisfying such a property throughout, then clearly all vertices must be tagged differently otherwise the directions among the vertices that are tagged similarly cannot be learned by the algorithm. Therefore, the algorithm has not succeeded in its task. If all vertices are tagged differently, then it means it is a separating system.

Construction of $\sigma^{(m)}$: We now construct $\sigma^{(m)}$ that can be shown to satisfy the induction hypothesis before step $m+1$. Before step m , consider the vertices in $C \in \mathcal{C}^{(m-1)}$ for any C. Let the current intervention be I_m chosen by the deterministic algorithm. We make the following changes: Modify $\sigma^{(m-1)}$ such that vertices in $I_m \bigcap C$ come before $(I_m)^c \bigcap C$ in the partial order $\sigma^{(m)}$ (vertices inside either sets are still not ordered amongst themselves) in the ordering and clearly the directions between these two sets are revealed by R0. By the induction hypothesis for step m and with the new tagging of vertices into $\mathcal{C}^{(m)}$, it is easy to see that only directions between distinct $C's$ in the new $\mathcal{C}^{(m)}$ have been revealed and all directions within a tag set C are not revealed and all vertices in a tag set are contiguous in the ordering so far. We need to only show that rule R2 cannot reveal anymore edges amongst vertices in $C \in \mathfrak{C}^{(m)}$ after the new $\sigma^{(m)}$ and intervention I_m. Suppose there are two vertices i, j such that just after intervention I_m and the modified $\sigma^{(m)}$, they are tagged identically and application of R2 reveals the direction between i and j before the next intervention. Then there has to be a vertex k tagged differently from i, j such that $j \rightarrow k$ and $k \rightarrow i$ are both known. But this implies that j and i are comparable in $\sigma^{(m)}$ leading to a contradiction. This implies the hypothesis holds for step $m + 1$.

Base case: Trivially, the induction hypothesis holds for step 0 where $\sigma^{(0)}$ leaves the entire set unordered.

F.2.2 Proof of Lemma 7.4.1

The proof is a direct obvious consequence of acyclicity, non-existence of immoralities and the definition of rule R1.

F.3 Proof of Lemma 7.4.2

By Lemma 7.4.1, it is sufficient for an algorithm to identify the root node of the tree. Suppose the root node is b unknown to the algorithm. Every tree has a single vertex separator that partitions the tree into components each of which has size at most $\frac{2}{3}n$ [121]. Choose that vertex separator a_1 (it can be found in by removing every node and determining the components left). If it is a root node we stop here. Otherwise, its parent p_1 (if it is not) after application of rule R0 is identified. Let us consider component trees $T_1, T_2 \ldots T_k$ that result by removing node a_1. Let T_1 contain p_1. All directions in all other trees are known after repeated application of R1 on the original tree after R0 is applied. Directions in T1 will not be known. For the next step, $\mathcal{E}(T_1)$ is the new skeleton which has no immoralities. Again, we find the best vertex separator a_2 and the process continues. This procedure will terminate at some step j when $a_j = b$ or there is only one node left which should be b by Lemma 7.4.1. Since the number of nodes reduce by about $1/3$ at least each time, and initially it can be at most n, this procedure terminates in at most $O(\log_2 n)$ steps.

F.4 Proof of Lemma 7.4.3

The graph induced by two colors classes in any graph is a bi-partite graph and bi-partite graphs do not have odd induced cycles. Since the graph and any induced subgraph is chordal, it implies the induced graph on a pair of color classes does not have a cycle. This proves the theorem.

F.5 Proof of Theorem 7.3.4

Assume n is even for simplicity. We define a family of partial order $\sigma^{(p)}$ as follows: Group $i, i+1$ into C_i. Ordering among i and $i+1$ is not revealed. But all the edges between C_i and C_j for any $j > i$ are directed from C_i to C_j. Now, one has to design a set of interventions such that exactly one node among every C_i is intervened on at least once. This is because, if neither i nor $i+1$ in C_i are intervened on, then the direction between i and $i+1$ cannot be figured out by applying rule R2 on any other set of directions in the rest of the graph. Since the size of every intervention is at most k and at least $n/2$ nodes need to be covered by intervention sets, the number of interventions required is at least $\frac{n}{2k}$.

F.6 Proof of Theorem 7.3.5

Separate n vertices arbitrarily into $\frac{n}{k}$ disjoint subsets C_i of size-k. Let the first n/k interventions $\{I_1, I_2, ..., I_{n/k}\}$ be such that $I_i(v) = 1$ if and only if $v \in C_i$. This divides the problem of learning a clique of size n into learning n/k cliques of size k. Then, we can apply the clique learning algorithm in [101] as a black box to each of the $\frac{n}{k}$ blocks: Each block is learned with probability k^{-c} after $\log c \log k$ experiments in expectation. For $k = cn^r$, choose $c > 1/r - 1$. Then the union bound over n/k blocks yields probability polynomial in n. Since each block takes $\mathcal{O}(\log \log k)$ experiments, we need $\frac{n}{k}\mathcal{O}(\log \log k)$ experiments.

F.7 Proof of Theorem 7.4.1

We need the following definitions and some results before proving the theorem.

Definition F.7.1. *A perfect elimination ordering $\sigma_p = \{v_1, v_2 \ldots v_n\}$ on the vertices of an undirected chordal graph G is such that for all i, the induced neighborhood of v_i on the subgraph formed by $\{v_1, v_2 \ldots v_{i-1}\}$ is a clique.*

Lemma F.7.1. *([98]) If all directions in the chordal graph are according to perfect elimination ordering (edges go only from vertices lower in the order to higher in the* *order), then there are no immoralities.*

We make the following observation: Let the directions in a graph D be oriented according to an ordering σ on the vertices. If a clique comes first in the ordering, then the knowledge of edge directions in the rest of the graph, excluding that of the clique, cannot help at any stage of the intervention process on the clique; because all the edges are directed outwards from the clique and hence none of the Meek rules apply. This is because, if $a \to b$ is to be inferred by Meek rules from other known directions, then either there has to be a known edge direction into a or b before the inference step. So if one of the directed edges not from the clique was to help in the discovery process, either that edge has to be directed towards a or b (like in Meek rules R1, R2 and R3), or it has to be directed towards c in another $c \to a$ (R4) which belongs to the clique. Both the above cases are not possible.

Lemma F.7.2. *([98]) Let C be a maximum clique of an undirected chordal graph $\mathcal{E}(D)$, then there is an underlying DAG D on the chordal skeleton that is oriented according to a perfect elimination ordering (implying no immoralities), where the clique C occurs first.*

By Lemmas F.7.1, F.7.2 and the observation above, given a chordal skeleton, we can construct a DAG on the skeleton with no immoralities such that the directions of the maximum clique in D cannot be learned by using knowledge of the directions outside. This means that only the intervention sets $\{I_1 \bigcap C, I_2 \bigcap C \ldots\}$ matter for learning the directions on this clique. Therefore inference on the clique is isolated. Hence, all the lower bounds for the clique case transfer to this case and since the size of the largest clique is exactly the coloring number of the chordal skeleton, the theorem follows.

F.8 Proof of Theorem 7.4.2

Example with a feasible solution with $|\mathcal{I}|$ close to the lower bound: Consider a graph G that can be partitioned into a clique of size χ and an independent set α. Such graphs are called split graphs and as $n \to \infty$, the fraction of split graphs to chordal graphs tends to 1. If $\mathcal{E}(D) = G$ where G is a split graph skeleton, it is enough to intervene only on the nodes in the clique and therefore the number of interventions that are needed is that for the clique. It is certainly possible to orient the edges in such a way so as to avoid immoralities, since the graph is chordal.

Example with $|\mathcal{I}|$ which needs to be close to the upper bound: We construct a connected chordal skeleton with independent set α and clique size χ (also coloring number) such that it would require $\frac{\alpha(\chi-1)}{2k}$ interventions at least for any algorithm over a class of orientations.

Consider a line L consisting of vertices $1, 2 \ldots 2\alpha$ such that every node $1 < i < 2\alpha$ is connected to $i-1$ and $i+1$. For, all $1 \leq p \leq \alpha$, consider a clique C_p of size χ which only has nodes $2p-1, 2p$ from the line L. Now assume that the actual orientation of the L is $1 \to 2 \ldots \to 2\alpha$. In every clique, the orientation is partially specified as follows: In every clique C_p, all edges from node $2p-1$ are outgoing. It is very clear that this partial orientation excludes all immoralities. Further, each clique $C_p - \{2p-1\}$ can have any arbitrary orientation out of $\chi - 1$ possible ones in the actual DAG. Now, even if all the specified directions are revealed to the algorithm, the algorithm has to intervene on all α disjoint cliques $\{C_p - \{2p-1\}\}_{p=1}^{\alpha}$ each of size $\chi - 1$ and directions in one clique will not force directions on the others through any of the Meek rules or rule R0. Therefore, the lower bound of $\frac{\alpha(\chi-1)}{2}$ total node accesses (total number of nodes intervened) is implied by Theorem 7.3.4. Given every intervention is of size k, these chordal skeletons with the revealed partial order needs at least $\frac{\alpha(\chi-1)}{2k}$ more experiments.

F.9 Proof of Theorem 7.4.3

We provide the following justifications for the correctness of Algorithm 6.

1. At line 4 of the algorithm, when Meek rules and R0 are applied after every intervention, the intermediate graph G, with unlearned edges, will always be a disjoint union of chordal components (refer to (7.1) and the comments below) and hence a chordal graph.

2. The number of unlearned edges before and after the main while loop in Algorithm 6 reduces by at least one. Every edge in E is incident on two colors and one of the colors is always picked for processing because we use a separating system on the colors. Therefore, one node belonging to some edge has a positive score and is intervened on. The edge direction is learnt through rule R0. Therefore, the algorithm terminates.

3. It identifies the correct $\vec{G}$ because every edge is inferred after some intervention I_t by applying rule R0 and Meek rules as in (7.1) both of which are

correct.

4. the algorithm has polynomial run time complexity because the main while loop ends in time $|E|$.

F.10 Proof of Theorem 7.4.4

When supplied with a $(\chi, \min(k, \lceil \chi/2 \rceil))$ completely separating system of size $R(\chi, \min(k, \lceil \chi/2 \rceil))$, algorithm 6 first chooses a set S of k colors classes to intervene based on the separating system construction. Suppose color class c is chosen for interventions, then all forests induced by pair of colors classes c, c', $c' \notin S$ are considered and a vertex v of color class c is chosen such that maximum number of edges could be identified in the worst case according to the score $P(v, c)$ in algorithm 6.

Now, we are going to force the algorithm to perform sub-optimal decisions as follows and then analyze it. Define an *epoch* to be at most χ runs of the inner for loop at Line 6 using the completely separating system as follows. In the ith run, when color class c belonging to set $S \subseteq [1 : \chi]$ is chosen for intervention, then pick the ith largest index c' in S^c, choose a vertex according to the score with respect to just the forest between c and c', i.e. $P(v, c) = |T(c, c', v)| - \max_{1 \leq j \leq d(i)} |T_j|$ without considering all other pairs of color classes.

Now, we will weaken the algorithm further as follows. Consider two color classes c and c' with $c \neq c'$. Since a completely separating system is used, for every run in an epoch, there is at least two sets of colors, S_1 and S_2 chosen such that $c \in S_1$, $c' \notin S_1$ and $c \notin S_2$, $c' \in S_2$ and S_1. There exists runs i_1, i_2 such that the pair of color classes considered is c and c'. Therefore, in this epoch, at least twice the forest between c and c' is considered. In one of those instances, a vertex from color class c is chosen while in the other instance a vertex from color class c' is chosen. Now consider one component in $F(c, c')$- the induced forest between color classes c and c'. Now, we restrict the algorithm to choose one good vertex (among both color classes) using one of those instances in an epoch, from the largest component of $F(c, c')$. Note that, not choosing any vertex at some iterations does not violate the size k constraint. Now, it is possible to simulate the algorithm provided in Lemma 7.4.2, to learn one component in $F(c, c')$ by choosing one vertex in $F(c, c')$ every epoch. Therefore, $\log T$ epochs are enough to learn a component between any two colors classes c and c'. C such runs again is enough to learn all components. All forests between all paris of color classes can be learnt resulting in the entire causal graph being learnt.

The total number of interventions would be at most $C \log T(R * \chi)$ proving the result.

F.11 Proof of Theorem 7.4.5

It is almost identical to the proof of Theorem 7.3.1. However, we provide the argument for the sake of clarity and completeness. By Lemma 7.3.1, ith place has at most $\lceil \frac{n}{\lceil n/k \rceil} \rceil \leq k$ occurrences of symbol j. Therefore, $|S_{i,j}| \leq k$. Now, consider the pair of distinct elements $p, q \in [1 : n]$. Since they are labelled distinctly (Lemma 7.3.1), there is at least one place i in their string labels where they differ. Suppose the distinct ith letters are $a, b \in \mathcal{A},\ a \neq b$. Then, clearly the criterion is met by the subsets $S_{i,a}$ and $S_{i,b}$.

Bibliography

[1] C. D. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999. 8, 97

[2] J.W. Woods. Markov image modeling. *IEEE Transactions on Automatic Control*, 23:846–850, October 1978. 8, 97

[3] M. Hassner and J. Sklansky. Markov random field models of digitized image texture. In *ICPR78*, pages 538–540, 1978. 8, 97

[4] G. Cross and A. Jain. Markov random field texture models. *IEEE Trans. PAMI*, 5:25–39, 1983. 8, 97

[5] E. Ising. Beitrag zur theorie der ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925. 8, 97

[6] B. D. Ripley. *Spatial statistics.* Wiley, New York, 1981. 8, 97

[7] Alexander J Hartemink, David K Gifford, Tommi S Jaakkola, and Richard A Young. Elucidating genetic regulatory networks using graphical models and genomic expression data. 8

[8] http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white/paper/c11-520862.html. 14

[9] A. F. Molisch. *Wireless communications.* IEEE Press - Wiley, 2011. 15

[10] V. Chandrasekhar, J. Andrews, and A. Gatherer. Femtocell networks: a survey. *Communications Magazine, IEEE*, 46(9):59–67, 2008. 15

[11] Michael J Neely. Wireless peer-to-peer scheduling in mobile networks. In *46th Annual Conference on Information Sciences and Systems (CISS), 2012*, pages 1–6. IEEE, 2012. 15

[12] Dilip Bethanabhotla, Giuseppe Caire, and Michael J Neely. Utility optimal scheduling and admission control for adaptive video streaming in small cell networks. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2013*, pages 1944–1948. IEEE, 2013. 15

[13] Chao Chen, Robert W Heath, Alan C Bovik, and Gustavo de Veciana. Adaptive policies for real-time video transmission: a markov decision process framework. In *18th IEEE International Conference on Image Processing (ICIP), 2011*, pages 2249–2252. IEEE, 2011. 15

[14] Vasilios A Siris, Xenofon Vasilakos, and George C Polyzos. A selective neighbor caching approach for supporting mobility in publish/subscribe networks. In *FIFTH ERCIM WORKSHOP ON EMOBILITY*, page 63, 2011. 16

[15] Vasilis Sourlas, Georgios S Paschos, Paris Flegkas, and Leandros Tassiulas. Mobility support through caching in content-based publish/subscribe net-

works. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 2010*, pages 715–720. IEEE, 2010. 16

[16] Abdulbaset Gaddah and Thomas Kunz. Extending mobility to publish/subscribe systems using a pro-active caching approach. *Mobile Information Systems*, 6(4):293–324, 2010. 16

[17] Xenofon Vasilakos, Vasilios A Siris, George C Polyzos, and Marios Pomonis. Proactive selective neighbor caching for enhancing mobility support in information-centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pages 61–66. ACM, 2012. 16

[18] Ivan Baev, Rajmohan Rajaraman, and Chaitanya Swamy. Approximation algorithms for data placement problems. *SIAM Journal on Computing*, 38(4):1411–1429, 2008. 16

[19] Sem Borst, Varun Gupta, and Anwar Walid. Distributed caching algorithms for content distribution networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010. 16

[20] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 126–134. IEEE, 1999. 16

[21] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *Proceedings of the Twenty-Nineth annual ACM symposium on Theory of computing*, pages 654–663. ACM, 1997. 16

[22] M. Rabinovich and O. Spatscheck. Web caching and replication. *SIGMOD Record*, 32(4):107, 2003. 16

[23] Mohammad Ali Maddah-Ali and Urs Niesen. Decentralized caching attains order-optimal memory-rate tradeoff. *arXiv preprint arXiv:1301.5848*, 2013. 16, 74, 76, 77, 81, 89

[24] Mohammad Ali Maddah-Ali and Urs Niesen. Fundamental limits of caching. *arXiv preprint arXiv:1209.5807*, 2012. 16

[25] Mingyue Ji, Giuseppe Caire, and Andreas F. Molisch. Fundamental limits of caching in wireless D2D networks. *CoRR*, abs/1405.5336, 2014. 17

[26] Urs Niesen, Devavrat Shah, and Gregory Wornell. Caching in wireless networks. In *IEEE International Symposium on Information Theory Proceedings (ISIT) 2009.*, pages 2111–2115. IEEE, 2009. 17

[27] N. Golrezaei, A.G. Dimakis, and A.F. Molisch. Scaling behavior for device-to-device communications with distributed caching. *IEEE Transactions on Information Theory*, 60(7):4286–4298, July 2014. 17

[28] S Gitzenis, G Paschos, and L Tassiulas. Asymptotic laws for joint content replication and delivery in wireless networks. 2012. 17

[29] S. El Rouayheb, A. Sprintson, and C. Georghiades. On the index coding problem and its relation to network coding and matroid theory. *IEEE Transactions on Information Theory*, 56(7):3187–3195, 2010. 17

[30] M. Cardei and D.Z. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333–340, 2005. 23

[31] Satoru Fujishige. *Submodular functions and optimization*, volume 58. Elsevier, 2005. 24

[32] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978. 25

[33] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a submodular set function subject to a matroid constraint. *Integer programming and combinatorial optimization*, pages 182–196, 2007. 25

[34] Alexander A Ageev and Maxim I Sviridenko. Pipage rounding: a new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004. 27, 140, 142

[35] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. *Integer programming and combinatorial optimization*, pages 182–196, 2007. 29

[36] Y. Birk and T. Kol. Informed-source coding-on-demand (iscod) over broadcast channels. In *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1257–1264. IEEE, 1998. 31, 32, 34, 35

[37] Z. Bar-Yossef, Y. Birk, TS Jayram, and T. Kol. Index coding with side information. In *47th Annual IEEE Symposium on Foundations of Computer Science, 2006. FOCS'06.*, pages 197–206. IEEE, 2006. 32, 35

[38] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim. Broadcasting with side information. In *49th Annual IEEE Symposium on Foundations of Computer Science, 2008. FOCS'08.*, pages 823–832. IEEE, 2008. 32

[39] A. Blasiak, R. Kleinberg, and E. Lubetzky. Index coding via linear programming. *arXiv preprint arXiv:1004.1379*, 2010. 32, 34, 35, 36, 42

[40] F. Arbabjolfaei, B. Bandemer, Young-Han Kim, E. Sasoglu, and Lele Wang. On the capacity region for index coding. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2013*, pages 962–966, 2013. 32, 33

[41] S. Unal and A.B. Wagner. General index coding with side information: Three decoder case. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2013*, pages 1137–1141, July 2013. 32, 33

[42] Salim El Rouayheb, Alex Sprintson, and Costas Georghiades. On the index coding problem and its relation to network coding and matroid theory. *IEEE Transactions on Information Theory*, 56(7):3187–3195, 2010. 33

[43] M. Effros, S.E. Rouayheb, and M. Langberg. An equivalence between network coding and index coding. *arXiv preprint arXiv:1211.6660*, 2012. 33

[44] H. Maleki, V. Cadambe, and S. Jafar. Index coding: An interference alignment perspective. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2012*, pages 2236–2240. IEEE, 2012. 33, 38

[45] S.A. Jafar. Topological interference management through index coding. *IEEE Transactions on Information Theory*, 60(1):529–568, 2014. 33, 34

[46] Mehrdad Tahmasbi, Amirbehshad Shahrasbi, and Amin Gohari. Critical graphs in index coding. *arXiv preprint arXiv:1312.0132*, 2013. 33

[47] A.S. Tehrani, A.G. Dimakis, and M.J. Neely. Bipartite index coding. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2012*, pages 2246–2250. IEEE, 2012. 34, 35, 43, 44

[48] W Haemers. An upper bound for the shannon capacity of a graph. In *Colloq. Math. Soc. János Bolyai*, volume 25, pages 267–272, 1978. 35

[49] Claude Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8–19, 1956. 35

[50] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. 36

[51] Eyal Lubetzky and Uri Stav. Nonlinear index coding outperforming the linear optimum. *IEEE Transactions on Information Theory*, 55(8):3544–3551, 2009. 36, 59, 71

[52] P. Erdős, Z. Füredi, A. Hajnal, P. Komjáth, V. Rödl, and A. Seress. Coloring graphs with locally few colors. *Discrete mathematics*, 59(1):21–34, 1986. 37

[53] J. Körner, C. Pilotto, and G. Simonyi. Local chromatic number and Sperner capacity. *Journal of Combinatorial Theory, Series B*, 95(1):101–117, 2005. 37, 45, 151, 152, 155, 156

[54] Gábor Simonyi, Gábor Tardos, and Ambrus Zsbán. Relations between the local chromatic number and its directed version. *arXiv preprint arXiv:1305.7473*, 2013. 37, 38, 51

[55] K. Shanmugam, A.G. Dimakis, and M. Langberg. Local graph coloring and index coding. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2013*, pages 1152–1156, 2013. 38, 46

[56] László Lovász. *Large networks and graph limits*, volume 60. AMS Bookstore, 2012. 38

[57] G. Simonyi and G. Tardos. Local chromatic number, Ky Fan's theorem, and circular colorings. *Combinatorica*, 26(5):587–626, 2006. 45

[58] Tracey Ho and Desmond Lun. *Network coding: an introduction*. Cambridge University Press, 2008. 55

[59] Abinesh Ramakrishnan, Abhik Das, Hamed Maleki, Athina Markopoulou, Syed Jafar, and Sriram Vishwanath. Network coding for three unicast sessions: Interference alignment approaches. In *48th Annual Allerton Con-*

ference on Communication, Control, and Computing (Allerton), 2010, pages 1054–1061. IEEE, 2010. 55

[60] Chun Meng, Abinesh Ramakrishnan, Athina Markopoulou, and Syed Ali Jafar. On the feasibility of precoding-based network alignment for three unicast sessions. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2012*, pages 1907–1911. IEEE, 2012. 55

[61] Randall Dougherty, Chris Freiling, and Kenneth Zeger. Networks, matroids, and non-shannon information inequalities. *IEEE Transactions on Information Theory*, 53(6):1949–1969, 2007. 55

[62] Michael Langberg and Alex Sprintson. On the hardness of approximating the network coding capacity. *IEEE Transactions on Information Theory,*, 57(2):1008–1014, 2011. 55

[63] Shachar Lovett. Linear codes cannot approximate the network capacity within any constant factor. *ECCC TR14-141*, 2014. 55, 59

[64] LR Ford and Delbert Ray Fulkerson. *Flows in networks*, volume 1962. Princeton Princeton University Press, 1962. 55

[65] Michael Saks, Alex Samorodnitsky, and Leonid Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24(3):525–530, 2004. 56

[66] Sudeep Kamath. *A study of some problems in network information theory.* PhD thesis, EECS Department, University of California, Berkeley, Aug 2013. 56, 67, 68, 72

[67] Gerhard Kramer and Serap A Savari. Edge-cut bounds on network coding rates. *Journal of Network and Systems Management*, 14(1):49–67, 2006. 56

[68] Nicholas JA Harvey, R Kleinberg, and AR Lehman. On the capacity of information networks. *IEEE Transactions on Information Theory*, 52(6):2345–2364, 2006. 56

[69] Satyajit Thakor, Alex Grant, and Terence Chan. Network coding capacity: A functional dependence bound. In *IEEE International Symposium on Information Theory (ISIT), 2009.*, pages 263–267. IEEE, 2009. 56

[70] Frederique Oggier and Anwitaman Datta. Self-repairing homomorphic codes for distributed storage systems. In *INFOCOM, 2011 Proceedings IEEE*, pages 1215–1223. IEEE, 2011. 57

[71] Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin. On the locality of codeword symbols. 2012. 57

[72] N Prakash, Govinda M Kamath, V Lalitha, and P Vijay Kumar. Optimal linear codes with a local-error-correction property. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2012*, pages 2776–2780. IEEE, 2012. 57

[73] Dimitris S Papailiopoulos, Jianqiang Luo, Alexandros G Dimakis, Cheng Huang, and Jin Li. Simple regenerating codes: Network coding for cloud storage. In *INFOCOM, 2012 Proceedings IEEE*, pages 2801–2805. IEEE, 2012. 57

[74] Dimitris S Papailiopoulos and Alexandros G Dimakis. Locally repairable codes. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2012*, pages 2771–2775. IEEE, 2012. 57

[75] Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, Sergey Yekhanin, et al. Erasure coding in windows azure storage. *USENIX ATC*, 2012. 57

[76] Arya Mazumdar. On a duality between recoverable distributed storage and index coding. *arXiv preprint arXiv:1401.2672.pdf*, 2014. 57

[77] Anna Blasiak. *A Graph-Theoretic Approach To Network Coding*. PhD thesis, Cornell University, 2013. 59, 71

[78] Karthikeyan Shanmugam, Megasthenis Asteris, and Alexandros G Dimakis. On approximating the sum-rate for multiple-unicasts. *arXiv preprint:1504.05294*, 2015. 59, 66, 70, 71

[79] Karthikeyan Shanmugam, Megasthenis Asteris, and Alexandros G Dimakis. On approximating the sum-rate for multiple-unicasts. In *IEEE International Symposium on Information Theory (ISIT), 2015*, pages 381–385. IEEE, 2015. 66, 71

[80] Sudeep U Kamath, David NC Tse, and Venkat Anantharam. Generalized network sharing outer bound and the two-unicast problem. In *International Symposium on Network Coding (NetCod), 2011*, pages 1–6. IEEE, 2011. 68

[81] Paul D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995. 69

[82] Guy Even, J Seffi Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998. 69, 171

[83] Zeev Nutov and Raphael Yuster. Packing directed cycles efficiently. In *Mathematical Foundations of Computer Science 2004*, pages 310–321. Springer, 2004. 69

[84] Mohammad Ali Maddah-Ali and Urs Niesen. Fundamental limits of caching. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2013*, pages 1077–1081. IEEE, 2013. 74, 76, 90, 91

[85] Urs Niesen and Mohammad Ali Maddah-Ali. Coded caching with nonuniform demands. *arXiv preprint arXiv:1308.0178*, 2013. 77

[86] Mingyue Ji, Antonia M Tulino, Jaime Llorca, and Giuseppe Caire. On the average performance of caching and coded multicasting with random demands. *arXiv preprint arXiv:1402.4576*, 2014. 77

[87] Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *Proceed-*

ings of the 11th international workshop, APPROX 2008, and 12th international workshop, RANDOM 2008 on Approximation, Randomization and Combinatorial Optimization: Algorithms and Techniques, APPROX '08 / RANDOM '08, pages 343–356. Springer-Verlag, 2008. 99

[88] Rashish Tandon and Pradeep Ravikumar. On the difficulty of learning power law graphical models. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2013*, pages 2493–2497. IEEE, 2013. 99

[89] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012. 99, 105, 106, 107, 108

[90] Animashree Anandkumar, Vincent YF Tan, Furong Huang, Alan S Willsky, et al. High-dimensional structure estimation in ising models: Local separation criterion. *The Annals of Statistics*, 40(3):1346–1375, 2012. 100, 104, 105, 109, 185

[91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. 103

[92] Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599, 1999. 105, 186

[93] Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Trans. Inf. Theor.*, 57(10):6976–6994, October 2011. 105

[94] Yuchen Zhang, John Duchi, Michael Jordan, and Martin J Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Advances in Neural Information Processing Systems 26*, pages 2328–2336. Curran Associates, Inc., 2013. 105

[95] Karthikeyan Shanmugam, Rashish Tandon, Alexandros G Dimakis, and Pradeep Ravikumar. On the information theoretic limits of learning ising models. *arXiv preprint arXiv:1411.1434*, 2014. 110

[96] Rashish Tandon, Karthikeyan Shanmugam, Pradeep K Ravikumar, and Alexandros G Dimakis. On the information theoretic limits of learning ising models. In *Advances in Neural Information Processing Systems*, pages 2303–2311, 2014. 110

[97] Judea Pearl. *Causality: Models, Reasoning and Inference.* Cambridge University Press, 2009. 112, 113, 115, 117

[98] Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55(4):926–939, 2014. 112, 113, 196, 197

[99] Frederich Eberhardt, Clark Glymour, and Richard Scheines. On the number of experiments sufficient and in the worst case necessary to identify all

causal relations among n variables. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 178–184. 112, 113

[100] Antti Hyttinen, Frederick Eberhardt, and Patrik Hoyer. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14:3041–3071, 2013. 112, 113

[101] Huining Hu, Zhentao Li, and Adrian Vetta. Randomized experimental design for causal graph discovery. In *Proceedings of NIPS 2014*, Montreal, CA, December 2014. 112, 114, 123, 196

[102] S Shimizu, P. O Hoyer, A Hyvarinen, and A. J Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006. 112

[103] Patrik O Hoyer, Dominik Janzing, Joris Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Proceedings of NIPS 2008*, 2008. 112

[104] Frederick Eberhardt. *Causation and Intervention (Ph.D. Thesis)*, 2007. 113, 114

[105] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. A Bradford Book, 2001. 116, 117

[106] Christopher Meek. Strong completeness and faithfulness in bayesian networks. In *Proceedings of the eleventh international conference on Uncertainty in Artificial Intelligence (UAI)*, 1995. 116

[107] Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997. 116

[108] Thomas Verma and Judea Pearl. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Proceedings of the Eighth international conference on Uncertainty in Artificial Intelligence (UAI)*, pages 323–330. Morgan Kaufmann Publishers Inc., 1992. 117

[109] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the eleventh international conference on Uncertainty in Artificial Intelligence (UAI)*, 1995. 117

[110] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(1):2409–2464, 2012. 118

[111] Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal networks from interventional data. In *Proceedings of Sixth European Workshop on Probabilistic Graphical Models*, 2012. 119

[112] Gyula Katona. On separating systems of a finite set. *Journal of Combinatorial Theory*, 1(2):174–194, 1966. 121, 122

[113] Ingo Wegener. On separating systems whose elements are sets of at most k elements. *Discrete Mathematics*, 28(2):219–222, 1979. 121, 122

[114] Cai Mao-Cheng. On a problem of katona on minimal completely separating systems with restrictions. *Discrete mathematics*, 48(1):121–123, 1984. 127

[115] G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988. 134

[116] C.D. Godsil, G. Royle, and CD Godsil. *Algebraic graph theory*, volume 8. Springer New York, 2001. 153

[117] Mohammad Asad R Chaudhry, Zakia Asad, Alex Sprintson, and Michael Langberg. On the complementary index coding problem. In *IEEE International Symposium on Information Theory Proceedings (ISIT), 2011*, pages 244–248. IEEE, 2011. 171

[118] Martin Raab and Angelika Steger. balls into binsa simple and tight analysis. In *Randomization and Approximation Techniques in Computer Science*, pages 159–170. Springer, 1998. 179

[119] Stasys Jukna. *Extremal combinatorics*, volume 2. Springer, 2001. 180

[120] Amir Dembo and Andrea Montanari. Ising models on locally tree-like graphs. *The Annals of Applied Probability*, 20(2):565–592, 04 2010. 187

[121] Richard J Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. 194

Vita

Karthikeyan Shanmugam obtained his B.Tech and M.Tech degrees in Electrical Engineering from IIT Madras in 2010 and an M.S degree in Electrical Engineering from the University of Southern California in 2012 . He was a summer research intern at Alcatel-Lucent Bell Labs at Crawford Hill, NJ during the summer of 2014. He is currently pursuing his Ph.D under the supervision of Prof. Alexandros G. Dimakis. His research interest are broadly in Machine learning, Graph algorithms, Coding Theory and Information Theory.

Permanent address: karthiksh@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.