A GRAPH-THEORETIC APPROACH TO NETWORK CODING

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Anna Blasiak

August 2013

A GRAPH-THEORETIC APPROACH TO NETWORK CODING

Anna Blasiak, Ph.D.
Cornell University 2013

The network coding problem is a generalization of the maximum flow problem in which nodes can, in addition to forwarding messages, send encodings of combinations of incoming packets. This problem addresses the transmission of information, rather than physical goods, as information can be scrambled and unscrambled in ways that have no physical analogue. Network coding has been extremely successful in the setting of multicast. In this setting, network coding is an improvement over flow both because coding can send information at a higher rate and also because it can be computed efficiently.

Much less is known about network coding in other settings, and this gap in knowledge is the focus of our work. Most significantly, we consider a problem called broadcasting with side information problem (BSIP): a problem that considers network coding on a restricted network structure but with arbitrary sending and receiving requests. The network structure is so simple that the problem is expressed only in terms of its senders and receivers. To go into more detail, the BSIP begins with a sender and sets of receivers and messages. Each receiver possesses a subset of the messages and desires an additional message from the set. The sender wishes to broadcast a message so that on receipt of the broadcast each user can compute her desired message. The objective is to find a minimum length broadcast that accomplishes this goal. The fundamental parameter of interest is β, the average broadcast length for sufficiently long source messages.

We obtain improved bounds on β by strengthening and extending previously known bounds. Additionally, we introduce a new class of bounds based on an information-theoretic linear program. We show that many of these bounds behave nicely under various product and sum operations. Most notably, β is sub-multiplicative, and the linear programming bounds are super-multiplicative under the same product operation.

We use these new bounds and our understanding of them under products to obtain a multitude of results. We are the first to pinpoint β precisely in nontrivial instances. We do this for many classes of symmetric instances including cycles and those derived from representable matroids.

We find polynomial gaps between β and its bounds in cases in which the largest previously known gaps were small constant factors or entirely unknown. We show a polynomial gap between β and the linear coding rate and also between β and its trivial lower bound. We construct a family of instances where β is constant while its upper bound derived from the naïve encoding scheme grows polynomially in the instance size. Finally, we give the first nontrivial approximation algorithm for computing β and we give a polynomial-time algorithm for recognizing instances with $\beta = 2$.

Apart from the BSIP, we consider the network coding variant of the maximum multicommodity flow problem in directed networks. We identify a class of networks on which the coding rate is equal to the size of the minimum multicut and show this class is closed under the strong graph product. We apply our result to strengthen the multicut bound for a famous construction of Saks *et al.*. We determine the exact value of the minimum multicut for their construction and give an optimal network coding solution with a matching rate.

BIOGRAPHICAL SKETCH

Born to James and Miriam Blasiak, Anna began life in the greater Pittsburgh area and grew up in the D.C. suburbs. Her education, though somewhat bogged down by the requisite studying, was filled with a preponderance of extracurricular activities: competitive swimming, orchestra oboe, oil painting, just to name a few. Academically, her interests from the start leaned towards the hard sciences, but a grueling two years of rewardless labwork at the end of high school pushed her toward her true passion, even if she didn't know it yet.

Luckily for Middlebury College, Anna rebuffed her obvious Ivy League upbringing, and turned to a more intimate liberal arts setting to study mathematics. Luckily for Computer Science, that same intimate setting introduced her to professor Daniel Scharstein, who taught her that CS wasn't just programming, it was the study of doing things efficiently. Not wanting to waste any time, she picked up a computer science major immediately. During a math-centric semester in Budapest, her zeal for graph theory and future course of research crystallized.

Anna graduated from Middlebury in 2007 and left the cozy confines of Vermont for the precipices of Ithaca to pursue a Ph.D. in Computer Science. At Cornell, she took up an interest in ultimate frisbee and network coding, the latter of which she studied under the supervision of Bobby Kleinberg, and the former she tried to hide from him. Anna is a recipient of the AT&T Labs fellowship and the NSF and NDSEG graduate fellowships. She plans to defend her thesis in July 2013 and begin work at Akamai Technologies.

ACKNOWLEDGEMENTS

There are so many people who have helped me along the way to my Ph.D. that I cannot possibly credit them all here. I offer an overarching ”thank you” to all my friends, family, professors, and colleagues. However, I would be remiss not to call out a special few.

None of this work would have been possible without my collaborators, Bobby Kleinberg and Eyal Lubetzky. It has been a pleasure to work with and learn from both of them. They have endless knowledge on a multitude of topics and are never without insight into tackling any problem. Bobby Kleinberg has additionally served as an amazing adviser and role model. He's patiently worked with me to become a better researcher. He has taught me to think and write more formally, speak more clearly, and ask the right questions. Bobby's excitement for new research ideas and results has been an inspiration and motivation.

Next, Aaron Archer, a constant source of advice and my mentor and collaborator on work absent here. Aaron has not only been there to help me decide which internships to apply to but also to inform me of all the best places to travel, the best music to listen to, the best of everything.

Then, the people here since day one, my parents, constantly supporting me and providing every opportunity and advantage. They have talked me through countless crises of confidence and indecision. Their unconditional love has given me the security and strength to brave this tough journey and every other.

My brothers, Jonah and Sam, inspire me to be a scientist, and continually assure me it's the best, nay, only life path. Jonah's ability to listen to my research and sort out my thoughts can only be attributed to divine patience.

Last, and accordingly the least, is Jesse Simons, my love and my rock here at Cornell. He has helped me through every tear - ones he caused, ones he didn't, and ones where we couldn't tell the difference. He is constantly driving me to be the best person I can be and to take life a little less seriously. He has been an amazing editor, and it's likely that every well-written paragraph and subtle joke in this dissertation is due to him. For that, he definitely deserves a high five.

Aside from people, I owe gratitude to the organizations that financially supported my Ph.D.. Thanks, AT&T Labs Research fellowship, the National Science Foundation (NSF) graduate fellowship, and National Defense and Science Engineering Graduate (NDSEG) fellowship.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

BSIP instance parameters

G	a BSIP or BSIP-G instance	Definition 1.2.1
V	set of vertices that index the message set	Definition 1.2.1
E	set of edges that index the receiver set	Definition 1.2.1
$f(j)$	the index of the message wanted by receiver j	Definition 1.2.1
$N(j)$	the indices of the set of messages known to receiver j	Definition 1.2.1
$S(j)$	$N(j) \cup \{f(j)\}$	Definition 1.2.1
$T(j)$	$V \setminus S(j)$	Definition 1.2.1
$\rightsquigarrow$	decodes	Definition 2.3.1
cl	closure	Definition 2.3.1
Σ	message alphabet	Definition 1.2.2
Σ_P	broadcast alphabet	Definition 1.2.2

Broadcast Rates

β	The broadcast rate	Definition1.2.3
β_k	The broadcast rate over alphabet $\{0,1\}^k$	Definition 1.2.2
β^*	The limiting broadcast rate of disjoint unions of G over alphabet $\{0,1\}$	Equation (1.8)
$\lambda^{\mathbb{F}}$	The vector linear broadcast rate over finite field $\mathbb{F}$	Definition 1.2.6
λ	The vector linear broadcast rate	Definition 1.2.6
$\lambda_1^{\mathbb{F}}$	The scalar linear broadcast rate over finite field $\mathbb{F}$	Definition 1.2.5
λ_1	The scalar linear broadcast rate	Definition 1.2.5

Upper bounds on the broadcast rate

$\overline{\chi}$	(hyper)clique-cover number	-
$\overline{\chi}_f$	Fractional (hyper)clique-cover number	Definition 2.1.6
minrk	Minrank	Definition 1.2.7
$\text{minrk}^{\mathbb{F}}$	Minrank over finite field $\mathbb{F}$	Definition 1.2.7
minrk_f	Fractional minrank	Definition 2.2.1
$\text{minrk}_f^{\mathbb{F}}$	Fractional minrank over finite field $\mathbb{F}$	Definition 2.2.1

Lower bounds on the broadcast rate

α	Minimum expanding sequence (independent set)	Definition 2.1.1
b	The Shannon bound	Definition 2.3.9
b_{ZY}	The Zhang-Yeung Lower Bound	Definition 2.3.13
$b_{\mathcal{F}}$	The Fano bound	Definition 2.3.17
$b_{\mathcal{N}}$	The non-Fano bound	Definition 2.3.18

Linear Programs

$\mathfrak{B}$	The shannon bound LP	Definition 2.3.9
$\mathfrak{B}_{ZY}$	The Zhang-Yeung Lower Bound LP	Definition 2.3.13
$\mathfrak{B}_{\mathcal{F}}$	The Fano bound LP	Definition 2.3.17
$\mathfrak{B}_{\mathcal{N}}$	The non-Fano bound LP	Definition 2.3.18

Subsets of $\mathbb{R}^{2^n}$

Γ_n	The set of vectors satisfying Shannon inequalities	Definition 3.1.1
Γ_n^*	The set of entropic vectors	Definition 3.1.1
$\overline{\Gamma}_n^*$	The closure of the set of entropic vectors on n random variables	Definition 3.1.1
Υ_n	The set of dimension vectors on n subspaces	Definition 3.1.1
$\Upsilon_n^{\mathbb{F}}$	The set of dimension vectors on n subspaces in a vector space over $\mathbb{F}$	Definition 3.1.1

Products and Sums

$\bullet$	lexicographic product of BSIP instances	Definition 4.1.1
$+$	disjoint union of BSIP instances	Section 4.3
$\boxtimes$	strong product of BSIP instances	Definition 4.2.2
$\otimes$	The kronecker product of two matrices	-

Matroids

$M = (E, r)$	matroid with groundset E and rank function r	Definition 3.1.1
G_M	BSIP instance associated to matroid M	Definition 3.1.2
$\vec{\mathbf{r}}$	matroid rank vector	Definition 3.1.1
cl	closure	Equation (3.1)
$\mathcal{F}$	fano matroid	Definition 5.2.2
$\mathcal{N}$	non-fano matroid	Definition 5.2.2

Other

$\mathcal{P}(I)$	the powerset of an index set I	-
$\mathsf{rank}(A)$	the rank of a matrix A	-
$[n]$	the subset $\{1, 2, 3, \ldots n\}$	-
$\mathbb{F}$	finite field	-
$GF(2)$	Galois field of order 2	-
$\mathbb{F}_p$	finite field of order p for p prime	-
char($\mathbb{F}$)	The characteristic of the field $\mathbb{F}$	-

CHAPTER 1

INTRODUCTION

The problem of *network coding* was first considered by Ahlswede, Cai, Li, and Yeung [4] in their paper "Network Information Flow" in the setting of *multicast*: given one source and many sinks, how many messages can be sent to all of the sinks simultaneously? If there is only one sink then fundamental theorems of *network flow* dictate that the amount of information that can be sent from the source to the sink is equal to the capacity of the minimum cut separating the source and sink. Moreover, it is possible to compute an optimal solution efficiently. But the addition of more sinks changes the problem significantly. From the network flow perspective, the answer to this question amounts to computing a fractional packing of Steiner trees as opposed to the fractional packing of paths needed for only one sink. Not only is the Steiner tree packing problem NP-hard [42], but the optimal solution is also far from the cut upper bound, the minimum of all minimum source-sink cuts. It can be a multiplicative factor of $\Omega\left((\log n/\log\log n)^2\right)$ smaller than the cut in directed graphs and at least $36/31 > 1.16$ and at most 1.55 in undirected graphs [15, 3].

Ahlswede, Cai, Li, Yeung [4] consider the multicast problem from a new perspective. They generalize the classic flow perspective in which nodes can only forward messages to the setting of *network coding* in which nodes can send encodings of combinations of incoming packets. This problem differs from classical routing by specifically addressing the rate of transmission of information: information, unlike physical goods, can be scrambled and unscrambled in ways that have no physical analogue. A simple, commonly-used code is to have a node send the XOR of all the packets it receives.

Ahlswede *et al.* [4] prove that in the multicast setting the new perspective of network coding can yield huge throughput gains. In particular, they show that the optimal coding rate is equal to the cut upper bound. Li *et al.* [50] show that this optimality doesn't depend on any complicated codes, in particular, the optimal rate can be achieved using only linear codes. Moreover, in contrast to the NP-hard packing problem needed to solve the multicast problem in the classical setting, Jaggi *et al.* [41] show that an optimal linear coding solution can be found in polynomial time. Even more impressively, with high probability, the solution in which each node outputs a random linear combination of its input is optimal [39]. These results have successfully carried over to practical applications. There are many examples where network coding provides faster transmission rates compared to traditional routing, e.g. [44] details a recent success in wireless networks.

Such successes began the study of network coding and motivated questions regarding coding in other settings. For examples, given a network flow optimization problem, how much benefit does coding provide over classical flow? What type of codes suffice to give the maximum rate? Is there an efficient way to compute the coding rate?

Much research has been dedicated to answering all three of these questions for various optimization problems. Before recounting previous work, we provide a more formal description of the problem. The definitions are modified from [48].

Definition 1.0.1. An instance of the *general network coding problem* is specified by a graph $G = (V, E)$, a non-negative integer capacity $c(e)$ for each edge e, a set I consisting of k commodities, and for each commodity $i \in I$, a set of sources $\mathsf{Src}(i) \subseteq V$ and a set of sinks $\mathsf{Snk}(i) \subseteq V$.

In the multicast problem $|I| = |\mathsf{Src}(i)| = 1$.

To define a coding solution we need to have a notion of what it means for a sink to receive all of the requested information and for each node to be able to send all the requested information.

Definition 1.0.2. Given functions $f_0, f_1, \ldots, f_k$ on the same domain Σ, we say that

$f_1, f_2, \ldots, f_k$ *determines* f_0, or f_0 is *computable* from $f_1, \ldots, f_k$ if for all $x, x' \in \Sigma$ such that $f_i(x) = f_i(x')$ for all $i = 1, \ldots, k$, we have that $f_0(x) = f_0(x')$.

Definition 1.0.3. A *network coding solution* specifies a source alphabets Σ_i, edge alphabets Σ_e for each edge $e \in E$, a function $f_e : \prod_{i=1}^{k} \Sigma_i \mapsto \Sigma_e$ such that for every k-tuple of messages $x = (x_1, x_2, \ldots, x_k) \in \Sigma^k$:

1. For every edge $(u, v) \in E$, the function $f(u, v)$ is computable from the functions on in-edges to u and messages for which u is a source. [1]
2. For every sink v for commodity i, the functions on in-edges to v together with the messages for which v is a source are sufficient to determine the value of x_i.

The *coding rate* is the supremum of $\log_b(\min_i(|\Sigma_i|))$ over b such that $\log_b |\Sigma_e| \leq c(e)$ for all $e \in E$. It captures the amount of information received at each sink when we insist that $|\Sigma_i| = |\Sigma_j|$ for all commodities i, j and scale down the message alphabet to obey capacity constraints.

A variation of the general network coding problem that is intensively studied is called *multiple unicast*, the coding analogue of the *concurrent multicommodity flow* problem, a fundamental problem in network flow theory. This special case is obtained by requiring $|\mathsf{Src}(i)| = |\mathsf{Snk}(i)| = 1$ in Definition 1.0.1.

We now give a short overview of the previous work addressing our three guiding questions in relation to the general network coding problem and the multiple unicast problem.

[1] In graphs with cycles this is not a sufficient characterization. One sufficient, but not necessary, characterization is to additionally define an ordering on the edges and require that each function $f(u, v)$ is computable form the functions on in-edges to u preceding (u, v) in the ordering. See [11, 43, 36, 38, 48] for more discussion on how to define coding for graphs with cycles.

1.1 Previous Work: Network Coding

1.1.1 How much benefit does coding provide over classical flow?

The utility of network coding for the directed and undirected versions of the multiple unicast problem differs drastically. In the latter, the network coding rate is sandwiched between the multicommodity flow rate and the *sparsest cut*, the integral solution to the dual of the multicommodity flow linear program. It is known that the worst-case gap between these two parameters, the *flow-cut gap*, is $\Theta(\min(\log n, \log k))$ where n is the number of nodes and k is the number of source-sink pairs. Further, on large classes of graphs the flow and cut values are known to coincide. For example, they coincide for $k = 2$ and certain planar graphs. Thus, the benefit of coding over flow is known to be limited.

But it may be even more limited; no example is known for which the coding rate is larger than the flow rate. In fact, Li and Li [51] predict that the flow rate and coding rate coincide in what they named the *undirected k-pairs conjecture.* Jain *et al.* [43] and Adler *et al.* [1] provide additional evidence for the conjecture, but it remains one of the most important open problems in network coding. The importance stems, in part, from the conjecture's complexity-theoretic implications: for example, if true, it implies an affirmative answer to a 25-year-old conjecture regarding the I/O complexity of matrix transposition [1].

The multiple unicast problem in directed graphs has a much different story. There are simple examples in which the coding rate achieves a rate that is a factor k and a factor $\Omega(n)$ larger than the flow rate [51, 35]. Moreover, the flow-cut gap does not pose a limit on the coding-flow gap. Here, and always, the coding rate is at least the flow rate, as coding is a generalization of the flow problem. But, the sparsest cut is not an upper bound on the coding rate, and in a strong sense: the coding rate can be a factor k larger than the sparsest cut [4]. Even in instances when the coding rate is upper bounded by the cut, the flow-cut gap is still not a limitation to huge throughput gains: the cut upper bound can be an $\Omega(k)$ [60] and $\tilde{\Omega}(n^{1/7})$ [20] factor larger than the best flow. It is unknown how much smaller the coding rate can be than the cut, and this is something we investigate in Chapter 7.

1.1.2 What type of codes suffice to give the maximum rate?

Linear codes can achieve the optimal solution for multicast, and the hope is that this holds true for other demand structures, at least in an approximate sense, since linear solutions are more practical and better understood. The literature classifies two types of linear codes: scalar linear codes, in which messages are required to be elements of a finite field, and vector linear codes, in which the messages are finite-dimensional vectors. Scalar linear codes suffice for multicast, but they're known to be insufficient for multiple unicast: non-linear coding can be a polynomial factor better than scalar linear [54]. At first, the more powerful vector linear coding was falsely conjectured to be sufficient [58]; Dougherty *et al.* [24] construct an example with a non-linear code that exceeds the vector linear code by a factor of 11/10. Yet this example leaves the possibility that vector linear codes could be a good approximation to the optimal code. We resolve this question in the negative in Section 5.2 and show an $\Omega\left(n^{\frac{1}{2}-\varepsilon}\right)$ multiplicative gap between vector linear and non-linear coding.

1.1.3 Is there an efficient way to compute the coding rate?

Though there has been a significant amount of work addressing this problem almost no progress has been made. It is unknown whether the multiple unicast problem is recursively decidable. The problem's potential undecidability stems from the fact that the functions used for encoding messages on the edges of the network may depend on an arbitrary number of bits of the data streams, giving rise to an infinite search space of solutions. Further, we have developed no non-trivial approximation algorithm.

Failures in finding algorithms have come hand-in-hand with failures in finding hardness results. There are no hardness results for the most general version of the problem with arbitrary coding functions and alphabets; that is, we have not even excluded linear time solvability. But there are a number of hardness results for restricted coding functions and fixed alphabet sizes. Lehman and Lehman [49] show that computing the scalar linear capacity of the multiple unicast problem is NP-hard via a reduction from 3-SAT. Langberg and Sprintson [47] show that for a fixed alphabet finding a constant approximation to the general network coding problem is hard assuming the unique games conjecture.

There has been a significant body of work devoted to finding upper bounds on the general network coding problem in directed graphs. Finding a good upper bound on the optimal solution value is a first step to most approximation algorithms. The upper bound gives a value to which you can compare the solution value of the algorithm. It is important that the maximum gap between the upper bound and the optimal solution is small because this gap is a limit on the approximation ratio one can prove using this upper bound.

Most work on determining an upper bound takes the following approach. Regard each edge of the network as defining a random variable on a probability space and associate to each set of edges the Shannon entropy of the joint distribution of their random variables. This gives us a vector of non-negative numbers, one for each edge set, called the *entropic vector* of the network code. The closure of the set of entropic vectors of network codes forms a convex set, and network coding problems can be expressed as optimization problems over this set [64]. This set is characterized by two types of constraints.

The first constraint type is derived from the combinatorial structure of the network. Yeung and Zhang [65] characterize this type of constraint for directed acyclic graphs: it captures requirements (1) and (2) in the definition of a network coding solution (Definition 1.0.1) by imposing a constraint for each node enforcing that the entropy of the node's incoming edge set equals the entropy of all of its incoming and outgoing edges. In cyclic graphs this is much more challenging and various constraints have been discovered over the years [43, 36, 38, 46] but there is not yet a complete classification.

The second type of constraint is purely information-theoretic. These constraints are referred to as *entropy inequalities* or *information inequalities*, and they hold universally for all n-tuples of random variables, regardless of their interpretation as coding functions on edges of a network. Just as we can consider the entropic vector of a network code, we can consider the entropic vector of an arbitrary set of n random variables. The set of all such vectors in 2^n dimensional space is denoted by Γ_n^*. Its closure is denoted $\overline{\Gamma}_n^*$ and characterizing it is equivalent to finding all possible information inequalities [64]. A related set of interest Γ, a superset of Γ^*, corresponds to a region bounded by the *Shannon-type inequalities*, the set of inequalities that can be derived from the non-negativity of conditional mutual information: $H(AC) + H(BC) - H(ABC) - H(C) = I(A; B|C) \geq 0$. Alternatively, they can be written as the combination of the polymatroidal axioms: monotonicity ($H(A) \leq H(AB)$), non-negativity ($H(A) \geq 0$), and submodularity ($H(A) + H(B) \geq H(AB) + H(A \cap B)$). For $n \leq 3$ it is known $\Gamma_n = \overline{\Gamma}_n^*$ [66], but, for $n > 3$, $\Gamma_n \supset \Gamma_n^*$ [67], so additional inequalities are needed to describe the set of entropic vectors in general. Numerous papers ([57, 67, 25, 16, 55] to name just a few) are devoted to deriving such inequalities. These so-called *non-Shannon Inequalities* are not implied by Shannon inequalities and are valid for all entropic vectors. But coming up with a complete characterization of the region has been elusive. Even for $n = 4$ it is known that there are infinitely many such inequalities [57].

Just as Shannon inequalities are insufficient to characterize entropic vectors, they are also insufficient in characterizing the optimal network coding solution. Dougherty *et al.* [22] show a network coding instance in which the combinatorial inequalities described above together with the Shannon inequalities do not give a tight bound on the coding rate. Furthermore, obtaining a complete characterization of the capacity of network coding will imply a characterization of $\overline{\Gamma}^*$ [17].

Rather than compute the capacity region, researchers interesting in bounding the network coding rate more commonly attempt to extend the cut perspective of the flow problem. Though the sparsest cut, an upper bound on the flow rate in the multicommodity flow problem, isn't an upper bound on the coding rate in directed graphs, entropy inequalities show that the capacity of a cut that disconnects all sinks from all sources is an upper bound on the network coding rate. There is work devoted to expanding that idea with more complicated information-theoretic arguments [11, 35, 37, 46, 61]. However, almost all of these bounds are known to be to be bad; each can be a factor n larger than the coding rate. The bound *iMeagerness* is introduced in [37] and they show it can be logarithmically larger than the coding rate. In Section 5.3 we show a polynomial separation.

Considering all of this previous work, one thing is clear: outside the multicast setting, network coding is hard. The difficulty of the general problem has motivated interest in *broadcasting with side information* or *index coding*, a special case of the network coding problem introduced by Birk and Kol [9] with the most general demand structure but a restricted network structure. It is interesting as a problem on its own,for its implications to network coding, and also for its nice connections to graph theory.

1.2 Broadcasting with Side Information

An instance of the broadcasting with side information problem (BSIP) consists of a sender and sets of users and messages. Each user possesses a subset of the messages and desires an additional message from the set. The sender wishes to broadcast a message over a noiseless channel so that on receipt of the broadcast each user can compute her desired message. The objective is to find a minimum length broadcast that accomplishes this goal.

1.2.1 Applications

One motivating application for the problem is satellite transmission of large files (e.g. video on demand), where a slow uplink may be used to inform the server of the side-information map, namely the identities of the files currently stored at each client due to past transmissions. The goal of the server is then to issue the shortest possible broadcast that allows every client to decode its target file while minimizing the overall latency. Another application is to optimize the recovery phase after a multicast transmission. After sending an IP multicast, acknowledgments are sent back to the server confirming which packets were received. The server now has an index coding problem at hand - each receiver has some set of packets they initially wanted but got lost, and a set of packets they received. The server needs to determine a short message to multicast so that all receivers get their required packets.

The BSIP is a special case of the general network coding problem. Given a BSIP instance, the corresponding network coding problem is given by a graph with one node, u_i for each message i, one node v_j for each receiver j, and two additional nodes w, w'. There is one edge of finite capacity, that we call the *bottleneck edge* going from w to w'. There is a infinite capacity edge from u_i to w for all messages i and from w' to v_j for all receivers j. Additionally, there is an infinite capacity edge between u_i and v_j if receiver j has side information containing source message i. There is a commodity for each message i , with source u_i and sink v_j for each receiver j desires message i. See Figure 1.1 for an illustration of the reduction. The only interesting part of any solution to this network coding instance is deciding what to send over the bottleneck edge: finding the optimal network coding rate is equivalent to finding the minimum length message to send over this edge. The requirements of the coding function on this edge are exactly the requirements of the broadcast message

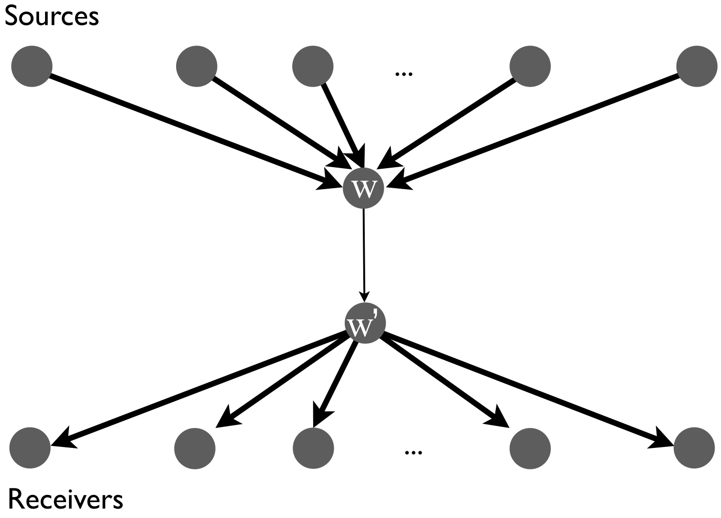


Figure 1.1: BSIP instance as a network coding problem.

The edge (w, w') is the bottleneck edge and the only finite capacity edge in the network. In addition to the edges shown, there is an edge directly from source node i to receiver node j if the receiver j has source i as side information.

in the corresponding BSIP instance.

BSIP seems to capture the instances in which network coding is the most powerful relative to flow. Any BSIP instance with k messages corresponds, according to our mapping described above, to a network coding problem in which the maximum flow has value $1/k$ for each receiver (assuming the bottleneck edge has capacity one), as the only path connecting all source-receiver pairs includes the bottleneck edge. The best network coding solution can use the bottleneck edge much more efficiently. For example, if all receivers know all the messages but the one they desire, sending the XOR of all messages along the bottleneck gives coding rate 1 for each receiver. The receiver can obtain its desired message by subtracting from the XOR all the messages it obtains from the other sources over infinite capacity links (the side information in the BSIP).

More surprisingly, BSIP actually captures the difficulty of all network coding problems. Recently, Effros, Rouayheb, and Langberg [26] show that every network coding instance can be reduced to an equivalent BSIP instance. Though this reduction is not approximation preserving, it does give a manageable way to solve the general network coding problem by solving the ostensibly simpler BSIP.

BSIP is related to other coding problems as well. The *topological interference alignment problem* is a coding problem that consists of wireless transmitters which each hold a message, and a set of receivers who can hear certain transmissions. The goal is to find a protocol so that each receiver can distinguish their desired message from the interference. Maleki *et al.* [56] show that this problem is equivalent to a certain BSIP instance on the same set of messages and receivers.

1.2.2 The Formal Problem Definition

Before recounting previous work, we establish a formal definition of BSIP as well as some related problems of interest.

Definition 1.2.1. A *BSIP instance* is given by a directed hypergraph $G = (V, E)$ where $V = [n]$ is the set of vertices, and $E = [m]$ is the set of directed hyperedges. Each vertex i corresponds to a message $x_i \in \Sigma$, where $|\Sigma| > 1$. Each hyperedge j specifies the values $f(j)$ and $N(j)$ and corresponds to a receiver R_j that is interested in one message, $x_{f(j)}$, and knows some subset, $\{x_i | i \in N(j)\}$, of the other messages. We will also use $S(j)$ to denote $\{f(j)\} \cup N(j)$ and $T(j)$ to denote $V \setminus S(j)$.

Definition 1.2.2. A *BSIP solution* specifies a finite message alphabet Σ, broadcast alphabet Σ_P to be used by the server, and an encoding scheme $\mathcal{E} : \Sigma^n \to \Sigma_P$ such that, for any possible values of $x_1, \ldots, x_n$, every receiver R_j is able to decode the message $x_{f(j)}$ from the value of

$\mathcal{E}(x_1, \dots, x_n)$ together with $\{x_i | i \in N(j)\}$. The minimum encoding length $\ell = \lceil \log_2 |\Sigma_P| \rceil$ for messages that are t bits long (i.e. $|\Sigma| = 2^t$) is denoted by $\beta_t(G)$.

As noted in [54], due to the overhead associated with relaying the side-information map to the server, the main focus is on the case $t \gg 1$.

Definition 1.2.3. The *broadcast rate* of a BSIP instance G is

$$\beta(G) := \inf_{\Sigma, \Sigma_P} \frac{\log |\Sigma_P|}{\log |\Sigma|} \quad \text{s.t. } \Sigma \text{ and } \Sigma_P \text{ are alphabets of a BSIP solution} \tag{1.1}$$

Alternatively, the broadcast rate can be defined using β_t.

$$\beta(G) := \lim_{t \to \infty} \frac{\beta_t(G)}{t} = \inf_t \frac{\beta_t(G)}{t} \tag{1.2}$$

Note that both of these limits always exist [7]. This is interpreted as the average asymptotic number of broadcast bits needed per bit of input, that is, the asymptotic broadcast rate for long messages.

An important special case of the problem arises when there is exactly one receiver for each message, i.e. $m = n$ and $f(j) = j$ for all j. Here the instance can be viewed as a special case of multiple unicast, rather than the general network coding problem. In this case, the side-information map $N(j)$ is equivalently described as the binary relation of pairs (i, j) such that $j \in N(i)$. These pairs can be thought of as the edges of a directed graph on the vertex set $[n]$ or, in case the relation is symmetric, as the edges of an undirected graph. This special case of symmetry allows us the following definition:

Definition 1.2.4. A broadcasting with side information problem on graphs (BSIP-G) instance is given by an undirected graph $G = (V, E)$ where each vertex $v \in V$ corresponds to a message x_v and a receiver R_v that is interested in x_v and knows $\{x_u | (u, v) \in E\}$.

BSIP-G corresponds to the index coding problem introduced by Birk and Kol [9]. It is extensively studied due to its rich connections with graph theory and Ramsey theory. These connections stem from simple relations between broadcast rates and other graph-theoretic parameters. Letting $\alpha(G)$ and $\overline{\chi}(G)$ denote the independence and clique-cover numbers of G, respectively, one has

$$\alpha(G) \leq \beta(G) \leq \beta_1(G) \leq \overline{\chi}(G)\,. \tag{1.3}$$

The first inequality above is due to an independent set being identified with a set of receivers with no mutual information, whereas the last one is obtained by broadcasting the bitwise XOR of the vertices per clique in the optimal clique-cover of G ([8, 9]).

We are also interested in the optimal rate when we require that the code is scalar linear or vector linear.

Definition 1.2.5. The *scalar linear broadcast rate* of a BSIP instance over $\mathbb{F}$, denoted $\lambda_1^{\mathbb{F}}(G)$, is the infimum of all broadcasting solutions in which the message alphabet is the finite field $\mathbb{F}$ and the encoding and decoding functions are linear.

The *scalar linear broadcast rate* is defined as

$$\lambda_1(G) := \inf_{\mathbb{F}} \lambda_1^{\mathbb{F}}(G) \tag{1.4}$$

Similarly, we can define the vector linear broadcast rate.

Definition 1.2.6. The *vector linear broadcast rate over* $\mathbb{F}$ of a BSIP instance, denoted $\lambda^{\mathbb{F}}(G)$, is the infimum of all broadcasting solutions in which the message alphabet is a finite dimensional vector space over a finite field $\mathbb{F}$ and the encoding and decoding functions are linear.

The *scalar linear broadcast rate* is defined as

$$\lambda(G) := \inf_{\mathbb{F}} \lambda^{\mathbb{F}}(G) \tag{1.5}$$

Observe that $\beta \leq \lambda \leq \lambda_1$.

1.2.3 Previous Work

The early work focuses on the more restricted BSIP-G. The first protocols developed are scalar linear codes hinging on a greedy clique-cover (related to the bound $\beta \leq \overline{\chi}$) [9]. Scalar linear coding schemes are expanded further by Bar-Yossef *et al.* [8] who proposed a new class of codes based on a matrix rank minimization problem. The solution to this problem, denoted $\mathrm{minrk}_2(G)$, was shown to achieve the optimal linear scalar capacity over $GF(2)$ and, in particular, to be superior to the clique-cover method, i.e. $\beta_1 \leq \mathrm{minrk}_2 \leq \overline{\chi}$. After establishing $\beta_1(G) = \mathrm{minrk}_2(G)$ for various families of graphs, the authors of [8] conjecture that the equality holds for all graphs; a claim that is refuted in [54] by defining an extension of minrk_2 to general fields, $\mathrm{minrk}^{\mathbb{F}}$, whose optimal solution is exactly $\lambda_1^{\mathbb{F}}$.

Definition 1.2.7. Let $A = (a_{ij})$ be an $n \times n$ matrix over some field $\mathbb{F}$. We say that A *represents* the graph G over $\mathbb{F}$ if $a_{ii} \neq 0$ for all i, and $a_{ij} = 0$ whenever $i \neq j$ and $(i, j) \notin E$. The *minrank* of a directed graph G with respect to the field $\mathbb{F}$ is defined by

$$\mathrm{minrk}^{\mathbb{F}}(G) := \min\{\mathsf{rank}^{\mathbb{F}}(A) : A \text{ represents } G \text{ over } \mathbb{F}\}. \tag{1.6}$$

and

$$\mathrm{minrk}(G) := \min_{\mathbb{F}}\{\mathrm{minrk}^{\mathbb{F}}(G)\}. \tag{1.7}$$

Lubetzky and Stav [54] use $\mathrm{minrk}^{\mathbb{F}}$ along with arguments from Ramsey theory to show that for any fixed $\varepsilon > 0$ there is a family of graphs on n vertices for which $\mathrm{minrk} \leq n^{\varepsilon}$ while $\mathrm{minrk}_2 \geq n^{1-\varepsilon}$. Additionally, they give a related family of graphs for which $\beta \leq n^{\varepsilon}$ while $\lambda_1 = \mathrm{minrk} \geq n^{\frac{1}{2}-\varepsilon}$. Ergo, the upper bounds on β, namely, $\overline{\chi}(G)$, $\mathrm{minrk}_2(G)$, $\mathrm{minrk}(G) = \lambda_1(G)$ are not bounded above by any polynomial function of $\beta(G)$.

Another focus of previous work is on the relationship between β and β_1. The first proof of a separation $\beta < \beta_1$ for graphs is presented by Alon *et al.* in [7]. The proof introduces a new capacity parameter β^*, which informally, is the minimum broadcast length if the network topology is replicated t independent times. Let $t \cdot G$ denote the disjoint union of t copies of G. We define $\beta_t^*(G) := \beta_1(t \cdot G)$, and sub-additivity justifies

$$\beta^*(G) := \lim_{t\to\infty} \frac{\beta_t^*(G)}{t} = \inf_t \frac{\beta_t^*(G)}{t} \tag{1.8}$$

The parameter satisfies $\beta \le \beta^* \le \beta_1$, and [7] shows that the second inequality can be strict using a characterization of β^* as the fractional chromatic number of a certain graph with $2^{|V(G)|}$ vertices. In addition, the paper studies BSIP and constructs several hard instances including ones where $\beta = 2$ while β^* is unbounded and others where $\beta^* < 3$ while β_1 is unbounded.

As with the general network coding problem, prior work on BSIP has been highly successful in bounding the broadcast rate above and below by various parameters (all of which, unfortunately, are either known or suspected to be NP-hard to compute) and in constructing examples that exhibit separations between these parameters. However, it has been less successful at providing general techniques that allow the determination, or even the approximation, of the broadcast rate β for large classes of problem instances. The following two facts, which held true prior to our work [13], starkly illustrate this limitation. (1) Excluding graphs whose trivial lower and upper bounds, $\alpha(G)$ and $\overline{\chi}(G)$, coincide, the exact value of $\beta(G)$ was unknown for *every* graph (and hypergraph) G. (2) It was unknown if the broadcast rate β could be approximated by a polynomial-time algorithm whose approximation ratio improves the trivial factor n by more than a constant factor.[2]

In this work we address both of the open questions stated in the preceding paragraph, give new bounds on β, and study the relationships between the bounds.

[2] When G is a graph (rather than a hypergraph), it is not hard to derive a polynomial-time $o(n)$-approximation from (1.3).

1.2.4 Our Contributions

Chapters 2 - 6 describe joint work with Robert Kleinberg and Eyal Lubetzky that appears in [13, 12]. Our contributions to BSIP encompass five topics: proving bounds on β, determining β exactly on structured instance classes, determining the behavior of bounds under products and sums of BSIP instances, exhibiting gaps between bounds, and approximating β.

Chapter 7 describes work that appears in [10]. We study relationships between network coding rates and cut bounds in a variant of the multiple unicast problem.

We now summarize all of our results and their locations in this work.

Bounds on the Broadcast Rate (Chapter 2)

Strong bounds on β are critical to proving approximations and exact computations. To this end, we extend many previous bounds to their fractional variants and from the BSIP-G to the more general BSIP setting. We also introduce new bounds derived via entropy inequalities.

In Section 2.1 we strengthen the clique-cover upper bound by showing that the fractional clique-cover is also an upper bound, giving $\beta \leq \overline{\chi}_f \leq \overline{\chi}$. Similarly, in Section 2.2 we strengthen the minrank upper bound by defining a fractional minrank, minrk_f, whose optimal solution is rate of the best vector linear solution, giving $\beta \leq \mathrm{minrk}_f \leq \mathrm{minrk}$.

We extend the notions of independent set, clique-cover, and minrank to hypergraphs and use this to extend the bounds α, $\overline{\chi}$, and minrk for BSIP-G to the more general BSIP.

In Section 2.3 we provide a class of information-theoretic linear programs whose solution values bound the broadcast rate. The basic linear program includes inequalities derived from the problem structure as well as the Shannon inequalities. Its solution, denoted b, gives the best known lower bound on β. We extend the linear program by adding non-Shannon inequalities and dimension inequalities. The solution value of the linear program with non-Shannon inequalities gives even better lower bounds on β, and with dimension inequalities it gives lower bounds on λ.

We use two dimension inequalities that are valid for linear functions over fields of odd (resp., even) characteristic but not vice-versa that we derive in Appendix A. We obtain these inequalities by considering the Fano and non-Fano matroids; the former is a matroid that is only realizable in characteristic 2 while the latter is only realizable in odd characteristic and in characteristic 0. For each of the two matroids, we are able to transform a proof of its non-realizability into a much stronger quantitative statement about dimensions of vector spaces over a finite field.

Structured Instances (Chapter 3)

We derive the exact value of $\beta(G)$ for various families of hypergraphs and BSIP and BSIP-G instances by providing matching lower and upper bounds. The lower bounds are obtained by analyzing the LP solution, $b(G)$. The upper bounds are obtained via $\overline{\chi}_f(G)$ and $\mathrm{minrk}(G)$.

In Section 3.1 we consider a class of BSIP instances that are derived from matroids. This builds on work of [27, 28, 22, 24] that established connections between matroids and network coding. In particular, El Rouayheb *et al.* [28] define a correspondence between certain BSIP instances and matroids and show that realizability of a matroid over a field $\mathbb{F}$ is equivalent to linear solvability of the corresponding BSIP. We give a different, and much simpler, correspondence between BSIP instances and matroids. We establish the broadcast rate for BSIP instances that are derived from representable matroids, and we give lower bounds for the broadcast rate of all matroidal BSIP instances.

In Section 3.2.1 we obtain the exact value of $\beta(G)$ for all cycles and cycle-complements. Precisely, $\beta(C_n) = n/2$ and $\beta(\overline{C_n}) = n/\lfloor \frac{n}{2} \rfloor$. This establishes the broadcast rate for the

5-cycle investigated in [7, 8]. In Section 3.2.2 we give the exact value of β for 3-regular Cayley graphs of $\mathbb{Z}_n$ and certain Circulant graphs.

Products and Sums (Chapter 4)

It is known that many graph parameters behave multiplicatively, sub-multiplicatively, or super-multiplicatively on graph products, and additively, sub-additively, or super-additively on graph sums. These insights allow for the analysis of graph parameters on large graphs and, often, the analysis of constructions via graph products and sums. We consider two graph products and a graph sum and extend them to be defined on general BSIP instances. We apply all of these results in Chapter 5 to give constructions yielding gaps between parameters of interest.

In Section 4.1 we consider the lexicographic product. We show that for this product operation it is not hard to compose the codes of the multiplicands to create a code for the product. Hence, β is sub-multiplicative. Further, we demonstrate that entropy based lower bounds proven using linear programming behave super-multiplicatively under lexicographic products. The proof analyzes the dual solutions of the two linear programs. We show how to combine the dual solutions so that the combined dual yields a dual solution of the linear program corresponding to the lexicographic product. Our technique not only applies to the standard linear program that uses only Shannon inequalities but to any family of linear programs constructed using what we call a *tight constraint schema*. In particular, the technique applies to all of the linear programming bounds we consider.

In Section 4.2 we consider an extension of the strong product operation. It was already known that the minrank parameter is sub-multiplicative on products of graphs. We extend this to show that the fractional minrank of BSIP instances is sub-multiplicative on the strong product.

In Section 4.3 we consider the sum, that is, the disjoint union of two BSIP instances. It is known that β^* is additive, but β_1 is not. We show that β and $\lambda^{\mathbb{F}}$ are additive.

Separating Broadcast Rates and Bounds (Chapter 5)

We continue by investigating gaps between the broadcast rate, the linear broadcast rates, and the upper and lower bounds. Our results improve upon several of the best previously known separations. We rely heavily on the techniques developed in the previous two chapters, starting with a small gap given by a structured instance and amplifying the separation via products or sums.

Dougherty *et al.* [22] show that the Shannon bound is not tight for the network coding rate on a multiple unicast problem. In Section 5.1 we use a matroidal BSIP instance and the linear programming bound with the addition of the Zhang-Yeung non-Shannon inequality [67] to demonstrate that Shannon inequalities are not sufficient for the BSIP. That is to say, the lower bound b can be strictly less than β. After our work, Sun *et al.* [62] give a similar example that shows even when restricted to BSIP-G instances b can be strictly less than β.

As mentioned earlier, Lubetzky and Stav [54] show that the scalar linear coding rate can be $\Omega\left(n^{\frac{1}{2}-\varepsilon}\right)$ factor larger than the non-linear rate. But, for the more powerful vector linear coding no gap was known for BSIP and only a 11/10 gap was known for network coding. In Section 5.2 we obtain a $\Omega\left(n^{\frac{1}{2}-\varepsilon}\right)$ separation between the vector linear and non-linear rates for BSIP-G by extending the technique of Lubetzky and Stav from standard minrank to fractional minrank. This implies a separation for the multiple unicast network coding problem. We provide another method for obtaining a polynomial gap between vector linear and non-linear codes via matroids and lexicographic products. We use a matroidal BSIP instance based on the Fano and non-Fano matroids and lower bound their linear coding rate using the extensions of the linear program which adds in the Fano and non-Fano inequalities (derived in Appendix A). Then we amplify it via lexicographic products to get a $\Omega(n^{\varepsilon})$ separation between vector linear and non-linear coding.

In Section 5.3 we again apply the technique of amplifying gaps via lexicographic products to show that the ratio between α and β can be as large as $n^{0.139}$. We amplify the separation of $\beta = 2.5$ and $\alpha = 2$ on the 5-cycle (shown in Section 3.2.1) using the super-multiplicativity of b and sub-multiplicativity of β on lexicographic products. This boosts the ratio β/α polynomially in n on a family of n-vertex graphs. Further, it implies a polynomial separation between the strongest known cut-based bound on the network coding rate in the directed multiple unicast problem, iMeagerness, and the network coding rate, thus improving the previous logarithmic separation.

Lubetzky and Stav's example [54] which gives a separation between scalar linear and non-linear also shows that $\overline{\chi}(G)$ is not bounded above by any polynomial function of $\beta(G)$. In Section 5.4 we strengthen this result by demonstrating $\overline{\chi}_f(G)$ is not bounded above by any function of $\beta(G)$. To do so, we utilize a class of projective Hadamard graphs due to Erdős and Rényi to prove that there is a family of graphs on n vertices with $\beta(G) = 3$ and $\overline{\chi}_f(G) = \Theta(n^{1/4})$. An implication is that the natural heuristic approach based on clique-covers is sometimes very bad.

In Section 5.5 we show that this heuristic can be bad even when $\overline{\chi}_f = \Theta(n)$ instead of $o(n)$. In particular, there is a family of triangle-free graphs on n vertices where $\overline{\chi}_f \geq n/2$, yet the broadcast rate satisfies $\beta \leq \frac{3}{8}n$.

In Section 5.6 we use results on the 5-cycle and the additivity of certain parameters under disjoint graph union (Section 4.3) to show additive separations of $\Omega(n)$ between many of the parameters of interest on instances with n messages.

Polynomial Time Algorithms (Chapter 6)

We provide the first non-trivial approximation algorithm for BSIP. For BSIP-G the inequality $\alpha(G) \leq \beta(G) \leq \overline{\chi}(G)$ implies a $o(n)$-approximation to β using results of [63, 14, 5]. Using the extensions of the bounds α and $\overline{\chi}$ to hypergraphs together with ideas from [63, 14, 5], Section 6.1 provides a $o(n)$-approximation to β for BSIP. In fact, the approximation holds in greater generality for the weighted case, where different messages may have different rates; in the motivating applications this can correspond to a server that holds messages of varying size. The generalization is explained in Section 6.1.1.

For BSIP-G, the equation $\alpha(G) \leq \beta(G) \leq \overline{\chi}(G)$ also implies a number of simple facts including $\beta(G) = 1 \Leftrightarrow \alpha(G) = 1$ and $\beta(G) = n \Leftrightarrow \alpha(G) = n$. These statements further imply polynomial time decision procedures. In Section 6.2 we give a simple characterization for $\beta(G) = 2$ and a polynomial time algorithm which determines if $\beta(G) = 2$ in BSIP.

Beyond Broadcasting (Chapter 7)

In Chapter 7 we analyze parameters of the products of the actual network, rather than a graph representation of the demands of a BSIP instance. We consider the multiple unicast problem in directed networks. We know that the maximum multicommodity flow rate can be factor $\Omega(k)$ smaller than the minimum multicut, but we don't know if the coding rate can be, too [3]. If yes, the construction of Saks *et al.* [60] that shows the $\Omega(k)$ flow-cut gap is a prime candidate for such an instance. But, we show that the example of [60] is no such instance, and, instead, is another example for which the coding rate is an $\Omega(k)$ factor larger than the flow.

The Saks *et al.* construction is the k-fold strong product of a path. We analyze the

[3] Note we can ask this question even though the cut can be smaller than the coding rate in the directed multicut setting

graph by analyzing the coding rate and cut capacity of products in general. In particular, we identify a property of a linear network code that guarantees the code is equal to the minimum cut. We also show that for the strong graph product of any two networks with such codes, this property is preserved. The following describes one consequence of our main result:

> Given a network G in which the optimal multicommodity flow solution consists of a set of disjoint paths, the optimal network coding rate is equal to the minimum multicut in the k-fold strong product of G.

By applying this result to a directed path of length n with source and sink, we give the exact value of the cut in the construction of Saks *et al.*, thereby strengthening their result. It provides an elegant network coding solution for the construction that is a $k - o(k)$ factor larger than the multicommodity flow rate.

CHAPTER 2

BOUNDS ON THE BROADCAST RATE

We introduce bounds on the broadcast rate that use tools from graph theory, linear algebra, and information-theoretic linear programs.

2.1 Graph Theoretic Bounds

Recall that BSIP-G is a special case of the BSIP where the instance is given by an undirected graph rather than a hypergraph (Definition 1.2.4). In this case, as outlined in Equation (1.3), the clique cover number, $\overline{\chi}$, gives an upper bound on β and the independent set number, α, gives a lower bound [8, 9]. In this section we extend the notions of independent set and clique-cover to hypergraphs and we use this to extend the lower and upper bounds, α and $\overline{\chi}$, for BSIP-G to the more general BSIP. We also further extend the clique-cover to the fractional clique-cover.

First we give a lower bound for BSIP via an extension of the independent set. This is critical for the approximation algorithm for general BSIP instances given in Section 6.1.

Definition 2.1.1. An *expanding sequence* of size k is a sequence of directed hyperedges $j_1, \ldots, j_k$ such that

$$f(j_\ell) \notin \bigcup_{i<\ell} S(j_i)$$

for $1 \leq \ell \leq k$.

For graphs, an independent set I corresponds to an expanding sequence. In particular, any sequence of the receivers R_i, $i \in I$ is an expanding sequence because each desires one of $x_i, i \in I$ and knows only messages $x_j, j \notin I$.

Definition 2.1.2. For a BSIP instance G, $\alpha(G)$ is the maximum size of an expanding sequence.

Lemma 2.1.3. *Every BSIP instance G satisfies the bound $\beta(G) \geq \alpha(G)$.*

Proof. The proof is by contradiction. Let $j_1, \ldots, j_k$ be an expanding sequence and suppose that there is an index code that achieves rate $r < k$. Let $J = \{j_1, \ldots, j_k\}$. For $b = \log_2 |\Sigma|$ we have

$$|\Sigma|^k = 2^{bk} > 2^{br} \geq |\Sigma_P|.$$

Let us fix an element $x_i^* \in \Sigma$ for every $i \notin \{f(j) : j \in J\}$, and define Ψ to be the set of all $\vec{x} \in \Sigma^n$ that satisfy $x_i = x_i^*$ for all $i \notin \{f(j) : j \in J\}$. The cardinality of Ψ is $|\Sigma|^k$, so the Pigeonhole Principle implies that the function $\mathcal{E}$, restricted to Ψ, is not one-to-one. Suppose that $\vec{x}$ and $\vec{y}$ are two distinct elements of Ψ such that $\mathcal{E}(\vec{x}) = \mathcal{E}(\vec{y})$. Let i be the smallest index such that $x_{f(j_i)} \neq y_{f(j_i)}$. Denoting j_i by j, we have $x_k = y_k$ for all $k \in N(j)$, because $N(j)$ does not contain $f(j_\ell)$ for any $\ell \geq i$, and the components with indices $j_i, j_{i+1}, \ldots, j_k$ are the only components in which $\vec{x}$ and $\vec{y}$ differ. Consequently receiver j is unable to distinguish between message vectors $\vec{x}, \vec{y}$ even after observing the broadcast message, which violates the condition that j must be able to decode message $f(j)$. □

Next, we consider an extension of the clique-cover to its fractional variant.

Definition 2.1.4. A *fractional clique-cover* of a graph G is a function that assigns a non-negative weight to each clique such that for every node v the total weight assigned to cliques containing v is at least 1. The *size* of the clique-cover is defined to be the sum of all weights. The *fractional clique-cover number* is the minimum size of any fractional clique-cover of G and is denoted $\overline{\chi}_f(G)$.

Just like the clique-cover number, $\overline{\chi}_f$ is NP-hard to compute, yet it has the advantage that for vertex transitive graphs $\overline{\chi}_f(G) = \frac{n}{\alpha(G)}$, and is thus easy to compute for some classes of graphs. Additionally, it is often easier to analyze and bound than the clique-cover number. We make use of this bound to get a tight upper bound on the broadcast rate in Section 3.2 and for an approximation algorithm in Section 6.1.

Now, we consider its extension for hypergraphs and show it is an upper bound on the broadcast rate.

Definition 2.1.5. A *hyperclique* of a BSIP instance $G = (V, E)$ is a subset of hyperedges $\mathcal{J} \subseteq E$ such that for every pair of distinct edges $i, j \in \mathcal{J}$, $f(i) \in S(j)$.

A *fractional hyperclique-cover* is a function that assigns a non-negative weight to each hyperclique such that for every hyperedge j the total weight assigned to hypercliques containing j is at least 1. The *size* of the hyperclique-cover is defined to be the sum of all weights.

For graphs, a clique K corresponds to the hyperclique K: for $u, v \in K$ we have that the receivers R_u and R_v satisfy $f(u) = u$ and $u \in K \subseteq S(v)$.

Definition 2.1.6. For a BSIP instance G, $\overline{\chi}_f(G)$ is the minimum size of a fractional hyperclique-cover of G.

We show that the fractional hyperclique-cover number gives an upper bound on β for any BSIP instance. This also implies that the fractional clique-cover number is an upper bound for BSIP-G.

Lemma 2.1.7. *Every BSIP instance G satisfies the bound $\beta(G) \leq \overline{\chi}_f(G)$.*

The clique-cover number is an upper bound on β because a clique-cover gives a feasible code - in particular a scalar linear code over $\mathbb{F}_2$. Correspondingly, the fractional clique-cover number is an upper bound because a fractional clique-cover gives a feasible code, but a vector linear code.

Proof. The linear program defining $\overline{\chi}_f(G)$ has a variable $w_{\mathcal{J}}$ for every hyperclique $\mathcal{J}$, and a constraint for every receiver j (hyperedge) specifying $\sum_{\mathcal{J}: j \in \mathcal{J}} w_{\mathcal{J}} \geq 1$. This linear program has integer coefficients, and thus G has a fractional hyperclique cover of weight $w = \overline{\chi}_f(G)$ in which the weight $w(\mathcal{J})$ of every hyperclique $\mathcal{J}$ is a rational number. Assume we are given such a fractional hyperclique-cover, and choose an integer d such that $w(\mathcal{J})$ is an integer multiple of $1/d$ for every $\mathcal{J}$. Let $\mathcal{C}$ denote a multiset of hypercliques containing $d \cdot w(\mathcal{J})$ copies of $\mathcal{J}$ for every hyperclique $\mathcal{J}$. Note that the cardinality of $\mathcal{C}$ is $d \cdot w$.

For any hyperclique $\mathcal{J}$, let $f(\mathcal{J})$ denote the set $\bigcup_{j \in \mathcal{J}} \{f(j)\}$. For each $i \in [n]$, let $\mathcal{C}_i$ denote the sub-multiset of $\mathcal{C}$ consisting of all hypercliques $\mathcal{J} \in \mathcal{C}$ such that $i \in f(\mathcal{J})$. Fix a finite field $\mathbb{F}$ such that $|\mathbb{F}| > dw$. Define $\Sigma = \mathbb{F}^d$ and $\Sigma_P = \mathbb{F}^{d \cdot w}$. Let $\{\xi_i^{\mathcal{J}}\}_{\mathcal{J} \in \mathcal{C}_i}$ be a set of vectors in Σ such that any d of these vectors constitute a basis for Σ. The existence of such a set of vectors is guaranteed by our choice of $\mathbb{F}$ with $|\mathbb{F}| > dw \geq d, |\mathcal{C}_i| < dw$. For example, we can take the vectors to be the rows of a $|\mathcal{C}_i| \times d$ Vandermonde matrix.

The encoding function $\mathcal{E}(x_1, \ldots, x_n)$ outputs a $|\mathcal{C}|$-tuple of elements of $\mathbb{F}$, by evaluating the following linear functions of the messages in V:

$$\sum_{\ell \in f(\mathcal{J})} \xi_\ell^{\mathcal{J}} \cdot x_\ell \;\; \forall \mathcal{J} \in \mathcal{C} \tag{2.1}$$

For each receiver j with $i = f(j)$, the set of vectors $\xi_i^{\mathcal{J}}$ with $j \in \mathcal{J}$ is a subset of $\{\xi_i^{\mathcal{J}}\}_{\mathcal{J} \in \mathcal{C}_i}$ of size at least d, and thus contains a basis of Σ. To show that j can decode message $x_i \in \mathbb{F}^d$ it is sufficient to prove that j can determine the value of $\xi_i^{\mathcal{J}}(x_i)$ whenever $j \in \mathcal{J}$. This holds because the public channel contains the value of $\sum_{\ell \in f(\mathcal{J})} \xi_\ell^{\mathcal{J}} \cdot x_\ell$, and receiver j knows x_ℓ for every $\ell \neq i$ in $f(\mathcal{J})$. □

2.2 Linear-Algebraic Bounds

The minrank parameter of a graph G was originally defined by [33]. It was later shown to relate to BSIP, and in particular, to coincide with the scalar linear broadcast rate of a BSIP-G instance G [54, 8] (see Definition 1.2.7). We extend minrank to be defined for a general BSIP instance. Additionally, we extend the idea fractionally. We define an extension of minrank, *fractional minrank*, denoted minrk_f, that corresponds exactly to the vector linear capacity of G, $\lambda(G) = \text{minrk}_f(G)$, and is an upper bound on β strictly greater than minrank. In Section 5.2 we use the fractional minrank to bound the vector linear broadcast rate and obtain a large separation between vector linear and non-linear coding.

Definition 2.2.1. Let $A = (a_{ij})$ be an $n \times m$ matrix whose entries are $k \times k$ matrices over some field $\mathbb{F}$. We say that A *fractionally represents* the BSIP instance $G = ([n], [m])$ over $\mathbb{F}^k$ if a_{ij} is the identity matrix of size k whenever $i = f(j)$, and $a_{ij} = 0$ whenever $i \notin N(j)$. The *fractional minrank* of G is defined by

$$\text{minrk}_f^{\mathbb{F}^k}(G) := \min\{\mathsf{rank}_{\mathbb{F}}(A) : A \text{ fractionally represents } G \text{ over } \mathbb{F}^k\}, \tag{2.2}$$

$$\text{minrk}_f^{\mathbb{F}}(G) := \inf_k \frac{\text{minrk}_f^{\mathbb{F}^k}(G)}{k}, \tag{2.3}$$

and

$$\text{minrk}_f(G) := \inf_{\mathbb{F}}\{\text{minrk}_f^{\mathbb{F}}(G)\}. \tag{2.4}$$

We will prove that minrk_f corresponds exactly to the vector linear capacity, thus implying it is an upper bound on β.

Lemma 2.2.2. *For all BSIP instances G, $\text{minrk}_f^{\mathbb{F}}(G) = \lambda^{\mathbb{F}}(G)$, and thus $\text{minrk}_f(G) = \lambda(G)$.*

Proof. First, we show that given a matrix A that represents G over $\mathbb{F}^k$ and has rank r, we have a vector linear broadcast of rate $\frac{r}{k}$. Regarding A as a $nk \times mk$ matrix rather than a $n \times m$ matrix with entries that are $k \times k$ matrices, let $A = BC$ be a rank factorization of A

such that B is a $nk \times r$ matrix and C is an $r \times mk$ matrix. Each message will be a k-tuple of elements of $\mathbb{F}$. Let x be a length nk row vector of messages. We claim that broadcasting xB is a valid code. This code sends r symbols in $\mathbb{F}$, and thus has the correct rate. To see that it is valid, note that each receiver can decode using the linear functions given by the corresponding columns in C precisely because $A = BC$ represents G.

Showing the other direction consists of simply reversing this process. Given a vector linear code of rate $\frac{r}{k}$ for G with messages that consist of k symbols in $\mathbb{F}$, we can find a rank r matrix A that fractionally represents G over $\mathbb{F}^k$ by taking the product of the encoding and decoding matrices. The vector linear broadcast can be represented by a $nk \times r$ matrix E, with entries in $\mathbb{F}$ and the row (i, z) corresponds to the encoding of the z^{th} symbol of message i. Moreover, the decoding matrix can be represented by a $r \times mk$ matrix, D, with entries in $\mathbb{F}$, and column (j, z) corresponds to the decoding function of the z^{th} symbol for receiver j. Consider the product ED. The fact that receiver j can decode its z^{th} symbol implies that column (j, z) in matrix ED is zero in rows $(i, -)$ for $i \notin N(j)$, row $(f(j), z)$ is non-zero, and rows $(f(z), z')$, $z' \neq z$ are zero. Scaling column (j, z) by the entry in row $(f(j), z)$ gives a matrix that represents G. □

2.3 Linear Program Bounds

We define a class of lower bounds on β similar to numerous results in network coding theory that bound the network coding rate (e.g., [1, 24, 36, 38, 61]) by combining entropy inequalities of two types (see Section 1.1.3). The first is derived from the graph structure. The second is purely information-theoretic and holds for any set of random variables. These are the so-called Shannon and non-Shannon type inequalities. The relevant inequality of the first type for BSIP is the *decoding* constraint. It enforces that for any receiver R the set of messages R knows together with the public channel determine the message R wants to know.

Definition 2.3.1. Let $G = (V, E)$ be a BSIP instance and let S and T be subsets of V. We say that S *decodes* T (denoted $S \rightsquigarrow T$) if $S \subseteq T \subseteq \mathrm{cl}(S)$, where

$$\mathrm{cl}(S) := \{i \in V | \exists\, j \in E \text{ with } f(j) = i \text{ and } N(j) \subseteq S\}$$

is the *closure* of S.

For BSIP-G instances, $A \rightsquigarrow B$ if $A \subseteq B$ and for every $v \in B \setminus A$ all the neighbors of v are in A.

Using this definition, we derive an entropy inequality for BSIP based on the structure of the problem as follows. For any BSIP instance G and broadcast solution P, sample each message independently and uniformly at random to obtain a finite probability space on which the messages and the public channel are random variables. If S is a subset of these random variables, denote the Shannon entropy of the joint distribution of the variables in S by $H(S)$. Then for every $S \rightsquigarrow T$ we have $H(S \cup \{P\}) = H(T \cup \{P\})$ because for any valid solution P any message in $T \setminus S$ can be determined using S and P, and T, P clearly determines S, P.

We will consider lower bounds generated by the following class of linear programs based on this probabilistic view of BSIP.

$$\begin{array}{llll} \min & z_\emptyset & & \\ \text{s.t.} & z_V = |V| & \text{(w) } (\textit{initialize}) & \\ & & & (\text{LP } \mathfrak{B}_A) \\ \forall S \subset T \subseteq V & z_T - z_S \leq |T \setminus \mathrm{cl}(S)| & \text{(x) } (\textit{decode}) & \\ & Az \geq 0 & \text{(z)} & \end{array}$$

The class of linear programs (LP $\mathfrak{B}_A$) has a variable for each subset of V. The first constraint (initialize) expresses the fact that the the broadcast message is determined by the values of the n messages, which are mutually independent. The next group of constraints (decoding) correspond to entropy inequalities derived from the graph structure.

The final line of the LP represents a set of constraints, corresponding to the rows of the matrix A, that are universally valid for any tuple of random variables indexed by the message set V. Alternatively, in the context of restricted classes of encoding and decoding functions (e.g. linear functions) there may be additional inequalities that are specific to that class of functions. In this case the constraint matrix A may incorporate these inequalities and we obtain a linear program with constraints that are valid for this restricted model of index coding but not valid in general.

We will use the following technical definition to instantiate the constraint matrix A.

Definition 2.3.2. A *constraint schema* is given by an index set I and a vector $\vec{\alpha} \in \mathbb{R}^{\mathcal{P}(I)}$. To each index set J it associates a matrix $A(J)$ with columns indexed by elements of $\mathcal{P}(J)$ and rows indexed by elements of

$$\mathcal{Q}(J) = \left\{(S_1, \ldots, S_{|I|}) \in \mathcal{P}(J)^{|I|} \mid \exists T \subseteq J \;\; s.t. \;\; S_i \cap S_j = T \;\; \forall i \neq j \in \{1, \ldots, |I|\}\right\}$$

such that:

$$A(J)_{qS} = \sum_{\substack{T \in \mathcal{P}(I): \\ S = \cup_{i \in T} q_i}} \alpha_T \qquad \forall q \in \mathcal{Q}(J), S \in \mathcal{P}(J),$$

where q_i is the i^{th} component of vector q.

We say that a subset $\Upsilon \subseteq \mathbb{R}^{\mathcal{P}(J)}$ satisfies a constraint schema if $A(J)\vec{d} \geq \vec{0}$, for all $\vec{d} \in \Upsilon$.

To make this more concrete, we consider the submodularity constraint schema. Submodularity is typically expressed as

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \text{ if } S, T \subseteq J \text{ for an index set } J. \qquad (2.5)$$

The constraint schema is providing a formalism to enumerate the constraints this implies for a specific index set J. To write submodularity as a constraint schema we need to rewrite the inequality to eliminate the intersection term. Here, and for the remainder of this work, we will use the concatenation AB to denote the union of two sets $A \cup B$. We now write submodularity as

$$f(AB) + f(BC) \geq f(ABC) + f(B) \text{ for } A, B, C \subseteq J. \tag{2.6}$$

The constraint schema lists all constraints in which $A \cap B = B \cap A = A \cap C$. This subset of constraints implied by Equation (2.6) corresponds exactly to the set of constraints implied by Equation (2.5). To see this notice that for any $S, T \subseteq J$ if we set $A = S, C = T$ and $B = S \cap T$ then the pairwise intersections of A, B, C are all equal to $S \cap T$ and $AB = S, BC = T, ABC = ST$, and $B = S \cap T$. For the other direction, if all the intersections of A, B, C are equal then the realization of Equation (2.5) with $S = A \cup B$ and $T = C \cup B$ is a matching constraint. Now, we can formally define the submodular constraint schema:

Definition 2.3.3. The *submodular constraint schema* is given by index set $I = \{A, B, C\}$ and the vector $\vec{\alpha}$, whose entries are all zero except $\vec{\alpha}_{AB} = \vec{\alpha}_{BC} = 1$, and $\vec{\alpha}_{ABC} = \vec{\alpha}_B = -1$.

The submodularity inequality is satisfied for many types of functions and subsets. It holds if J indexes a set of random variables and f is entropy, if J indexes a set of vector spaces and f is the dimension function, and if J indexes the ground set of a matroid and f is the rank function. These facts are equivalently expressed by the fact that the submodularity constraint schema is satisfied for the subsets of $\mathbb{R}^{\mathcal{P}(J)}$ of entropic vectors, dimension vectors and rank vectors. The following definition of subsets of $\mathbb{R}^{\mathcal{P}(J)}$ will be useful.

Definition 2.3.4. We say a vector $\vec{v} \in \mathbb{R}^{2^n}$, indexed by subsets of $[n]$, is *entropic* if there exist random variables $X_1, X_2, \ldots X_n$ sampled from the same probability space such that $\vec{v}_S = H(\{X_i | i \in S\})$ for all $S \subseteq [n]$.

We use Γ_n^* to denote the set of all entropic vectors in 2^n dimensional space. Its closure is denoted $\overline{\Gamma}_n^*$. See Section 1.1.3 for more background on the sets of entropic vectors.

Definition 2.3.5. We say a vector $\vec{v} \in \mathbb{R}^{2^n}$, indexed by subsets of $[n]$, is a *dimension vector* if there exist vector spaces $W_1, W_2, \dots W_n$ of an underlying vector space such that $\vec{v}_S$ is equal to the dimension of the span of $\{W_i | i \in S\}$ for all $S \subseteq [n]$.

We use Υ_n to denote the set of all dimension vectors in 2^n dimensional space. Further, we use $\Upsilon_n^{\mathbb{F}}$ to denote the subset of Υ_n when we restrict the vector spaces to be over the field $\mathbb{F}$.

Our earlier claims about submodularity can now be written: for all $n \in \mathbb{N}$, Υ_n and $\overline{\Gamma}_n^*$ satisfy the submodularity constraint schema.

We can also instantiate constraint matrix A of LP $\mathfrak{B}_A$ with multiple constraint schemas.

Definition 2.3.6. A *constraint schemata* is given by a collection of constraint schemas $(I_1, \vec{\alpha}_1), (I_2, \vec{\alpha}_1), \dots, (I_k, \vec{\alpha}_k)$. Let $A_i(J)$ be the constraint matrix of $(I_i, \vec{\alpha}_i)$ parameterized by J. To each index set J the constraint schemata associates a matrix

$$A(J) = \begin{pmatrix} A_1(J) \\ \dots \\ A_k(J) \end{pmatrix}.$$

Note that the dimensions match correctly as all matrices $A_i(J)$ have columns indexed by $\mathcal{P}(J)$. Also use $\mathcal{Q}(J)$ to denote the index set of the rows of $A(J)$. $\mathcal{Q}(J)$ is the disjoint union of $\mathcal{Q}_i(J)$ over $i \in \{1, \dots, k\}$.

We are finally ready to state and prove that our LP is a lower bound on the broadcast rate for certain instantiations of A.

Theorem 2.3.7. *Every BSIP instance G satisfies $\mathsf{OPT}(\mathfrak{B}_A(G)) \leq \beta(G)$ for any matrix A given by constraint schemata $C_1, \dots, C_k$ such that $\overline{\Gamma}_n^*$ satisfies C_i for all $n \in \mathbb{N}, i \in [k]$.*

Proof of Theorem 2.3.7. Let $G = (V, E)$. For all $\varepsilon > 0$ there is a solution to G specified by finite alphabets Σ and Σ_P and a valid encoding scheme $\mathcal{E} : \Sigma^n \to \Sigma_P$ such that $\frac{\log |\Sigma_P|}{\log |\Sigma|} = \ell = \beta(G) + \varepsilon$. Sample each message independently and uniformly at random, and consider the input messages and the broadcast message, $\mathcal{E}(\{x_i | i \in V\})$, as random variables. Denote the random variables by $X_i, i \in V$ and P respectively. Let H be the entropy function using log base $|\Sigma|$. This normalization, along with independence of the source messages, gives that $H(\{X_i | i \in S\}) = |S|$ for any subset of V and $H(P) = \ell$.

Now, let $z_S = H(\{X_i | i \in S\} \cup \{P\})$ for $S \subseteq V$. We show that z satisfies all the constraints of the LP $\mathfrak{B}_A$.

The solution z satisfies the first constraint because $H(\{X_i | i \in V\}) = |V|$ and P is determined by our message set.

The decoding constraints $z_T - z_S \le |T \setminus \mathrm{cl}(S)|, \quad S \subseteq T$ hold using submodularity together with the decoding equality we described above. We have that $z_S = z_{\mathrm{cl}(S)}$ because in any valid encoding messages of $\mathrm{cl}(S) \setminus S$ must be determined by S and the broadcast message. Submodularity, which is satisfied for the entropy of any random variables, gives $z_{\mathrm{cl}(S)} + H(\{X_i | i \in T \setminus \mathrm{cl}(S)\}) \ge z_T + H(\emptyset)$. Combining this with $H(\{X_i | i \in S\}) = |S|$ implies the decoding constraint.

Finally, z satisfies the constraints in matrix A because z is a vector giving the joint entropy of a set of $|V|$ random variables and hence is in $\Gamma^*_{|V|}$ which satisfies our constraint schemata by assumption. Let random variable Y_i be given by the joint distribution of X_i and P for all $i \in V$. Recall that $z_S = H(\{X_i | i \in S\} \cup \{P\})$ and notice that $H(\{X_i | i \in S\} \cup \{P\}) = H(\{Y_i | i \in S\})$.

This gives a feasible solution with value $z_\emptyset = H(P) = \ell$. Taking $\varepsilon \to 0$ gives a sequence of upper bounds on $b(G)$ whose values tend to $\beta(G)$, implying our result. □

The simplest constraint matrix A we consider is the empty matrix, giving us LP $\mathfrak{B}_\emptyset$. It turns out that the optimal solution value of $\mathfrak{B}_\emptyset$ is equal to the independent set number, and thus provides an alternate proof that the independent set number is a lower bound on β.

Remark 2.3.8. For any BSIP instance G, $\mathsf{OPT}(\mathfrak{B}_\emptyset(G)) = \alpha(G)$

Proof. First we show $z_\emptyset \geq \alpha(G)$. Decoding implies that $z_V = z_{V\setminus I}$ for any independent set I. Combining that constraint with $z_V = n$ and $z_{V\setminus I} - z_\emptyset \leq |V \setminus I|$ gives that $z_\emptyset \geq |I|$ for any feasible z and independent set I.

To show $z_\emptyset \leq \alpha(G)$ we present a feasible solution to the primal attaining the value $\alpha(G)$,

$$z_S = |S| + \max\{|I| \,:\, I \text{ is an independent set disjoint from } S\}\,, \tag{2.7}$$

We verify that the solution is feasible by checking that it satisfies all the constraints of $\mathcal{B}_\emptyset$. There is no independent set disjoint from V, so $z_V = n$ as needed. To prove the decoding constraint for $S \subseteq T \subseteq V$ let I, J be maximum-cardinality independent sets disjoint from S, T respectively. Note that J itself is disjoint from S, implying $|J| \leq |I|$. Thus we have

$$z_T = |T| + |J| = |S| + |T \setminus S| + |J| \leq |S| + |T \setminus S| + |I| = z_S + |T \setminus S|.$$

□

The primary linear program we consider is the one where constraint matrix A is instantiated with the submodular constraint schema.

Definition 2.3.9. The *LP-Shannon lower bound* of a BSIP instance $G = (V, E)$, denoted $b(G)$, is the optimal solution to LP $\mathfrak{B}_A$, which we will denote simply as $\mathfrak{B}$, where A is given by the submodular constraint schema.

Equivalently, we could write $\mathfrak{B}$ as the following linear program.

$$\begin{array}{lll} \min & z_\emptyset & \\ \text{s.t.} & z_V = |V| & (\textit{initialize}) \\ \forall S \subset T \subseteq V & z_T - z_S \leq |T \setminus \mathrm{cl}(S)| & (\textit{decode}) \\ \forall S, T \subseteq V & z_S + z_T \geq z_{S \cup T} + z_{S \cap T} & (\textit{submod}) \end{array} \qquad (\text{LP } \mathfrak{B})$$

The following is a Corollary of Theorem 2.3.7 because $\overline{\Gamma}^*_n$ satisfies the submodular constraint schema.

Corollary 2.3.10. *Every BSIP instance G satisfies $b(G) \leq \beta(G)$.*

We can further strengthen this lower bound by adding additional constraint schema coming from non-Shannon inequalities. For example, the following is a non-Shannon-type inequality due to Zhang and Yeung [67]. This is the first non-Shannon inequality discovered, i.e. an inequality not implied by non-negativity of conditional mutual information. The Shannon-type inequalities are known to characterize entropic vectors induced by at most three random variables, and so naturally, this inequality is parameterized by a index set of size four.

Definition 2.3.11. The *Zhang-Yeung constraint schema* is given by index set $I = \{A, B, C, D\}$ and the vector $\vec{\alpha} \in \mathbb{R}^{\mathcal{P}(I)}$ with values corresponding to the coefficients in the following inequality:

$$\begin{aligned} 3d_{BD} + {} & 3d_{CD} + 3d_{BC} + d_{AB} + d_{AC} \\ & - 2d_B - 2d_C - d_{AD} - d_D - d_{ABC} - 4d_{BCD} \geq 0 \end{aligned}$$

Theorem 2.3.12 ([67])**.** *For all $n \in \mathbb{N}$, Γ^*_n satisfies the Zhang-Yeung constraint schema.*

Now, we can add the constraint schema of the Zhang-Yeung inequality to the linear program $\mathfrak{B}$ to get an even stronger lower bound.

Definition 2.3.13. The *LP-Zhang-Yeung bound* of a BSIP instance $G = (V, E)$, denoted $b_{ZY}(G)$, is the optimal solution to LP $\mathfrak{B}_A$, denoted as $\mathfrak{B}_{ZY}$, where A is given by the submodular and Zhang-Yeung constraint schemata.

Theorem 2.3.14. *Every BSIP instance G satisfies $b(G) \leq b_{ZY}(G) \leq \beta(G)$.*

This follows immediately from Theorems 2.3.12 and 2.3.7.

There are many instances when the lower bound $b(G)$ is tight. We make use of this extensively to obtain a diverse set of results. We use it to analyze specific structured graphs (Chapter 3), obtain gaps between β and other parameters (Sections 5.2, 5.3), and determine if $\beta = 2$ (Section 6.2). But, the lower bound b is not always tight, and in Section 5.1 we use the stronger parameter b_{ZY} to show that b can be strictly less than β. It is likely that b_{ZY} is also strictly less than β. There are an infinite number of non-Shannon inequalities, and the addition of each one gives us a stronger lower bound, and perhaps strictly stronger lower bound. If we add all such inequalities to the linear program then the optimal solution is equal to β.

2.3.1 Linear Programming Bounds on the Linear Rate

There is a correspondence between vector spaces and linear codes, as seen in the minrank parameter (Definition 1.2.7). Any vector linear code has source and broadcast alphabets that are vector spaces over some finite field $\mathbb{F}$ and each message can be given by a linear function on these vector spaces. If we sample each message independently and uniformly at random, and consider the input messages and the broadcast message as random variables, then the entropy (scaled by $\log |\mathbb{F}|$) of these random variables is given by the rank of the linear transformation defined by the message. This is simply because if the transformation has dimension d, then there are $\mathbb{F}^d$ distinct possible messages, each occurring with equal probability.

Thus, an inequality that holds for dimensions of vector subspaces also holds for the entopic vector of a linear code. That is, the entropic vector of a linear coding function is in $\Upsilon_n^{\mathbb{F}}$.

There are inequalities known to hold for all vectors in Υ_n. The most famous such inequality is the Ingleton Inequality [40] which, along with the Shannon inequalities, characterizes Υ_4. It is an active area of research to find more inequalities (e.g. [45, 23]). We contribute to this effort by deriving two inequalities that bound the vector linear capacity over certain fields. Like many similar inequalities, we derive our inequalities using the non-representability of the Fano and non-Fano matroids over certain fields. We present the inequalities using the constraint schema formalism.

Definition 2.3.15. The *Fano constraint schema* is given by index set $I = \{A, B, C, D, E, F, G\}$ and the vector $\vec{\alpha} \in \mathbb{R}^{\mathcal{P}(I)}$ with values corresponding to the coefficients in the following inequality:

$$\begin{aligned}
2d_{AH} &+ 2d_{BH} + 3d_{CH} + 11d_{GH} + 3d_{ABH} + 2d_{ACH} + 2d_{BCH} \\
&+ d_{ABDH} + d_{ACEH} + d_{AFGH} + d_{BCFH} + d_{BEGH} + d_{CDGH} + d_{ABCGH} \\
&+ d_{ABCDEGH} + d_{ABCDFGH} + d_{ABCEFGH} + 3d_{ABCDEFH} \\
&- 15d_{H} - d_{AGH} - d_{BGH} - d_{CGH} - 4d_{ABCH} - 3d_{ABGH} - 3d_{ACGH} \\
&- 3d_{BCGH} - d_{DEFH} - 6d_{ABCDEFGH} \geq 0
\end{aligned}$$

Definition 2.3.16. The *non-Fano constraint schema* is given by index set $I = \{A, B, C, D, E, F, G\}$ and the vector $\vec{\alpha} \in \mathbb{R}^{\mathcal{P}(I)}$ with values corresponding to the coefficients in the following inequality:

$$\begin{aligned}
&3d_{AH} + 3d_{BH} + 9d_{CH} + 6d_{GH} + 6d_{ABH} + 3d_{ABDH} \\
&\quad +3d_{ACEH} + 3d_{BCFH} + d_{DEFH} + 3d_{ABCGH} \\
&\quad +4d_{ABCDEGH} + 4d_{ABCDFGH} + 4d_{ABCEFGH} \\
&\quad -13d_{H} - 12d_{ABCH} - 3d_{ABGH} - 3d_{ACGH} - 3d_{BCGH} \\
&\quad -12d_{ABCDEFGH} \geq 0
\end{aligned}$$

In Appendix A we derive these inequalities and prove that for all $n \in \mathbb{N}$, $\Upsilon_n^{\mathbb{F}}$ satisfies the Fano inequality when char($\mathbb{F}$) is even and the non-Fano inequality when char($\mathbb{F}$) is odd. These results, along with the proof of Theorem 2.3.7 immediately imply that the corresponding LP bounds we get by using the submodular and Fano (non-Fano) constraint schemata to define matrix A of LP $\mathfrak{B}_A$ give lower bounds on the linear coding rate over fields of even (resp. odd) characteristic.

Definition 2.3.17. The *Fano bound* of a BSIP instance $G = (V, E)$, denoted $b_{\mathcal{F}}(G)$, is the optimal solution to LP $\mathfrak{B}_A$, denoted as $\mathfrak{B}_{\mathcal{F}}$, where A is given by the submodular and Fano constraint schemata. We have that $b_{\mathcal{F}}(G) \leq \lambda^{\mathbb{F}}(G)$ when char(F) is even.

Definition 2.3.18. The *non-Fano bound* of a BSIP instance $G = (V, E)$, denoted $b_{\mathcal{N}}(G)$, is the optimal solution to LP $\mathfrak{B}_A$, denoted as $\mathfrak{B}_{\mathcal{N}}$, where A is given by the submodular and non-Fano constraint schemata. We have that $b_{\mathcal{N}}(G) \leq \lambda^{\mathbb{F}}(G)$ when char(F) is odd.

In Section 5.2.2 we use these bounds to obtain large separations between the linear and non-linear coding rates for a BSIP instance derived from the Fano and non-Fano matroid. Neither the Fano nor non-Fano bound alone gives a lower bound on the linear rate, but the minimum of the value of the two bounds does, as every linear code is either over a field of odd or even characteristic.

CHAPTER 3

STRUCTURED BROADCASTING WITH SIDE INFORMATION INSTANCES

3.1 Matroids

In this section we give a mapping from matroids to BSIP instances in which the dependencies in the corresponding BSIP instance exactly capture the dependencies in the matroid. We demonstrate connections between matroid properties and the broadcast rate of the corresponding BSIP instance, allowing us to bound the broadcast rate of such instances, and for representable matroids, determine it exactly.

There are many equivalent definitions of a matroid. The most useful definition in our setting is given in terms of a rank function.

Definition 3.1.1. A matroid is a pair $M = (E, r)$ where E is a ground set and $r : 2^E \to \mathbb{N}$ is a rank function satisfying

(i) $r(A) \leq |A|$ for all $A \subseteq E$;

(ii) $r(A) \leq r(B)$ for all $A \subseteq B \subseteq E$ (monotonicity);

(iii) $r(A) + r(B) \geq r(A \cup B) + r(A \cap B)$ for all $A, B \subseteq E$ (submodularity).

The rank vector of a matroid, $\vec{\mathbf{r}}(M)$, is a $2^{|E|}$-dimensional vector indexed by subsets of $S \subseteq E$, such that its S-th coordinate is $r(S)$. A subset $S \subseteq E$ is called *independent* if $r(S) = |S|$ and it is called a *basis* of M if $r(S) = |S| = r(E)$. A set S is called *dependent* if it is not independent and S is a *circuit* if it is a minimal dependent set.

Now we describe our matroid to BSIP mapping.

Definition 3.1.2. Let $M = (E, r)$ be a matroid. The BSIP instance *associated* to M, denoted by G_M, has a message set E and a receiver $j_{C,e}$ for each $e \in C$ and circuit $C \subseteq E$ with $f(j_{C,e}) = e$ and $S(j_{C,e}) = C$.

Remark. A similar yet slightly more complicated construction was given in [28]. Our construction is (essentially) a subset of the one appearing there. A construction that maps a matroid to a network coding problem is given in [22, 24]. They prove an analog of Proposition 3.1.3.

It is useful to observe that for any $S \subseteq E$ the closure of S in matroid theory is defined to be

$$\mathrm{cl}(S) = \{x \in E \mid r(S) = r(S \cup \{x\})\} \tag{3.1}$$

thus coinciding with our definition of $\mathrm{cl}(S)$ in the context of the index coding problem G_M (see Definition 2.3.1).

Proposition 3.1.3. *For a matroid $M = (E, r)$, $b(G_M) = |E| - r(E)$.*

Proof. In what follows we will let $n = |E|$ and $r = r(E)$. To show that $b(G_M) \leq n - r$ it suffices to show $z_S = r(S) + n - r$ is a feasible primal solution to the LP $\mathfrak{B}(G_M)$. The feasibility of initialization and submodular constraints follows trivially from the definition of G_M and properties of a matroid. The feasibility of the decoding constraint: $z_T - z_S \leq c_{ST}\ \forall S \subset T$ follows from repeated application of submodularity:

$$\begin{aligned} z_T - z_S = r(T) - r(S) &\leq \sum_{x \in T \setminus S} r(S \cup \{x\}) - r(S) \\ &\leq \sum_{x \in \mathrm{cl}(S)} (r(S \cup \{x\}) - r(S)) \ + \sum_{x \in T \setminus \mathrm{cl}(S)} r(\{x\}) \leq |T \setminus \mathrm{cl}(S)| = c_{ST}. \end{aligned}$$

To prove the reverse inequality, let S be any basis of M and note that $z_\emptyset = z_E - (z_E - z_S) - (z_S - z_\emptyset) \geq n - c_{SE} - c_{\emptyset S} = n - r$. □

The following definition relaxes the notion of a representation for a matroid.

Definition 3.1.4. A matroid $M = (E, r)$ with $|E| = n$ is *under-representable* in d dimensions over a finite field $\mathbb{F}$ if there exists a rank d matrix with entries in $\mathbb{F}$ and n columns indexed by elements of E such that if $r(x \cup S) = r(S)$ then the column indexed by x can be written as a linear combination of the columns indexed by S.

If a matrix represents M then additionally any independent set S corresponds to a set of independent columns. If there exists such a matrix it will have rank $r(E)$, and we say that M is representable.

We next show a relation between under-representations for M over $\mathbb{F}$ and the scalar linear rate $\lambda_1^{\mathbb{F}}$. The following is the analogue of Theorem 8 in [28] for our version of the matroid to index coding mapping.

Theorem 3.1.5. *A matroid $M = (E, r)$ with $|E| = n$ is under-representable in d dimensions over a finite field $\mathbb{F}$ if and only if $\lambda_1^{\mathbb{F}}(G_M) \leq n - d$. In particular, if M is representable over $\mathbb{F}$ then $\lambda_1^{\mathbb{F}}(G_M) = \beta(G_M) = n - r(E)$.*

Proof. Let R be a matrix which under-represents M in d dimensions over $\mathbb{F}$. Without loss of generality, we can assume that R is a $d \times n$ matrix because we can apply row operations that result in all but d rows being zero. These row operations do not effect the rank of sets of columns because we could apply the same operations to a subset of columns independently. Let Q be an $(n - d) \times n$ matrix whose rows span the kernel of R. We will show that Q is a valid encoding matrix for G_M. Let $y \in \mathbb{F}^E$ be some input message set and consider a receiver (x, S), who wishes to decode y_x from $\{y_z : z \in S\}$ and the broadcast message Qy. Extend $\ker(Q)$ arbitrarily into a basis B for $\mathbb{F}^E$ and let $y = y' + y''$ be the unique decomposition according to B such that $y' \in \ker(Q)$. Clearly, $Qy'' = Qy$ since $y' \in \ker(Q)$, hence one can recover y'' from the public channel by triangulating Q. It remains for the receiver (x, S) to recover y'_x. To this end, observe that the rows of R span $\ker(Q)$ and recall that by Definitions 3.1.2 and 3.1.4, column x of R is a linear combination of the columns of

R indexed by S. Since y' is in the row-space of R it follows that y'_x is equal to the exact same linear combination of the components of y' indexed by S, all of which are known to the receiver. Altogether, the receiver can recover both y'_x and y''_x and obtain the message x. As this holds for any receiver, we conclude that Q is a valid encoding matrix and thus $\lambda_1^{\mathbb{F}}(G_M) \leq n - d$. When $d = r(E)$ the inequality is tight because this upper bound coincides with the lower bound given by Proposition 3.1.3.

Conversely, suppose that there exists a scalar linear code for G_M over $\mathbb{F}$ with rate $n - d$, and let Q be a corresponding $(n-d) \times n$ encoding matrix of rank $n-d$. Let R be a $d \times n$ matrix whose rows span the kernel of Q. We claim that R under-represents M. Indeed, consider a receiver (x, S). It is easy to verify that this receiver has a linear decoding function[1] of the form $u^{\mathsf{T}} \cdot Qy + v^{\mathsf{T}} \cdot y_S$ for some vectors u, v, where y_S is the vector formed by restricting y to the indices of S. As Q is a valid encoding matrix for G_M, this evaluates to y_x for any $y \in \mathbb{F}^E$. In particular, if y^{T} is a row of R then $Qy = 0$ and so $v^{\mathsf{T}} \cdot y_S = y_x$, and applying this argument to every row of R verifies that column x of R is a linear combination of the columns of R indexed by S (with coefficients from v). Since this holds for any receiver we have that R under-represents M, as required. □

We conclude this section with a result that will be useful in establishing lower bounds on the value of LP $\mathfrak{B}_A(G_M)$ for alternate constraint matricies A.

Theorem 3.1.6. *Suppose that $M = (E, r)$ is a matroid and A is a matrix such that $A\mathbf{1} = 0$ and $A\vec{\mathbf{r}}(M) \ngeq 0$. Then the value of LP $\mathfrak{B}_A$ is strictly greater than $|E| - r(E)$.*

Proof. We will give a dual solution (w, x, y) to the LP with value strictly greater than $|E| - r(E)$.

[1] This follows e.g. from decomposing y as above into $y' + y''$ where $y' \in \ker(Q)$. By definition y''_x is a linear combination of the Qy entries. Similarly, y'_x must be a linear combination of $\{y_z : z \in S\}$, otherwise there would exist some $y \in \ker(Q)$ with $y_x \neq 0$ and $y_z = 0$ for all $z \in S$, making it indistinguishable to this receiver from $y = 0$.

Recalling the hypothesis $A\vec{\mathbf{r}}(M) \ngeq 0$, let q be a row of A such that $\sum_{S \subseteq E} a_{qS} r(S) < 0$. Let $\mathcal{S}^+ = \{S \subseteq E \mid a_{qS} > 0,\ S \neq E, \emptyset\}$ and $\mathcal{S}^- = \{S \subseteq E \mid a_{qS} < 0,\ S \neq E, \emptyset\}$. Note that the hypothesis that $A\mathbf{1} = 0$ implies that $a_{q\emptyset} + \sum_{S \in \mathcal{S}^+} a_{qS} = -\left(a_{qE} + \sum_{S \in \mathcal{S}^-} a_{qS}\right)$. Assume that A is scaled so $a_{q\emptyset} + \sum_{S \in \mathcal{S}^+} a_{qS} = -\left(a_{qE} + \sum_{S \in \mathcal{S}^-} a_{qS}\right) = 1$. This assumption is without loss of generality since $a_{qE} + \sum_{S \in \mathcal{S}^-} a_{qS}$ is strictly negative, as can be seen from the following calculation:

$$r(E)\Bigg(a_{qE} + \sum_{S \in \mathcal{S}^-} a_{qS}\Bigg) \leq a_{qE} r(E) + \sum_{S \in \mathcal{S}^-} a_{qS} r(S) \leq a_{qE} r(E) + \sum_{S \in \mathcal{S}^-} a_{qS} r(S) + \sum_{S \in \mathcal{S}^+} a_{qS} r(S)$$

$$= \sum_{S} a_{qS} r(S) \quad < \quad 0\,.$$

Define the dual vector y by setting $y_q = 1$ and $y_{q'} = 0$ for rows $q' \neq q$ of A. To define the dual vector x, let us first associate to every set $S \subseteq E$ a matroid basis $b(S)$ such that the set $m(S) = b(S) \cap S$ is a maximal independent subset of S, i.e. $|m(S)| = r(m(S)) = r(S)$. Let $u(S) = S \cup b(S)$. For every $S \in \mathcal{S}^+$, let $x_{\emptyset m(S)} = x_{m(S)S} = a_{qS}$ and for every $S \in \mathcal{S}^-$, let $x_{Su(S)} = x_{u(S)E} = -a_{qS}$. Set all other values of x_{ST} to zero. Finally, set $w = 1$. By construction, (w, x, y) satisfies all of the dual constraints. Using the relations $c_{\emptyset m(S)} = r(S)$, $c_{Su(S)} = r(E) - r(S)$, $c_{m(S)S} = c_{u(S)E} = 0$, we find that the dual LP objective value is

$$\begin{aligned}
|E|\, w - \sum_{S \subset T} c_{ST} x_{ST} &= |E| - \sum_{S \in \mathcal{S}^+} (c_{\emptyset m(S)} + c_{m(S)S}) a_{qS} - \sum_{S \in \mathcal{S}^-} (c_{Su(S)} + c_{u(S)E})(-a_{qS}) \\
&= |E| - \sum_{S \in \mathcal{S}^+} r(S) a_{qS} + \sum_{S \in \mathcal{S}^-} (r(E) - r(S)) a_{qS} \\
&= |E| + \sum_{S \in \mathcal{S}^-} a_{qS} r(E) - \sum_{S} a_{qS} r(S) + a_{q\emptyset} r(\emptyset) + a_{qE} r(E) \\
&= |E| - r(E) - \sum_{S} a_{qS} r(S).
\end{aligned}$$

By hypothesis $\sum_S a_{qS} r(S) < 0$, and the proposition follows. □

3.2 Regular Graphs

In this section we use the lower bound LP $\mathfrak{B}$ and the upper bound $\overline{\chi}_f$ to compute the exact broadcast rate of some classes of BSIP-G instances. To avoid too many subscripts, we will use $z(S)$ to denote the variable z_S in LP $\mathfrak{B}$.

3.2.1 The broadcast rate of cycles and their complements

The following theorem establishes the value of β for BSIP-G instances given by cycles and their complements.

Theorem 3.2.1. *For any integer $n \geq 4$ the n-cycle satisfies $\beta(C_n) = n/2$ whereas its complement satisfies $\beta(\overline{C_n}) = n/\lfloor n/2 \rfloor$. In both cases $\beta_1 = \lceil \beta \rceil$ while $\alpha = \lfloor \beta \rfloor$.*

Proof. As the case of n even is trivial with all the inequalities in (1.3) collapsing into an equality (which is the case for any perfect graph), assume henceforth that n is odd. We first show that $\beta(C_n) = n/2$. Letting $n = 2k+1$ for $k \geq 2$, we aim to prove that $b(C_n) \geq k + 1/2$. This together with Lemma 2.1.7 will imply the required result because $\overline{\chi}_f(C_n) = k + 1/2$. The main idea of this proof, as with the ones to follow, is that we sum together inequalities of the LP $\mathfrak{B}$ so that all the variables cancel out except for $z(\emptyset)$, leaving us with a bound on $z(\emptyset)$.

Denote the vertices V of the cycle by $0, 1, \ldots, 2k$. Further define:

$$\begin{aligned}
E &= \{i \ : \ i \equiv 0 \bmod 2,\ i \neq 2k\} && \text{(Evens)} \\
O &= \{i \ : \ i \equiv 1 \bmod 2\} && \text{(Odds)} \\
E^+ &= \{i \ : \ 0 \leq i \leq 2k-2\} && \text{(Evens decoded)} \\
O^+ &= \{i \ : \ 1 \leq i \leq 2k-1\} && \text{(Odds decoded)} \\
M &= \{i \ : \ 1 \leq i \leq 2k-2\} && \text{(Middle)}.
\end{aligned}$$

Next, consider the following constraints in the LP $\mathfrak{B}$:

$$\begin{aligned}
z(\emptyset) + k &\geq z(E) && \text{(decode)} \\
z(\emptyset) + k &\geq z(O) && \text{(decode)} \\
z(\emptyset) + 1 &\geq z(\{2k\}) && \text{(decode)} \\
z(E) &\geq z(E^+) && \text{(decode)} \\
z(O) &\geq z(O^+) && \text{(decode)} \\
z(E^+) + z(O^+) &\geq z(V) + z(M) && \text{(submod,decode)} \\
z(M) + z(\{2k\}) &\geq z(V) + z(\emptyset) && \text{(submod,decode)} \\
2z(V) &\geq 2(2k+1) && \text{(initialize)}\,.
\end{aligned}$$

Summing and canceling we obtain $z(\emptyset) \geq k + 1/2$ as desired.

It remains to treat complements of odd cycles and show $\beta(\overline{C_n}) = 2 + \frac{1}{k}$. We use the upper bound $\overline{\chi}_f$ from Lemma 2.1.7: $\overline{\chi}_f(\overline{C_n}) = 2 + \frac{1}{k}$ because $\overline{C_n}$ is vertex transitive with independent set number $\frac{1}{k}$. We use the lower bound b. We show $b(\overline{C_n}) \geq 2 + \frac{1}{k}$ by using results that we prove in Section 6.2. There we define a type of BSIP instance called an almost alternating cycle of size k, AAC_k, see Definition 6.2.4, and show that $\beta(\mathsf{AAC}_k) \geq 2 + (k-1)^{-1}$ (Theorem 6.2.5). To give a lower bound on $\overline{C_n}$, we interpret $\overline{C_n}$ as a BSIP instance and show $b(\overline{C_n}) \geq b(\mathsf{AAC}_{k+1})$, which is at least $2 + (k)^{-1}$ as desired.

Interpreting $\overline{C_n}$ as a BSIP gives an instance on a directed hypergraph with vertices $v_1, \ldots, v_n$, and edges $j_1, \ldots, j_n$ such that $f(j_i) = v_i$ and $T(j_i) = \{v_{i-1 \mod n}, v_{i+1 \mod n}\}$. It is sufficient to show that this instance contains an almost alternating cycle of size $k+1$. We claim that vertices $u_1, \ldots, u_{k+1}$ where $u_i = v_{2i-1}$, and edges $e_1, \ldots, e_{k+1}$ where $e_i = j_{2i}, i \in \{1, \ldots, k\}, e_{k+1} = j_1$ form an almost alternating cycle. To see this notice that for $i = 1, \ldots, k$, we have that $T(e_i) = T(j_{2i}) = \{v_{2i-1}, v_{2i+1}\} = \{u_i, u_{i+1}\}$ and $T(e_{k+1}) = T(j_1)$ and therefore contains $v_{2k+1} = u_{k+1}$, and $f(j_1) = v_1 = u_1$.

□

3.2.2 The broadcast rate of cyclic Cayley Graphs

In this section we demonstrate how the same framework of the proof of Theorem 3.2.1 may be applied with a considerably more involved sequence of entropy-inequalities to establish the broadcast rate of two classes of Cayley graphs of the cyclic group $\mathbb{Z}_n$. Recall that a *cyclic Cayley graph* on n vertices with a set of generators $G \subseteq \{1, 2, \ldots, \lfloor n/2 \rfloor\}$ is the graph on the vertex set $\{0, 1, 2, \ldots, n-1\}$ where (i, j) is an edge iff $j - i \equiv g \pmod{n}$ for some $g \in G$.

Theorem 3.2.2. *For any $n \geq 4$, the 3-regular Cayley graph of $\mathbb{Z}_n$ has broadcast rate $\beta = n/2$.*

Theorem 3.2.3 (Circulant graphs)**.** *For any integers $n \geq 4$ and $k \leq \frac{n-1}{2}$, the Cayley graph of $\mathbb{Z}_n$ with generators $\{\pm 1, \ldots, \pm k\}$ has broadcast rate $\beta = n/(k+1)$.*

To simplify the exposition of the proofs of these theorems we make use of the following definition.

Definition 3.2.4. A *slice* of size i in $\mathbb{Z}_n$ indexed by x is the subset of i contiguous vertices on the cycle given by $\{x + j \pmod{n} : 0 \leq j < i\}$.

Proof of Theorem 3.2.2. It is not hard to see that for a cyclic Cayley graph to be 3-regular it must have two generators, 1 and $n/2$, and n must be even. If n is not divisible by four, then it is easy to check that there is an independent set of size $n/2$ and $\overline{\chi}_f$ is also $n/2$. Thus, it immediately follows that $\beta = n/2$. For 3-regular cyclic Cayley graphs where n is divisible by four, α is strictly less than $n/2$. So to prove that $\beta = n/2$ we use the LP $\mathfrak{B}$ to show $b \geq n/2$.

Let $0, 1, 2, \ldots, 4k-1$ be the vertex set of the graph. We assume that any solution z has cyclic symmetry. That is, $z(S) = z(\{s+i | s \in S\})$ for all $i \in [0, 4k-1]$. This assumption is without loss of generality because we can take any LP solution z and find a new one z' that is symmetric and has the same value by setting $z'(S) = \frac{1}{4k}\sum_{i=0}^{4k-1} z(\{s+i|s \in S\})$. The solution z' is feasible because it is simply the average of $4k$ feasible solutions.

In our proof we will be using the following subsets of vertices:

$$\begin{aligned}
[i] &= \{0, 1, 2, \ldots, i-1\} \;\; \text{(a slice of size } i\text{)} \\
D &= \{0, 2, \ldots, 2k-4, 2k-2, 2k+1, 2k+3, \ldots, 4k-5, 4k-3\} \\
D^+ &= \{0, 1, 2, \ldots, 2k-4, 2k-3, 2k-2, 2k+1, 2k+2, 2k+3, \ldots, 4k-4, 4k-3\}\,.
\end{aligned}$$

Observe from Figure 3.1 that $D \rightsquigarrow D^+$. Also note that D^+ is missing only four vertices, two on each side almost directly across from each other, and $|D| = 2k-1$.

We prove $b \geq n/2$ by listing a sequence of constraints in the LP $\mathfrak{B}$ that sum and cancel to give us $z(\emptyset) \geq n/2$. The fact that any two slices of size i have the same z value is used heavily in the sequence of inequalities that make up our proof.

First, we create $2k-1$ $z(D^+)$ terms on the right-hand-side:

$$(2k-2) + z(\emptyset) \geq z(D \setminus \{0\}) \qquad \text{(decode)} \tag{3.2}$$

$$z([1]) + z(D \setminus \{0\}) \geq z(D^+) + z(\emptyset) \qquad \text{(submod , decode)} \tag{3.3}$$

$$(2k-2)((2k-1) + z(\emptyset) \geq z(D^+)) \qquad \text{(decode)} \tag{3.4}$$

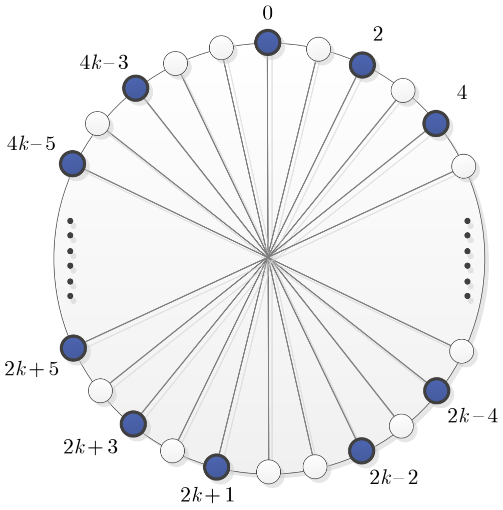


Figure 3.1: A 3-regular cyclic Cayley graph on $4k$ vertices.

Highlighted vertices mark the set D used in the proof of Theorem 3.2.2.

summing to

$$z([1]) + 2k(2k-2) + (2k-2)z(\emptyset) \geq (2k-1)z(D^+) \tag{3.5}$$

Next we apply submodularity and decoding to slices of size $i = 2 \dots 2k$ and an $z(D^+)$ term — canceling all the $z(D^+)$ terms we created on the right-hand-side in the previous step. We pick our slices so that the union decodes a slice missing only two vertices, and the intersection is a slice of size $i-1$. This gives us the following set of inequalities:

$$z(D^+) + z([i]) \geq z([4k-2]) + z([i-1]) \text{ for } 2 \leq i \leq 2k \tag{3.6}$$

Then we again use submodularity and decoding to combine all $2k-1$ of the $z([4k-2])$ terms to get full cycles using this set of inequalities:

$$z([4k-2]) + z([i]) \geq z(V) + z([i-1]) \text{ for } 2k+1 \leq i \leq 4k-2 \tag{3.7}$$

Now summing the inequalities (3.5), (3.6) and (3.7) we are left with:

$$2k(2k-2) + (2k-2)z(\emptyset) \geq (2k-2)z(V)$$

Finally we use the initialization constraint $z(V) \geq n$, yielding:

$$2k(2k-2) + (2k-2)z(\emptyset) \geq (2k-2)4k$$

thus $z(\emptyset) \geq 2k$ for any feasible solution, implying $b \geq 2k = n/2$. □

Proof of Theorem 3.2.3. It is easy to check that $\overline{\chi}_f$ for these graphs is $n/(k+1)$, so it is sufficient to prove that $b \geq n/(k+1)$. As we did in the proof of Theorem 3.2.2 we will assume that our solution z has cyclic symmetry. Suppose that $n \mod (k+1) \equiv j$. Now, consider dividing the cycle into sections of size $k+1$ and let S be the set of vertices consisting of the first k in each complete section ($|S| = k(n-j)/(k+1)$). If $j = 0$ then $\mathrm{cl}(S) = V$ and decoding gives $|S| + z(\emptyset) \geq z(V)$, and our result. If $j > 0$ then decoding implies $\mathrm{cl}(S) = [n-j-1]$. To complete the proof we will show how to combine inequalities to iteratively reduce to the $j = 0$ case.

Lemma 3.2.5. *$(p+1)z([q]) + z([p]) \geq (p+1)z([q+1]) + z(\emptyset)$ for $p \leq q \leq n-1 \in \mathbb{N}^+$.*

Proof. Submodularity and the cyclic symmetry of z implies that the inequality $z([q]) + z([r]) \geq z([q+1])) + z([r-1])$ holds for all $q \in \{1, \ldots, n-1\}$ and $r \leq q$ by considering slices of size q and r with union of size $q+1$ and intersection of size $r-1$. Adding up these inequalities for $r = \{q-p+1, \ldots, q\}$ gives us

$$(p+1)z([q]) \geq pz([q+1]) + z([q-p]). \tag{3.8}$$

Submodularity together with decoding also implies the inequality

$$z([q-p]) + z([p]) \geq z(\emptyset) + z([q+1]) \tag{3.9}$$

by considering disjoint slices $[q-p]$, and $[p]$ separated by one vertex. Summing Equation (3.8) and Equation (3.9) implies our result. □

To complete the proof of Theorem 3.2.3 we start by considering $k+1$ separate instances of the graph and in each apply the decoding inequality to the set S (the set of the first k vertices in each section of size $k+1$):

$$k(n-j)+(k+1)z(\emptyset) \geq (k+1)z(S) \geq (k+1)z([n-j-1]) \tag{3.10}$$

where both inequalities are due to the decoding constraints.

By summing the inequality from Lemma 3.2.5 for $p = k$ and $q \in \{n-j-1, \ldots, n-2\}$ (Note the assumption $k \leq \frac{n-1}{2}$ implies $p = k \leq n-k-1 \leq n-j-1 \leq q$), we obtain:

$$(k+1)z([n-j-1]) + jz([k]) \geq (k+1)z([n-1]) + jz(\emptyset) \tag{3.11}$$

Applying the decoding constraint $z([n-1]) = z(V)$ and the initialize constraint $z(V) = n$ to Equation (3.11) and summing with Equation (3.10) gives

$$(k-j+1)z(\emptyset) + jz([k]) \geq n + jk \tag{3.12}$$

And summing that with j copies of the decoding constraint $k + z(\emptyset) \geq z([k])$ gives $z(\emptyset) \geq \frac{n}{k+1}$, as wanted. □

3.2.3 The broadcast rate of specific small graphs

For any specific graph one can attempt to solve LP $\mathfrak{B}$ to yield a possibly tight lower bound $\beta \geq b$. The following corollary lists a few examples obtained using an AMPL/CPLEX solver.

Fact 3.2.6. *The following graphs satisfy $b = \beta = \overline{\chi}_f$:*

(1) Petersen graph (Kneser graph on $\binom{5}{2}$ vertices): $n = 10$, $\alpha = 4$ and $\beta = 5$.

(2) Grötzsch graph (smallest triangle-free graph with $\chi = 4$*):* $n = 11$*,* $\alpha = 5$ *and* $\beta = \frac{11}{2}$*.*

(3) Chvátal graph (smallest triangle-free 4*-regular graph with* $\chi = 4$*):* $n = 12$*,* $\alpha = 4$ *and* $\beta = 6$*.*

CHAPTER 4

PRODUCTS AND SUMS OF BROADCASTING WITH SIDE INFORMATION INSTANCES

The power of our upper and lower bounds extends only so far as our ability to analyze them. We know of no way to efficiently compute the linear programming bounds, and given that the number of variables and constraints are exponential in the instance size, we expect it to be intractable even for small instances. But, as we saw in the previous section, when our instances are structured we can find structure in the linear program as well, and use that structure to find its value.

The other setting in which we can analyze our bounds is when our instance is built via a product operation. In particular, we show that under an extension of the lexicographic graph product many parameters of a BSIP instance behave sub-multiplicatively and/or super-multiplicatively. In Chapter 5 we use this extensively to amplify small gaps between parameters into gaps that are polynomial in the instance size.

4.1 Lexicographic Products

We begin by defining the lexicographic product operation for BSIP instances.

Definition 4.1.1. The lexicographic product of BSIP instances G, F, denoted by $G \bullet F$, is a BSIP instance whose vertex set is the Cartesian product $V(G) \times V(F)$. The edge set of $G \bullet F$ contains a directed hyperedge j for every pair of hyperedges $(j_G, j_F) \in E(G) \times E(F)$ with $f(j) = (f(j_G), f(j_F))$ and $N(j) = (N(j_G) \times V(F)) \cup (\{f(j_G)\} \times N(j_F))$. Denote by $G^{\bullet n}$ the n-fold lexicographic power of G.

Remark. In the special case where G and F are BSIP-G instances the above definition coincides with the usual lexicographic graph product (where $G \bullet F$ has the vertex set $V(G) \times V(F)$ and an edge from (u, v) to (u', v') iff either $(u, u') \in E(G)$ or $u = u'$ and $(v, v') \in E(F)$).

It is well known that the independence number and fractional clique-cover number of a graph G are multiplicative under the lexicographic product. Thus, both our lower and upper bounds on β for BSIP-G instances are multiplicative, giving some indication that the broadcast rate itself might behave nicely under this product operation.

4.1.1 β Under Lexicographic Products

Theorem 4.1.2. *The broadcast rate is sub-multiplicative under the lexicographic product. That is, $\beta(G \bullet F) \leq \beta(G)\,\beta(F)$ for any two BSIP instances G and F.*

Proof. Let $\varepsilon > 0$ and, recalling the definition of β in (1.2) as the limit of β_t/t, let K be a sufficiently large integer such that for all $t \geq K$ we have $\beta_t(G)/t \leq \beta(G) + \varepsilon$ as well as $\beta_t(F)/t \leq \beta(F) + \varepsilon$. Let $\Sigma = \{0, 1\}^K$ and consider the following scheme for the index coding problem on $G \bullet F$ with input alphabet Σ, which will consist of an inner and an outer code.

Let $\mathcal{E}_F$ denote an encoding function for F with input alphabet Σ achieving an optimal rate, i.e. minimizing $\log(|\Sigma_P|)/\log(|\Sigma|)$. For each $v \in V(G)$, the inner code applies $\mathcal{E}_F$ to the $|V(F)|$-tuple of messages indexed by the set $\{v\} \times V(F)$, obtaining a message x_v. Note that our assumption on $|\Sigma|$ implies that the length of x_v is equal to K' for some integer K' such that $K \leq K' \leq (\beta(F) + \varepsilon)K$. Next, let $\mathcal{E}_G$ denote an optimal encoding function for G with input $\{0, 1\}^{K'}$. The outer code applies $\mathcal{E}_G$ to $\{x_v\}_{v \in V(G)}$ and the assumption on K ensures its output is at most $(\beta(G) + \varepsilon)K'$ bits long.

To verify that the scheme is a valid index code, consider a receiver j in $G \bullet F$ with $f(j) = (f(j_G), f(j_F))$ and $N(j) = (N(j_G) \times V(F)) \cup (\{f(j_G)\} \times N(j_F))$. To decode $f(j)$, the receiver first computes x_v for all $v \in N(j_G)$. Since $\mathcal{E}_G$ is valid for G, receiver j can compute $x_{f(j_G)}$, and since $\mathcal{E}_F$ is valid for F, this receiver can use the messages indexed by $\{f(j_G)\} \times N(j_F)$ along with $x_{f(j_G)}$ to compute $(f(j_G), f(j_F))$.

Altogether, we have an encoding of K bits using at most $(\beta(F)+\varepsilon)(\beta(G)+\varepsilon)K$ bits of the public channel, and the required result follows from letting $\varepsilon \to 0$. □

4.1.2 LP bounds under Lexicographic products

We identify some axioms on constraint schemata that constitute a sufficient condition for the LP value to be super-multiplicative.

Definition 4.1.3. Let $\mathbf{1}$ be the $\mathcal{P}(J)$-indexed vector such that $\mathbf{1}_S = 1$ for all S, and for all $i \in J$ let $\mathbf{1}_i$ be the vector where $(\mathbf{1}_i)_S = 1$ for all S containing i and otherwise $(\mathbf{1}_i)_S = 0$. We say that a constraint schemata is *tight* if $A(J)\mathbf{1} = A(J)\mathbf{1}_i = 0$ for every index set J and element $i \in J$.

It may be possible that all constraint schemas have an equivalent tight constraint schema. In Theorem A.0.6 we show how to find an equivalent tight schema for any constraint schema that is satisfied by $\Upsilon_n^{\mathbb{F}}$. Chan *et al.* [18] took a similar approach to show that all inequalities satisfied by $\overline{\Gamma}_n^*$ are *balanced*, capturing the equality $A(I)\mathbf{1}_i = 0$, part of our notion of tight.

Lemma 4.1.4. *Constraint schemata* $(I_1, \vec{\alpha}_1), (I_2, \vec{\alpha}_2), \ldots, (I_k, \vec{\alpha}_k)$ *is tight if for all* $j \in \{1, \ldots k\}$ *we have that* $\vec{\alpha}_j^\intercal \mathbf{1} = \vec{\alpha}_j^\intercal \mathbf{1}_i = 0$ *for all* $i \in I_j$.

All the constraint schemas defined in Section 2.3 satisfy this property. It is easy to verify this for the submodularity and Zhang-Yeung constraint schemas. It is easy, but more tedious, to verify this for the Fano and non-Fano constraint schemas, and it is additionally proven as part of their derivations in Theorems A.0.7 and A.0.8.

Proof. Let $(I, \vec{\alpha})$ be an arbitrary constraint schema in our constraint schemata. It is sufficient to prove that for any index set J and row $q \in \mathcal{Q}(J)$ of $A(J)$ of constraint schema $(I, \vec{\alpha})$ that $A(J)\mathbf{1} = A(J)\mathbf{1}_i = 0$ for all $j \in J$.

We begin by calculating row q of $A(J)\mathbf{1}$. Let $q(T) := \cup_{i \in T} S_i$ for $q = (S_1, \ldots, S_{|I|})$.

$$\sum_{S \in \mathcal{P}(J)} A(J)_{qS} = \sum_{S \in \mathcal{P}(J)} \sum_{\substack{T \in \mathcal{P}(I) \\ S = q(T)}} \alpha_T = \sum_{T \in \mathcal{P}(I)} \alpha_T = \vec{\alpha}^\mathsf{T} \mathbf{1} = 0$$

Now we calculate row q $A(J)\mathbf{1}_j$ for an arbitrary $j \in J$.

$$\sum_{\substack{S \in \mathcal{P}(J) \\ j \in S}} A(J)_{qS} = \sum_{\substack{S \in \mathcal{P}(J) \\ j \in S}} \sum_{\substack{T \in \mathcal{P}(I) \\ q(T) = S}} \alpha_T = \sum_{\substack{T \in \mathcal{P}(I) \\ j \in q(T)}} \alpha_T$$

At this point the argument splits into three cases. Let $q = (S_1, \ldots, S_{|I|})$. If $j \notin S_i$ for any $i \in I$ then the right side is an empty sum and clearly equals 0. If $j \in S_i$ for all $i \in I$ then the right side is $\vec{\alpha}^\mathsf{T}\mathbf{1}$, which equals 0. Otherwise, there is a unique $i \in I$ such that $j \in S_i$ because by definition $q \in \mathcal{Q}(J)$ implies all pairwise intersections of S_i and S_j are equal. The right side of the equation above is thus equal to $\vec{\alpha}^\mathsf{T}\mathbf{1}_i$, which equals 0.

□

Definition 4.1.5. Let J, K be an index sets and let $A(J), A(K)$ be the constraint matrices parameterized by J and K of constraint schemata C. The rows of $A(J)$ are indexed by $\mathcal{Q}(J)$ and columns are indexed by $\mathcal{P}(J)$. Let h be any Boolean lattice homomorphism[1] $h : \mathcal{P}(J) \to \mathcal{P}(K)$. Let P_h be a matrix representing the linear transformation h induces on $\mathbb{R}^{\mathcal{P}(J)} \to \mathbb{R}^{\mathcal{P}(K)}$. More specifically, P_h has zeros everywhere except $(P_h)_{h(S)S} = 1$.

We say that a constraint schemata C is *homomorphic* if there exists a $\mathcal{Q}(K) \times \mathcal{Q}(J)$ non-negative matrix Q_h such that

$$A(K)^\mathsf{T} Q_h = P_h A(J)^\mathsf{T}$$

[1] A Boolean lattice homomorphism preserves unions and intersections, but does not necessarily map the empty set to the empty set nor the universal set to the universal set, and does not necessarily preserve complements.

for all Boolean lattice homomorphisms h.

Lemma 4.1.6. *Every constraint schemata* $(I_1, \vec{\alpha}_1), (I_2, \vec{\alpha}_2), \ldots, (I_k, \vec{\alpha}_k)$ *is homomorphic.*

Proof. First, we show that every constraint schemata given by a single constraint schema $(I, \vec{\alpha})$ with constraint matrix A is homomorphic. Let $h : \mathcal{P}(J) \to \mathcal{P}(K)$ be a Boolean lattice homomorphism. We let Q_h be a $\mathcal{Q}(K) \times \mathcal{Q}(J)$ matrix that is zeros everywhere except

$$(Q_h)_{h(q)q} = 1 \;\; \forall q \in \mathcal{Q}(J).$$

where $h(q) = \big(h(S_1), h(S_2), \ldots, h(S_{|I|})\big)$ for $q = (S_1, S_2, \ldots, S_{|I|})$ is the natural extension of the homomorphism h to tuples. Notice that $\big(h(S_1), h(S_2), \ldots, h(S_{|I|})\big) \in \mathcal{Q}(K)$ because if $S_i \cap S_j = T$ for all distinct $i, j \in I$ then $h(S_i) \cap h(S_j) = h(S_i \cap S_j) = h(T)$ for all distinct $i, j \in I$ because Boolean lattice homomorphisms preserve intersections.

Let $R = A(K)^{\mathsf{T}} Q_h$, $R' = P_h A(J)^{\mathsf{T}}$. Our goal is to show that $R = R'$. We verify the entry $S, q \in \mathcal{P}(K) \times \mathcal{Q}(J)$ of R and R' are equal. Recalling the definitions of P_h, Q_h we see that

$$R_{Sq} = A(K)_{h(q)S}$$
$$R'_{Sq} = \sum_{S' : h(S') = S} A(J)_{qS'}$$

Now, recall that entry $q, S \in \mathcal{Q}(J) \times \mathcal{P}(J)$ of $A(J)$ is $\sum_{\substack{T \in \mathcal{P}(I),\\ S = \cup_{i \in T} q_i}} \alpha_T$, where q_i is the i^{th} subset of tuple q. From this we see that

$$R_{Sq} = \sum_{\substack{T \in \mathcal{P}(I) \\ S = \cup_{i \in T} h(q_i)}} \alpha_T$$
$$R'_{Sq} = \sum_{S' : h(S') = S} \; \sum_{\substack{T \in \mathcal{P}(I) \\ S' = \cup_{i \in T} q_i}} \alpha_T$$

Combining the sums in the second equality gives a sum over $T \in \mathcal{P}(I), h(\cup_{i \in T} q_i) = S$, which is identical to the summation in the top sum because Boolean lattice homomorphisms preserve unions, giving $R_{Sq} = R'_{Sq}$ as needed.

Now, let A be the constraint matrix of the constraint schemata. For all index sets J and vectors $v \in \mathbb{R}^{\mathcal{P}(J)}$, the constraint matrix satisfies

$$A(J)\mathbf{v} = \begin{pmatrix} A_1(J) \\ \vdots \\ A_k(J) \end{pmatrix} \mathbf{v} = \begin{pmatrix} A_1(J)\mathbf{v} \\ \vdots \\ A_k(J)\mathbf{v} \end{pmatrix}$$

From our argument above that a single constraint schema yields a homomorphic constraint matrix, we have Q_{ih} such that $A_i(K)^\mathsf{T} Q_{ih} = P_h A_i(J)^\mathsf{T}$ for all $i \in [k]$. Letting Q_h be

$$\begin{pmatrix} Q_{1h} & 0 & \dots & 0 \\ 0 & Q_{2h} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & Q_{kh} \end{pmatrix},$$

we have

$$\begin{aligned} A(K)^\mathsf{T} Q_h &= \begin{pmatrix} A_1(K)^\mathsf{T} Q_{1h} & \dots A_k(K)^\mathsf{T} Q_{kh} \end{pmatrix} \\ &= \begin{pmatrix} P_h A_1(J)^\mathsf{T} & \dots & P_h A_k(J)^\mathsf{T} \end{pmatrix} = P_h A(J)^\mathsf{T}, \end{aligned}$$

which confirms that the constraint schemata is homomorphic. □

We can now state our main result.

Theorem 4.1.7. *Let A be given by a tight constraint schemata. For every BSIP instance G let $\rho(G)$ denote the optimum of the LP $\mathfrak{B}_A$. Then for every two index coding problems G and F, we have $\rho(G \bullet H) \geq \rho(G)\,\rho(F)$.*

Observe that this holds if we instantiate A with any constraint schemata containing constraint schema defined in Section 2.3.

Corollary 4.1.8. *The optimal solutions to LPs $\mathfrak{B}, \mathfrak{B}_{ZY}, \mathfrak{B}_{\mathcal{F}}$, and $\mathfrak{B}_{\mathcal{N}}$ given by $b, b_{ZY}, b_{\mathcal{F}}$, and $b_{\mathcal{N}}$ are super-multiplicative.*

$$
\begin{array}{lll}
\min & z_\emptyset & \\
\text{s.t.} & z_I = |I| & (w) \\
\forall S \subset T & z_T - z_S \leq c_{ST} \triangleq |T \setminus \mathrm{cl}(S)| & (x) \\
 & Az \geq 0 & (y)
\end{array}
$$

$$
\begin{array}{ll}
\max & |I| \cdot w - \sum_{S \subset T} c_{ST} x_{ST} \\
\text{s.t.} & \sum_q a_{qS} y_q + \sum_{T \supset S} x_{ST} - \sum_{T \subset S} x_{TS} = 0 \quad \forall S \neq \emptyset, I \\
 & \sum_q a_{q\emptyset} y_q + \sum_{T \neq \emptyset} x_{\emptyset T} = 1 \\
 & \sum_q a_{qI} y_q - \sum_{T \neq I} x_{TI} + w = 0 \\
 & x, y \geq 0
\end{array}
$$

Figure 4.1: LP $\mathfrak{B}_A$ and its dual.

Proof of Theorem 4.1.7. Our analysis will depend on the dual linear program, as shown in Figure 4.1.

It will be useful to rewrite the constraint set of the dual LP in a more succinct form. First, if x is any vector indexed by pairs S, T such that $S \subset T \subseteq I$, let $\nabla x \in \mathbb{R}^{\mathcal{P}(I)}$ denote the vector such that for all S, $(\nabla x)_S = \sum_{T \supset S} x_{ST} - \sum_{T \subset S} x_{TS}$. Next, for a set $S \subseteq I$, let e_S denote the standard basis vector vector in $\mathbb{R}^{\mathcal{P}(I)}$ whose S component is 1. Then the entire constraint set of the dual LP can be abbreviated to the following:

$$
A^\mathsf{T} y + \nabla x + w e_I = e_\emptyset \,, \quad x, y \geq 0 \,. \tag{4.1}
$$

Some further simplifications of the dual can be obtained using the fact that the constraint schemata is tight. For example, multiplying the left and right sides of (4.1) by the row vector $\mathbf{1}^\mathsf{T}$ gives

$$
\mathbf{1}^\mathsf{T} A^\mathsf{T} y + \mathbf{1}^\mathsf{T} \nabla x + w = 1 \,.
$$

By the tightness of the constraint schemata $\mathbf{1}^\mathsf{T} A^\mathsf{T} = 0$. It is straightforward to verify that $\mathbf{1}^\mathsf{T} \nabla x = 0$ and after eliminating these two terms from the equation above, we find simply that $w = 1$. Similarly, if we multiply the left and right sides of (4.1) by the row vector $\mathbf{1}_i^\mathsf{T}$ and substitute $w = 1$, we obtain $\mathbf{1}_i^\mathsf{T} A^\mathsf{T} y + \mathbf{1}_i^\mathsf{T} \nabla x + 1 = 0$ and consequently (again by the tightness) we arrive at $1 = -\mathbf{1}_i^\mathsf{T} \nabla x$. At the same time, $-\mathbf{1}_i^\mathsf{T} \nabla x = \sum_{\substack{S \subset T \\ i \in T \setminus S}} x_{ST}$ by definition of ∇x, hence summing over all $i \in I$ yields

$$|I| = \sum_{S \subset T} |T \setminus S| \, x_{ST}.$$

Plugging in this expression for $|I|$ and $w = 1$, and letting $c_{ST} := |T \setminus \mathrm{cl}(S)|$, the LP objective of the dual can be rewritten as

$$|I| - \sum_{S \subset T} c_{ST} x_{ST} = \sum_{S \subset T} \left(|T \setminus S| - c_{ST} \right) x_{ST} = \sum_{S \subset T} |T \cap (\mathrm{cl}(S) \setminus S)| \, x_{ST},$$

where the last equation used the fact that $c_{ST} = |T \setminus \mathrm{cl}(S)|$. We now define

$$d(S, T) = |T \cap (\mathrm{cl}(S) \setminus S)|$$

and altogether we arrive at the following reformulation of the dual LP.

$$\begin{aligned} \max \quad & \textstyle\sum_{S \subset T} d(S, T) \, x_{ST} \\ \text{s.t.} \quad & A^\mathsf{T} y + \nabla x = e_\emptyset - e_I \\ & x, y \geq 0 \, . \end{aligned} \tag{4.2}$$

Now suppose that $(\xi^G, \eta^G), (\xi^F, \eta^F)$ are optimal solutions of the dual LP for G, F, achieving objective values $\rho(G)$ and $\rho(F)$, respectively. (Here ξ, η play the role of x, y from (4.2), resp.) We will show how to construct a pair of vectors $(\xi^{G \bullet F}, \eta^{G \bullet F})$ that is feasible for the dual LP of $G \bullet F$ and achieves an objective value of at least $\rho(G) \, \rho(F)$. The construction is as follows. Let $g : \mathcal{P}(V(G)) \to \mathcal{P}(V(G \bullet F))$ be the mapping $g(X) = X \times V(F)$. For sets $S \subset T \subseteq V(G)$, let $h^{ST} : \mathcal{P}(V(F)) \to \mathcal{P}(V(G \bullet F))$ be the mapping $h^{ST}(X) = (T \times X) \cup (S \times V(F))$. Observe that both mappings are Boolean lattice homomorphisms.

To gain intuition about the mappings g, h^{ST} it is useful to think of obtaining the vertex set of $G \bullet F$ by replacing every vertex of G with a copy of F. Here $g(\{v\})$ maps the vertex v in G to the copy of F that replaces v. The mapping $h^{ST}(\{u\})$ maps a vertex u in F to the vertex u in the copies of F that replace vertices in T, and then adds the set $\{u\} \times V(F)$.

Recall that our constraint schemata is homomorphic by Lemma 4.1.6, and thus by Definition 4.1.5, for every Boolean lattice homomorphism $h : \mathcal{P}(I) \to \mathcal{P}(J)$ we have matrices P_h, Q_h such that $A(J)^\mathsf{T} Q_h = P_h A(I)^\mathsf{T}$. It is also useful to define a matrix R_h as follows: the columns and rows of R_h are indexed by pairs $S \subset T \subseteq I$ and $X \subset Y \subseteq J$, respectively, with the entry in row XY and column ST being equal to 1 if $X = h(S)$ and $Y = h(T)$, otherwise 0. Under this definition,

$$\nabla(R_h x) = P_h \nabla x \quad \text{ for any } x \in \mathbb{R}^{\mathcal{P}(I)}\,. \tag{4.3}$$

Indeed, if $x = e_{S,T}$ for some $S \subset T \subseteq I$ then $\nabla e_{S,T} = e_S - e_T$ and so $P_h\, e_{S,T} = e_{h(S)} - e_{h(T)}$, whereas $\nabla(R_h e_{S,T}) = \nabla(e_{h(S),h(T)}) = e_{h(S)} - e_{h(T)}$.

We may now define

$$\xi^{G\bullet F} = \sum_{S\subset T} (\xi^G)_{ST}\, (R_{h^{ST}}\, \xi^F)\,, \tag{4.4}$$

$$\eta^{G\bullet F} = Q_g\, \eta^G + \sum_{S\subset T} (\xi^G)_{ST}\, (Q_{h^{ST}}\, \eta^F)\,. \tag{4.5}$$

In words, the dual solution for $G \bullet F$ contains a copy of the dual solution for F lifted according to h^{ST} for every pair $S \subset T$ and one copy of the dual solution of G lifted according to g. The feasibility of $(\xi^{G\bullet F}, \eta^{G\bullet F})$ will follow from multiple applications of the homomorphic property of the constraint schemata and the feasibility of (ξ^F, η^F) and (ξ^G, η^G), achieved by the following claim.

Claim 4.1.9. *The pair* $(\xi^{G\bullet F}, \eta^{G\bullet F})$ *as defined in* (4.4),(4.5) *is a feasible dual solution.*

Proof. The matrices Q_g, $R_{h^{ST}}$, $Q_{h^{ST}}$ all have $\{0,1\}$-valued entries thus clearly $\xi^{G\bullet F}, \eta^{G\bullet F} \geq$

0. Letting $A = A(G \bullet F)$, we must prove that $A^\mathsf{T}\eta^{G\bullet F} + \nabla\xi^{G\bullet F} = e_\emptyset - e_{V(G\bullet F)}$. Plugging in the values of $(\xi^{G\bullet F}, \eta^{G\bullet F})$ we have

$$\begin{aligned} A^\mathsf{T}\eta^{G\bullet F} + \nabla\xi^{G\bullet F} &= A^\mathsf{T}Q_g\eta^G + \sum_{S\subset T}(\xi^G)_{ST}\,(A^\mathsf{T}Q_{h^{ST}}\eta^F) + \sum_{S\subset T}(\xi^G)_{ST}\,\nabla(R_{h^{ST}}\,\xi^F)\,, \\ &= P_gA(G)^\mathsf{T}\eta^G + \sum_{S\subset T}(\xi^G)_{ST}\,\left(P_{h^{ST}}A(F)^\mathsf{T}\eta^F + \nabla(R_{h^{ST}}\,\xi^F)\right)\,. \end{aligned} \tag{4.6}$$

where the second equality applied the homomorphic property of the constraint schemata. To treat the summation in the last expression above, recall (4.3) which implies that

$$P_{h^{ST}}A(F)^\mathsf{T}\eta^F + \nabla(R_{h^{ST}}\,\xi^F) = P_{h^{ST}}A(F)^\mathsf{T}\eta^F + P_{h^{ST}}\nabla\xi^F = P_{h^{ST}}(e_\emptyset - e_{V(F)})\,, \tag{4.7}$$

with the last equality due to the fact that (ξ^F, η^F) achieves the optimum of the dual LP for F. Recalling that $P_h e_S = e_{h(S)}$ for any h and combining it with the facts $h^{ST}(\emptyset) = S\times V(F)$ and $g(S) = S\times V(F)$ gives $P_{h^{ST}}e_\emptyset = e_{S\times V(F)} = P_g e_S$. Similarly, since $h^{ST}(V(F)) = T\times V(F)$ we have $P_{h^{ST}}e_{V(F)} = e_{T\times V(F)} = P_g e_T$, and plugging these identities in (4.7) combined with (4.6) gives:

$$A^\mathsf{T}\eta^{G\bullet F} + \nabla\xi^{G\bullet F} = P_g\left[A(G)^\mathsf{T}\eta^G + \sum_{S\subset T}(\xi^G)_{ST}\,(e_S - e_T)\right].$$

Collecting together all the terms involving e_S for a given $S \in \mathcal{P}(I)$, we find that the coefficient of e_S is $\sum_{T\supset S}(\xi^G)_{ST} - \sum_{T\subset S}(\xi^G)_{ST} = (\nabla\xi^G)_S$. Hence,

$$A^\mathsf{T}\eta^{G\bullet F} + \nabla\xi^{G\bullet F} = P_g\left[A(G)^\mathsf{T}\eta^G + \nabla\xi^G\right] = P_g\left[e_\emptyset - e_{V(G)}\right] = e_\emptyset - e_{V(G\bullet F)}\,,$$

where the second equality was due to (ξ^G, η^G) achieving the optimum of the dual LP for G. □

To finish the proof, we must evaluate the dual LP objective and show that it is at least $\rho(G)\,\rho(F)$, as the next claim establishes:

Claim 4.1.10. *The LP objective for the dual solution given in Claim 4.1.9 has value at least $\rho(G)\,\rho(F)$.*

Proof. To simplify the notation, throughout this proof we will use K, L to denote subsets of $V(G \bullet F)$ while referring to subsets of $V(G)$ as S, T and to subsets of $V(F)$ as X, Y. We have

$$\begin{aligned}\sum_{K\subset L} d(K,L)(\xi^{G\bullet F})_{KL} &= \sum_{K\subset L} d(K,L) \sum_{S\subset T} (\xi^G)_{ST}\,(R_{h^{ST}}\,\xi^F)_{KL} \\ &= \sum_{S\subset T} (\xi^G)_{ST} \Bigg(\sum_{K\subset L} d(K,L)\,(R_{h^{ST}}\,\xi^F)_{KL} \Bigg) \\ &= \sum_{S\subset T} (\xi^G)_{ST} \Bigg(\sum_{X\subset Y} d\big(h^{ST}(X), h^{ST}(Y)\big)\,(\xi^F)_{XY} \Bigg), \qquad (4.8)\end{aligned}$$

where the last identity is by definition of R_h.

At this point we are interested in deriving a lower bound on $d\big(h^{ST}(X), h^{ST}(Y)\big)$, to which end we first need to analyze $\mathrm{cl}_{G\bullet F}(h^{ST}(X))$. Recall that $E(G\bullet F)$ consists of hyperedge $j = (f(j), N(j))$ with $f(j) = (f(j_G), f(j_G))$ and $N(j) = (N(j_G) \times V(F)) \cup (\{f(j_G)\} \times N(j_F))$ for each pair of edges $j_G \in E(G), j_F \in E(F)$. We first claim that for any $S \subset T$ and $X \subset V(F)$,

$$\mathrm{cl}_{G\bullet F}\left(h^{ST}(X)\right) \setminus h^{ST}(X) \;\supseteq\; \Big((\mathrm{cl}_G(S) \setminus S) \cap T \Big) \times \Big(\mathrm{cl}_F(X) \setminus X \Big). \qquad (4.9)$$

To show this, let $L \subseteq V(G \bullet F)$ denote the set on the right side of (4.9). Note that L contains no ordered pairs whose first component is in S or whose second component is in X, and therefore L is disjoint from $h^{ST}(X) = (T \times X) \cup (S \times V(F))$. Consequently, it suffices to show that $\mathrm{cl}_{G\bullet F}\left(h^{ST}(X)\right) \supseteq L$. Consider any message $i = (i_G, i_F)$ belonging to L. As $i_G \in \mathrm{cl}_G(S) \setminus S$, there must exist an edge $j_G \in E(G)$ such that $f(j_G) = i_G$ and $N(j_G) \subseteq S$. Similarly, $i_F \in \mathrm{cl}_F(X) \setminus X$ implies there must exist an edge $j_F \in E(F)$ such that $f(j_F) = i_F$ and $N(j_F) \subseteq X$. Recall from the definition of L that $\{i_G\} = \{f(j_G)\} \subseteq T$. Now letting $K = (N(j_G) \times V(F)) \cup (\{f(j_G)\} \times N(j_F))$, we find that $K \subseteq (S \times V(F)) \cup (T \times X) = h^{ST}(X)$ and that $(i, K) \in E(G \bullet F)$, implying that $i \in \mathrm{cl}_{G\bullet F}\left(h^{ST}(X)\right)$ as desired.

Let $\hat{X} = h^{ST}(X)$ and $\hat{Y} = h^{ST}(Y)$, and recall that $d(\hat{X}, \hat{Y})$ is defined as $\big|\big(\,\mathrm{cl}_{G\bullet F}(\hat{X}) \setminus$

$\hat{X}\big) \cap \hat{Y}\big|$. Using (4.9) and noting that $\hat{Y} \supseteq (T \times Y)$ we find that

$$\Big(\mathrm{cl}_{G \bullet F}(\hat{X}) \setminus \hat{X}\Big) \cap \hat{Y} \;\supseteq\; \Big((\mathrm{cl}_G(S) \setminus S) \cap T\Big) \times \Big((\mathrm{cl}_F(X) \setminus X) \cap Y\Big)$$

and hence

$$d(\hat{X}, \hat{Y}) \geq |(\mathrm{cl}_G(S) \setminus S) \cap T| \,\cdot\, |(\mathrm{cl}_F(X) \setminus X) \cap Y| = d(S,T)\, d(X,Y)\,.$$

Plugging this bound into (4.8) we find that

$$\begin{aligned}\sum_{K \subset L} d(K,L)(\xi^{G \bullet F})_{KL} &\geq \sum_{S \subset T} (\xi^G)_{ST} \sum_{X \subset Y} d(S,T) d(X,Y)(\xi^F)_{XY} \\ &= \Big(\sum_{S \subset T} d(S,T)(\xi^G)_{ST}\Big)\Big(\sum_{X \subset Y} d(X,Y)(\xi^F)_{XY}\Big) = \rho(G)\,\rho(F)\,,\end{aligned}$$

as required. □

Combining Claims 4.1.9 and 4.1.10 concludes the proof of the Theorem 4.1.7. □

Remark. The two sides of (4.9) are in fact equal for any non-degenerate index coding instances G and F, namely under the assumption that every $j_G \in E(G)$ has $f(j_G) \notin N(j_G)$ (otherwise this receiver already knows the required $f(j_G)$ and may be disregarded) and $N(j_G) \neq \emptyset$ (otherwise the public channel must include $f(j_G)$ in plain form and we may disregard this message), and similarly for F. To see this, by definition of $\mathrm{cl}_{G \bullet F}(\cdot)$ and the fact that $h^{ST}(X) = (T \times X) \cup (S \times V(F))$ it suffices to show that every edge $(i, K) \in E(G \bullet F)$ with $K \subseteq h^{ST}(X)$ satisfies $i \in \big(\mathrm{cl}_G(S) \cap T\big) \times \mathrm{cl}_F(X)$. Take $(i, K) \in E(G \bullet F)$ and let $j_G \in E(G)$ and $j_F \in E(F)$ be the edges forming it as per Definition 4.1.1 of the lexicographic product. A prerequisite for $K \subseteq h^{ST}(X)$ is to have $f(j_G) \in T$ as otherwise $\{f(j_G)\} \times N(j_F) \nsubseteq h^{ST}(X)$ (recall that $S \subset T$ and that $N(j_F) \neq \emptyset$). Moreover, as X is strictly contained in $V(F)$ we must have $N(j_G) \subseteq S$ in order to allow $N(j_G) \times V(F) \subseteq h^{ST}(X)$, thus (using the fact that $f(j_G) \notin N(j_G)$ and so $f(j_G) \notin S$) we further require that $N(j_F) \subseteq X$. Altogether we have $N(j_G) \subseteq S$, $N(j_F) \subseteq X$ and $f(j_G) \in T$, hence $(f(j_G), f(j_F)) \in \big(\mathrm{cl}_G(S) \cap T\big) \times \mathrm{cl}_F(X)$ as required.

Open Question 4.1.11. Is β super-multiplicative on lexicographic products?

It is tempting to hope that Theorem 4.1.7 can also be applied to show that β is super-multiplicative. The linear programming lower bounds eventually converge to give β if one includes constraint schemata given by all non-Shannon inequalities. This linear program has infinitely many constraints and can never actually be written down, but if one can show that it is possible to express all of the non-Shannon inequalities as tight constraints, then this would imply that β is super-multiplicative using Theorem 4.1.7. As mentioned earlier, the work of Chan *et al.* [18] even gets us half-way there. They show that all information inequalities have a "balanced" counterpart, which captures part of our notion of tight.

Open Question 4.1.12. Is b sub-multiplicative on lexicographic products?

One way to show b is sub-multiplicative would be to give a way to compose primal solutions to the LP. We already know how to compose a certain type of primal solution, namely, if it corresponds to a code, then our proof that β is sub-multiplicative gives us this composition. Can we determine a representation for all primal solutions and show that the representative object composes under the product? What about a representation for LP $\mathfrak{B}_A$ for an arbitrary constraint matrix A?

4.2 Strong Products

The next product we consider is an extension of the strong product, which we will denote by $\boxtimes$. One motivation for considering the strong product is that it is used to define the Shannon capacity of a graph, a parameter that has many similarities to the parameter β.

Shannon capacity is a coding-related graph parameter. If we take the vertices of the graph to represent symbols of an alphabet and edges to represent confusion between symbols, then the Shannon capacity is the limit, as the message size goes to infinity, of the maximum (normalized) number of messages that can't be confused. For messages of length one, a set of messages that can't be confused is just an independent set in G. For messages of length k, it is an independent set in the k-fold strong product of G. Normalizing for the message length gives us the Shannon capacity of G,

$$c(G) := \lim_{k\to\infty} \sqrt[k]{\alpha(G^{\boxtimes k})}. \tag{4.10}$$

Like β, this parameter is sandwiched between the independence number of G and the fractional clique-cover number, and thus their values coincide for perfect graphs.

Remark 4.2.1. Any parameter $f(G)$ for which $f(G) \geq \alpha(G)$ and $f(G)$ is sub-multiplicative on strong products is an upper bound on the Shannon capacity.

Proof.

$$f(G)^k \geq f(G^{\boxtimes k}) \geq \alpha(G^{\boxtimes k}) \implies f(G) \geq \sqrt[k]{\alpha(G^{\boxtimes k})}.$$

Taking the limit as k goes to infinity gives $f(G) \geq c(G)$. □

The minrank parameter that is equal to λ_1 and an upper bound on β was introduced by Haemers [34, 33]. He shows that minrank upper bounds α and is sub-multiplicative, and thus an upper bound on the Shannon capacity. In [54] Lubetzky and Stav use the sub-multiplicatively of the minrank parameter and its relation to $c(G)$ to show a large separation between scalar linear and non-linear rates. In Section 5.2, we do the same for vector linear codes using the fractional minrank. As a first step, we show here that the fractional minrank is also sub-multiplicative on the strong product.

The relationship between Shannon upper bounds and sub-multiplicatively show that unlike the lexicographic product, β is not sub-multiplicative on the strong product. This follows from $\beta > \alpha$, and an instance in which $c(G) > \beta(G)$. Lubetzky and Stav [54] give such an instance that we describe in Section 5.2.1. Though note that $c(G)$ is not an upper bound on β because for the 5-cycle $c(C_5) = \sqrt{5}$ [53] while $\beta(C_5) = 2.5$.

We now define the strong product for BSIP instances and we show that fractional minrank is sub-additive for this product.

Definition 4.2.2. Given two BSIP instances $G = (V(G), E(G)), F = (V(F), E(F))$, the strong product, denoted $G \boxtimes F$, is the BSIP instance with message set $V = V(G) \times V(F)$ and edge set $E = E(G) \times E(H)$ where $e = (e_G, e_F) \in E$ has $f(e) = (f(e_G), f(e_F))$ and $S(e) = S(e_G) \times S(e_F)$.

When G and F are graphs (BSIP-G) instances, this is equivalent to the strong product of graphs which is given by vertex set $V(G) \times V(F)$ and an edge between distinct vertices (u, v) and $(u'v')$ if for the first coordinate $u = u'$ or $(u, u') \in E(G)$ and also for the second coordinate $v = v'$ or $(v, v') \in E(F)$.

Theorem 4.2.3. *For any BSIP instances G, F, $\operatorname{minrk}_f^{\mathbb{F}}(G \boxtimes F) \leq \operatorname{minrk}_f^{\mathbb{F}}(G) \operatorname{minrk}_f^{\mathbb{F}}(F)$*

Proof. It is well-known that matrix rank is multiplicative under the Kronecker product, thus it is sufficient to prove that given matrices A_G and A_F that fractionally represent G and F over $\mathbb{F}$, $A = A_G \otimes A_F$ fractionally represents $G \boxtimes F$. Let k_G (resp. k_F) be such that A_G (resp. A_F) fractionally represents G (resp. F) over $\mathbb{F}^{k_G}$(resp. $\mathbb{F}^{k_F}$). Then A_G has rows indexed by tuples in $V(G) \times [k_G]$ and columns indexed by tuples in $E(G) \times [k_G]$, and similarly for A_F. Thus, the rows of A are indexed by ordered tuples $V(G) \times [k_G] \times V(F) \times [k_F]$ and columns indexed by ordered tuples in $E(G) \times [k_G] \times E(F) \times [k_F]$. We need to show that for each receiver (e_G, e_F) and message (v_G, v_F) the corresponding $k_G k_F \times k_G k_F$ block matrix is $I_{k_G k_F}$ if $v_G = f(e_G)$ and $v_F = f(e_F)$ and all zeros if $(v_G, v_F) \notin S(e_G) \times S(e_F)$. Let $A_G[v_G, e_G]$ be the $k_G \times k_G$ sub-matrix of A_G restricting to rows and columns indexed by v_G and e_G respectively. Now, notice that the $((e_G, e_F), (v_G, v_F))$ block of A is obtained by $A_G[v_G, e_G] \otimes A_F[v_F, e_F]$.

This gives us exactly what we need: for $A_G[f(e_G), e_G] \otimes A_F[f(e_F), e_F] = I_{k_G} \otimes I_{k_F} = I_{k_G k_F}$. And, if either $v_G \notin S(e_G)$ or $v_F \notin S(e_F)$, we have $A_G[v_G, e_G] \otimes A_F[v_F, e_F] = 0$. □

Remark 4.2.1 and Theorem 4.2.3 together give the following corollary.

Corollary 4.2.4. *For any BSIP-G instance G, $c(G) \leq \operatorname{minrk}_f^{\mathbb{F}}(G)$ for all finite fields $\mathbb{F}$.*

4.3 Sums

It will be useful to consider the sum or disjoint union of BSIP instances as well. Let $G + H$ denote the disjoint union of the BSIP instances G and H, and let $t \cdot G$ denote the disjoint union of t copies of G. We show that β and β^* are additive for the disjoint union. Though this feels intuitive, it is not obvious. We must be careful, as $\beta_k(G)$ is not additive [7], nor is λ_1 [54].

Theorem 4.3.1. *The parameters β and β^* are additive with respect to disjoint unions, that is for any two BSIP instances G, H we have $\beta(G + H) = \beta(G) + \beta(H)$ and $\beta^*(G + H) = \beta^*(G) + \beta^*(H)$.*

Proof. The fact that β^* is additive w.r.t. disjoint unions follows immediately from the results of [7]. Indeed, it was shown there that for any BSIP instance G on n vertices $\beta^*(G) = \log_2 \chi_f(\mathfrak{C}(G))$ where $\mathfrak{C} = \mathfrak{C}(G)$ is an appropriate undirected Cayley graph on the group $\mathbb{Z}_2^n$. Furthermore, it was shown that $\mathfrak{C}(G + H) = \mathfrak{C}(G) \vee \mathfrak{C}(H)$, where $\vee$ denotes the OR-graph-product. It is well-known (see, e.g., [52, 30]) that the fractional chromatic number is multiplicative w.r.t. this product. Combining these statements we deduce that

$$2^{\beta^*(G+H)} = \chi_f(\mathfrak{C}(G + H)) = \chi_f(\mathfrak{C}(G) \vee \mathfrak{C}(H)) = \chi_f(\mathfrak{C}(G))\chi_f(\mathfrak{C}(H)) = 2^{\beta^*(G)+\beta^*(H)}.$$

We shall now use this fact to show that β is additive. The inequality $\beta(G+H) \le \beta(G)+\beta(H)$ follows from concatenating the codes for G and H and it remains to show a matching lower bound.

As observed by [54], a BSIP instance G with n messages that are t bits long has an equivalent formulation as a problem on a graph with tn messages that are 1-bit long; denote this BSIP instance by G_t. Under this notation $\beta_t(G) = \beta_1(G_t)$. Notice that $(G+H)_t = G_t + H_t$ for any t and furthermore that for any s and t, $s \cdot G_t$ and G_{st} both have st copies of each message and receiver in G. For a given receiver in G the copy in $s \cdot G_t$ knows a subset of the messages that the corresponding receiver in G_{st} knows. This implies that $\beta_1(s \cdot G_t) \ge \beta_1(G_{st})$.

Fix $\varepsilon > 0$ and let t be a large enough integer such that $\beta(G+H) \ge \beta_t(G+H)/t - \varepsilon$. Further choose some large s such that $\beta^*(G_t) \ge \beta_1(s \cdot G_t)/s - \varepsilon$ and $\beta^*(H_t) \ge \beta_1(s \cdot H_t)/s - \varepsilon$. We now get

$$\beta(G+H) + \varepsilon \ge \beta_1(G_t + H_t)/t \ge \beta^*(G_t + H_t)/t = \beta^*(G_t)/t + \beta^*(H_t)/t\,,$$

where the last inequality used the additivity of β^*. Since

$$\beta^*(G_t)/t \ge \beta_1(s \cdot G_t)/st - \varepsilon \ge \beta_1(G_{st})/st - \varepsilon \ge \beta(G) - \varepsilon$$

and an analogous statement holds for $\beta^*(H_t)/t$, altogether we have $\beta(G+H) \ge \beta(G) + \beta(H) - 3\varepsilon$. Taking $\varepsilon \to 0$ completes the proof of the lemma. □

Though the linear rate isn't additive, the linear rate over a specific field is additive. This fact was observed for scalar linear rate in [54] but we prove it rigorously here and for vector linear rate as well.

Theorem 4.3.2. *The parameters $\lambda^{\mathbb{F}} = \mathrm{minrk}_f^{\mathbb{F}}$ and $\lambda_1^{\mathbb{F}} = \mathrm{minrk}^F$ are additive with respect to disjoint unions. Moreover, for an BSIP instance G, $t\lambda(G) = \lambda(t \cdot G)$ and $t\lambda_1(G) = \lambda_1(t \cdot G)$.*

Proof. Let G, H be BSIP instances. Consider a matrix A that fractionally represents $G+H$ over $\mathbb{F}^k$. Any entry corresponding to a receiver from G and message from H or vice versa must be all zeros as a receiver in G has no side information about messages in H. Thus, A is of the form $\begin{pmatrix} A_G & 0 \\ 0 & A_H \end{pmatrix}$ where A_G fractionally represents G over $\mathbb{F}^k$ and A_H fractionally represents h over $\mathbb{F}^k$. It is well-known that for such matrices, $\mathsf{rank}(A) = \mathsf{rank}(A_G) + \mathsf{rank}(A_H)$. Thus $\text{minrk}_f^{\mathbb{F}^k}(G+H) \geq \text{minrk}_f^{\mathbb{F}^k}(G) + \text{minrk}_f^{\mathbb{F}^k}(H)$. If $k=1$ and thus A, A_G, A_F represent (not fractionally) $G+H, G$, and H this gives $\text{minrk}^{\mathbb{F}}(G+H) \geq \text{minrk}^{\mathbb{F}}(G) + \text{minrk}^{\mathbb{F}}(H)$. Otherwise, applying this with k minimizing $\text{minrk}_f^{\mathbb{F}^k}(G+H)$ and observing that $\text{minrk}_f^{\mathbb{F}^k}(G) \geq \text{minrk}_f^{\mathbb{F}}(G)$, gives $\text{minrk}_f^{\mathbb{F}}(G+H) \geq \text{minrk}_f^{\mathbb{F}}(G) + \text{minrk}_f^{\mathbb{F}}(H)$. Now, we show that this can be achieved.

Let A_G and A_H be matrices that achieve the optimal fractional minrank over $\mathbb{F}$ for G and H respectively [2] . Let k_G and k_H be their respective block sizes. If $k_G = k_H$ or if we are consider the standard, rather than fractional, minrank, we see that the following matrix has the required rank and represents $G+H$:

$$\begin{pmatrix} A_G & 0 \\ 0 & A_H \end{pmatrix}.$$

If $k_H \neq k_G$, then we set $A'_G = A_G \otimes I_{k_H}$ and $A'_H = A_H \otimes I_{k_G}$. These new matrices represent G and H over $\mathbb{F}^{k_G k_H}$ and have ranks $k_H \mathsf{rank}(A_G) = k_H k_G \,\text{minrk}_f^{\mathbb{F}}(A_G)$ and $k_G k_H \,\text{minrk}_f^{\mathbb{F}}(A_H)$. Now the matrix

$$\begin{pmatrix} A'_G & 0 \\ 0 & A'_H \end{pmatrix},$$

fractionally represents $G+H$ and has the required rank.

We can apply the same argument to a k-fold sum as well. If all the summands are identical then each can achieve the optimal minrank and fractional minrank over the same

[2] If the optimal minrank can only be achieved in the limit as the block size goes to infinity, then we can take a representation with minrank ε-close to optimal and later take limits as $\varepsilon \to 0$

field, implying $t\,\mathrm{minrk}(G) = \mathrm{minrk}(t \cdot G)$ and $t\,\mathrm{minrk}_f(G) = \mathrm{minrk}_f(t \cdot G)$. □

CHAPTER 5

SEPARATING BROADCAST RATE BOUNDS

This chapter shows separations between the broadcast rate and the parameters that bound it. These separations reveal information about the power of different encoding schemes and the quality of lower and upper bounds.

5.1 Insufficiency of the Shannon Bound

In Chapter 3 we give many structured instances of BSIPs for which $b(G) = \beta(G)$ including cycles, complements of cycles, some cyclic Cayley graphs, and instances from representable matroids. Later in this Chapter, in Sections 5.2 and 5.3, we will see instances when $b = \beta$ while $\alpha \ll \beta$ or $\beta \ll \lambda$. It is natural to ask if b is always equal or approximately equal to β. In [22], the authors show that an entropy based LP bound on the network coding rate — similar to b — is not always equal to the optimal coding rate. Here we use a similar approach to show that β can be strictly smaller than b.

Theorem 5.1.1. *There exists a BSIP instance G for which $b(G) < \beta(G)$. In particular, $b(G) = 4$ and $\beta(G) \geq \frac{45}{11}$.*

The proof relies on a BSIP instance associated to the Vámos matroid, which is the smallest non-representable matroid.

Definition 5.1.2. The Vámos matroid is an eight-element rank-four matroid whose ground set is $V = \{a, b, c, d, w, x, y, z\}$ and whose dependent sets are all the subsets of cardinality at least five as well as the four-element sets $\{b, c, x, y\}$, $\{a, c, w, y\}$, $\{a, b, w, x\}$, $\{c, d, y, z\}$, and $\{b, d, x, z\}$. The eight elements can be thought of as the eight vertices of a cube and the dependent four element sets can be viewed as five of the coplanar sets of the cube as depicted in Figure 5.1.

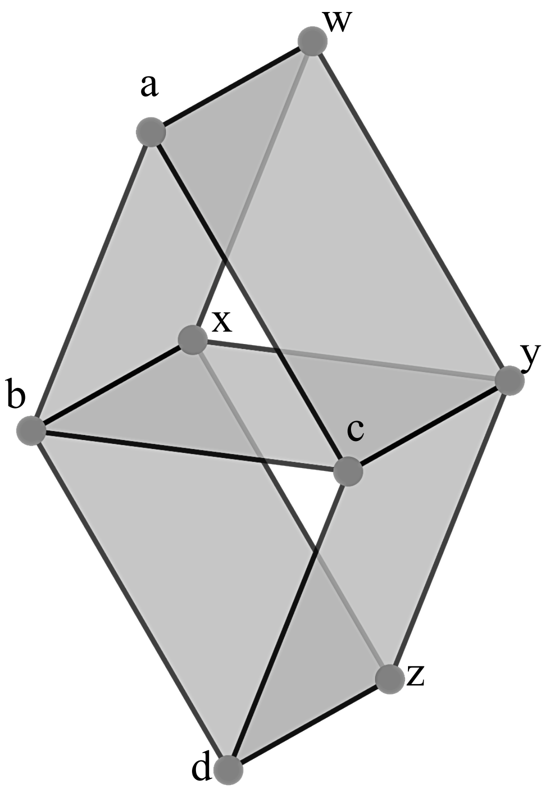


Figure 5.1: A representation of the Vámos matroid.

The vertices correspond to the eight elements. The planes in gray show the 4-element dependent sets.

Proof. Let $G_{\mathcal{V}}$ be the BSIP instance associated to the Vámos matroid according to Definition 3.1.2. Proposition 3.1.3 gives that $b(G_{\mathcal{V}}) = |E| - r(E) = 8 - 4 = 4$, as needed.

We will next show that $\beta \geq \frac{45}{11}$ via LP $\mathfrak{B}_{ZY}$. Recall that the Zhang-Yeung inequality is given by:

$$\begin{aligned} 3d_{BD}+&3d_{CD} + 3d_{BC} + d_{AB} + d_{AC} \\ &- 2d_B - 2d_C - d_{AD} - d_D - d_{ABC} - 4d_{BCD} \geq 0 \end{aligned}$$

We use the row of the Zhang-Yeung constraint schema corresponding to sets $A = \{a, w\}$, $B = \{b, x\}$, $C = \{c, y\}$, and $D = \{d, z\}$. Observe that the rank function of the Vamos matroid does not satisfy this inequality since the sets with positive coefficients are each dependent sets of size four, giving a rank of 33, yet the sets with negative coefficients include six sets with rank 4 and five with rank 2, giving a total rank of 34. This implies by Theorem 3.1.6 that $\beta > 4$. But, we use specific constraints of the LP to get a tighter bound.

Summing and rearranging the following inequalities of LP $\mathfrak{B}_{ZY}$ will produce the desired result.

$$\begin{aligned}
&11z_\emptyset + 33 \geq 6z_V + 2z_{bx} + 2z_{cy} + z_{dz} && \text{(Yeung-Zhang, decode)}\\
&2 \times [z_{bx} + 2 \geq z_V] && \text{(decode)}\\
&2 \times [z_{cy} + 2 \geq z_V] && \text{(decode)}\\
&z_{dz} + 2 \geq z_V && \text{(decode)}\\
&11 \times [z_V \geq 8] && \text{(initialize)}\,.
\end{aligned}$$

To see the validity of the decoding constraints let $\mathcal{D}$ be the set of four-element dependent sets. Note that any subset of size three of $D \in \mathcal{D}$ decodes D, and any subset of four elements of V that is not in $\mathcal{D}$ decodes all of V. Altogether, $\beta \geq z_\emptyset \geq \frac{45}{11}$ while $b = 4$, completing the proof. □

Open Question 5.1.3. Does there a family of BSIP instances on n messages such that $\Omega(n^\varepsilon)b \leq \beta$?

If we take lexicographic powers of $G_\mathcal{V}$ then using the super-multiplicativity of b_{ZY} and b we get $\beta \geq \frac{45}{11}^k$, and $b \geq 4^k$. To complete a polynomial separation we need an upper bound on b. For $G_\mathcal{V}$ this upper bound is achieved using the connection to matroids. If one could find a connection between the product instance and a polymatroid then maybe this would yield an upper bound on b for the product. Such a result would be a nice contribution to the field of information theory for its implications to the relationship between Γ_n and $\overline{\Gamma}^*_n$.

5.2 Strong Insufficiency of Vector Linear Coding

This section is devoted to showing a separation between λ and β. We find polynomial separations for two infinite families of graphs. This implies a polynomial separation between vector linear and non-linear for general network coding as well. It improves upon the best previously known separation of 11/10 by Dougherty *et al.* [24], and disproves the conjecture of Medard *et al.* [58] that vector linear coding is sufficient for general network coding even in an approximate sense.

5.2.1 Separation via Fractional Minrank

In [54], Lubetzky and Stav show that for any $\varepsilon > 0$ and any suciently large n, there is a BSIP-G instance G on n vertices (messages) so that $\lambda_1(G) \geq \frac{\sqrt{n}}{n^{\varepsilon}}\beta(G)$. They consider a graph $H = G + \overline{G}$ and give an efficient non-linear code for H via the concatenation of linear codes over two different fields. They then show that $\lambda_1(H) = \text{minrk}(H)$ is large by showing that Shannon capacity (Equation (4.10)), a lower bound on minrank, is large.

We follow the analogous approach using fractional minrank, thus getting a separation between λ and β.

Theorem 5.2.1. *For any $\varepsilon > 0$ and sufficiently large n, there exists a BSIP-G instance G with n messages such that $\lambda(G) \geq \Omega\left(\frac{\sqrt{n}}{n^{\varepsilon}}\right)\beta(G)$*

Proof of Theorem 5.2.1. We begin with a graph construction from [54]. Let $\varepsilon > 0$, and let k denote a (large) integer satisfying

$$3^l < 2^k < (1+\varepsilon)3^l \text{ where } l = \lfloor k \log_3 2 \rfloor.$$

Let H be the BSIP-G instance defined by the graph on $n = \binom{r}{s}$ vertices each represented by an s-element subset of $[r]$. Two vertices are adjacent iff their corresponding sets have an intersection whose cardinality is congruent to 1 modulo 2^k. Let $\mathbb{F}_2 = GF(2), \mathbb{F}_3 = GF(3)$. The proof of Proposition 2.2 in [54] shows that $\text{minrk}_f^{\mathbb{F}_2}(H) \leq \text{minrk}^{\mathbb{F}_2}(H) \leq n^{\varepsilon}$ and $\text{minrk}_f^{\mathbb{F}_3}(\overline{H}) \leq \text{minrk}^{\mathbb{F}_3}(\overline{H}) \leq n^{\varepsilon}$. We will consider the graph $K = H + \overline{H}$. Applying Theorem 4.3.1, we have $\beta(K) = \beta(H) + \beta(\overline{H}) \leq 2n^{\varepsilon}$. It remains to show that $\lambda(K) \geq \Omega(\sqrt{n})$.

To show that the best scalar linear code for K is much larger, Lubetzky and Stav [54] show that $c(K)$ is large and use the fact that the Shannon capacity is a lower bound on the minrank. We can do the same for fractional minrank because Corollary 4.2.4 gives that $c(G) \leq \text{minrk}_f^{\mathbb{F}}(G)$ for all $\mathbb{F}$, and thus $c(G) \leq \text{minrk}_f(G) = \lambda(G)$.

We show that $c(K) \geq \sqrt{2n}$ by giving an independent set of size $2n$ in $K^{\boxtimes 2}$. Observe $K^{\boxtimes 2} = 2 \cdot H \boxtimes \overline{H} + \overline{H} \boxtimes \overline{H} + H \boxtimes H$. In [54], they observe that for any graph G the set $\{(u,u) | u \in V(G)\}$ is an independent set in $G \boxtimes \overline{G}$ because for $u \neq v$ the edge (u,v) is present in exactly one of G and $\overline{G}$, implying (u,u) is not adjacent to (v,v) in the product. The graph $K^{\boxtimes 2}$ contains two disjoint copies of $H \boxtimes \overline{H}$ and thus an independent set of size $2n$. □

5.2.2 Separation via LPs $\mathfrak{B}_{\mathcal{F}}$ and $\mathfrak{B}_{\mathcal{N}}$

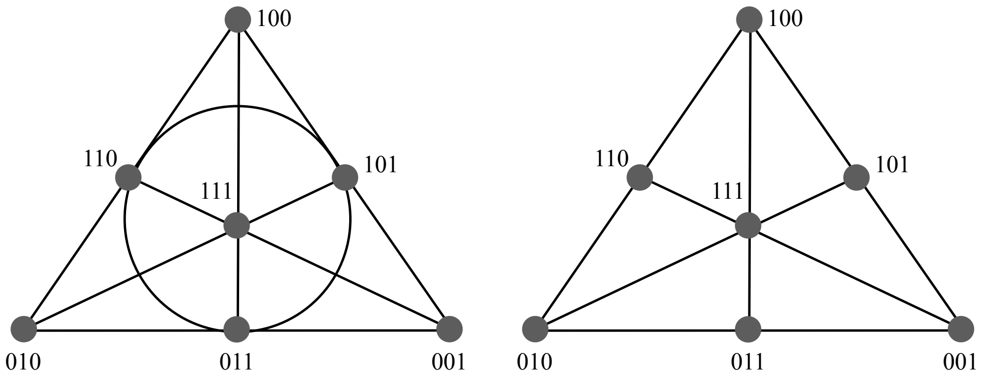
We show a polynomial gap between vector linear and non-linear coding use a different approach. The proof shows the power of the linear programming bounds and their supermultiplicativity under lexicographic products. Further, unlike the construction in Section 5.2.1, in this construction the best non-linear code we know is not just the concatenation of two linear codes over different fields, but rather, involves a recursive, intricate combination of linear codes.

For this construction we consider BSIP instances associated to the Fano and non-Fano matroids.

Definition 5.2.2. The *Fano matroid*, denoted $\mathcal{F}$, and the *non-Fano matroid*, denoted $\mathcal{N}$, are 7 element, rank 3 matroids. The seven columns of the matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

constitute a linear representation of the Fano matroid when $\text{char}(\mathbb{F}) = 2$ and one for the non-Fano matroid when $\text{char}(\mathbb{F}) \neq 2$. We will use $\mathcal{U} = \{100, 010, 001, 110, 101, 011, 111\}$ to index the elements of the two matroids. They are often shown visually via the images in Figure 5.2.



(a) Fano Matroid Representation (b) non-Fano Matroid Representation

Figure 5.2: A representation of the Fano and non-Fano matroids.

The vertices correspond to the seven elements. The lines show the dependent sets of size 3. The Fano matroid has one more dependent set of size 3 than the non-Fano matroid represented by the circle.

We consider the BSIP instances associated to the Fano and non-Fano matroids, denoted $G_{\mathcal{F}}$ and $G_{\mathcal{N}}$ respectively, as defined in Definition 3.1.2. We show a polynomial gap for the instance $(G_{\mathcal{F}} \bullet G_{\mathcal{N}})^{\bullet k}$. We will first show that the linear rate over a field of even characteristic is strictly better than the linear rate over a field of odd characteristic for $G_{\mathcal{F}}$, and that the reverse relation holds for the non-Fano matroid. We use this to show that the there is a gap between the linear and non-linear coding rates of $G_{\mathcal{F}} \bullet G_{\mathcal{N}}$, and then amplify that gap via the k-fold lexicographic product.

Theorem 5.2.3. *Let $G = G_{\mathcal{F}} \bullet G_{\mathcal{N}}$. For all $k \in \mathbb{N}$, $\beta(G^{\bullet k}) = 16^k$ whereas $\lambda(G^{\bullet k}) \geq (16.12)^k$. Thus showing that $\lambda(G^{\bullet k}) \geq \Omega(n^{0.002})\beta(G^{\bullet k})$ where n is the number of messages in instance $G^{\bullet k}$.*

Proof. The fact that $b(G^{\bullet k}) = 16^k$ is a direct application of Theorems established in Sections 3.1 and 4.1. Theorem 3.1.5 and the representability of both $\mathcal{F}$ and $\mathcal{N}$ gives that $\beta(G_{\mathcal{F}}) = \beta(G_{\mathcal{N}}) = |E| - r(E) = 4$. The sub-multiplicativity of β under the lexicographic product (Theorem 4.1.2) then implies that $G = G_{\mathcal{F}} \bullet G_{\mathcal{N}}$ satisfies $\beta(G^{\bullet k}) \leq (4 \cdot 4)^k = 16^k$. A lower bound of the form $\beta(G^{\bullet k}) \geq 16^k$ is a consequence of Proposition 3.1.3 which implied that $b(G_{\mathcal{F}}) = b(G_{\mathcal{N}}) = 4$, from which it follows by the super-multiplicativity of b under lexicographic products (Theorem 4.1.7) that $16^k \leq b(G^{\bullet k}) \leq \beta(G^{\bullet k})$. Combining these upper and lower bounds, we find that $\beta(G^{\bullet k}) = 16^k$.

It is worth noting, incidentally, that although each of $G_{\mathcal{F}}$, $G_{\mathcal{N}}$ individually has a linear solution over the appropriate field, the index code for $G = G_{\mathcal{F}} \bullet G_{\mathcal{N}}$ implied by the proof of Theorem 4.1.2 — which concatenates these two linear codes together by composing them with an arbitrary one-to-one mapping from a mod-2 vector space to a mod-p vector space (p odd) — is highly non-linear, and not merely a side-by-side application of two linear codes.

To establish the lower bound on $\lambda(G^{\bullet k})$, we distinguish two cases, $\mathrm{char}(\mathbb{F}) = 2$ and $\mathrm{char}(\mathbb{F}) \neq 2$, and in both cases we prove $\lambda^{\mathbb{F}}(G^{\bullet k}) \geq (16 + \varepsilon)^k$ using the LPs $\mathfrak{B}_{\mathcal{F}}$ and $\mathfrak{B}_{\mathcal{N}}$ respectively.

Theorems A.0.4 and A.0.5 show that there is a row of the Fano (resp. non-Fano) constraint matrices that are violated for the rank vector of $\mathcal{N}$ (resp. $\mathcal{F}$) matroid. This implies by Theorem 3.1.6 that $b_{\mathcal{F}}$ and $b_{\mathcal{N}}$ are strictly greater than four. Now we will show directly that $b_{\mathcal{F}}(G_{\mathcal{N}}) \geq 4.043$ and $b_{\mathcal{N}}(G_{\mathcal{F}}) \geq 4.03$.

Let $\vec{\alpha}_{\mathcal{F}}$ and $\vec{\alpha}_{\mathcal{N}}$ denote the inequalities that describe the Fano and non-Fano constraint schema.

First, observe that the rank function of $\mathcal{N}$ does indeed violate the inequality $\vec{\alpha}_{\mathcal{F}}$ setting $A = 100, B = 010, C = 001, D = 110, E = 101, F = 011, G = 111, H = \emptyset$. We write the values of $d = \vec{\mathbf{r}}(\mathcal{N})$ in square brackets next to each term. Notice that the dependent sets of size three appear with a positive coefficient and the independent sets of size three appear with a negative coefficient.

$$\begin{aligned}
&2d_{AH}[1] + 2d_{BH}[1] + 3d_{CH}[1] + 11d_{GH}[1] + 3d_{ABH}[2] + 2d_{ACH}[2] + 2d_{BCH}[2] \\
&\qquad + d_{ABDH}[2] + d_{ACEH}[2] + d_{AFGH}[2] + d_{BCFH}[2] + d_{BEGH}[2] + d_{CDGH}[2] + d_{ABCGH}[3] \\
&\qquad + d_{ABCDEGH}[3] + d_{ABCDFGH}[3] + d_{ABCEFGH}[3] + 3d_{ABCDEFH}[3] \\
&\qquad - 15d_{H}[0] - d_{AGH}[2] - d_{BGH}[2] - d_{CGH}[2] - 4d_{ABCH}[3] - 3d_{ABGH}[3] - 3d_{ACGH}[3] \\
&\qquad - 3d_{BCGH}[3] - d_{DEFH}[3] - 6d_{ABCDEFGH}[3] \geq 0
\end{aligned}$$

Summing we get $1 \cdot (2 + 2 + 3 + 11) + 2 \cdot (3 + 2 + 2 + 6 \cdot 1) + 3 \cdot (4 \cdot 1 + 3) - 0 \cdot 15 - 2 \cdot (1 + 1 + 1) - 3 \cdot (4 + 3 + 3 + 3 + 1 + 6) = -1$. Note that if we plugged in the rank vector of matroid $\mathcal{F}$ then the only thing that would change would be the term $-d_{DEFH}$. It would have a value of 2 rather than 3, and would give us a sum of zero.

Now, we use a sequence of constraints of LP $\mathfrak{B}_{\mathcal{F}}$ to get a bound on $b_{\mathcal{F}}(G_{\mathcal{N}})$.

$$\begin{aligned}
38z_{\emptyset} + 65 &\geq 15z_{\emptyset} + 20z_V + z_{100,111} + z_{010,111} + z_{001,111} && \text{(decode, } \alpha_{\mathcal{F}}\text{)} \\
z_{100,111} + 1 &\geq z_V && \text{(decode)} \\
z_{010,111} + 1 &\geq z_V && \text{(decode)} \\
z_{010,111} + 1 &\geq z_V && \text{(decode)} \\
23z_V &\geq 23 \cdot 7 && \text{(initialize)}\,.
\end{aligned}$$

Summing, we get that $23z_\emptyset \geq 23 \cdot 7 - 68$, and $b_{\mathcal{F}}(G_{\mathcal{N}}) \geq 4.043$.

Now we do the corresponding thing to show $b_{\mathcal{N}}(G_{\mathcal{F}}) > 4$ via LP $\mathfrak{B}_{\mathcal{N}}$. We use the same mapping from sets to elements of the matroid as before. To start, we again verify that the rank function of $G_{\mathcal{F}}$ violates $\vec{\alpha}_{\mathcal{N}}$. We write the rank of each term next to it in the inequality in square brackets.

$$\begin{aligned}
&3d_{AH}[1] + 3d_{BH}[1] + 9d_{CH}[1] + 6d_{GH}[1] + 6d_{ABH}[2] + 3d_{ABDH}[2] \\
&\quad +3d_{ACEH}[2] + 3d_{BCFH}[2] + d_{DEFH}[2] + 3d_{ABCGH}[3] \\
&\quad +4d_{ABCDEGH}[3] + 4d_{ABCDFGH}[3] + 4d_{ABCEFGH}[3] \\
&\quad -13d_{H}[0] - 12d_{ABCH}[3] - 3d_{ABGH}[3] - 3d_{ACGH}[3] - 3d_{BCGH}[3] \\
&\quad -12d_{ABCDEFGH}[3] \geq 0
\end{aligned}$$

Summing we get $1\cdot(3+3+9+6)+2\cdot(6+3+3+3+1)+3\cdot(3+4+4+4)-0\cdot(13)+3\cdot(12+3\cdot3+12) = -1$.

Now, we use a sequence of constraints of LP $\mathfrak{B}_{\mathcal{N}}$ to get a bound on $b_{\mathcal{N}}(G_{\mathcal{F}})$.

$$\begin{aligned}
52z_\emptyset + 98 &\geq 13z_\emptyset + 33z_V && (\text{decode, } \vec{\alpha}_{\mathcal{N}}) \\
33z_V &\geq 33 \cdot 7 && (\text{initialize})\,.
\end{aligned}$$

Summing, we get that $39z_\emptyset \geq 33 \cdot 7 - 98$, and $b_{\mathcal{N}}(G_{\mathcal{F}}) \geq 4.030$.

Now, using that the lexicographic product of both LPs is super-multiplicative we know $b_{\mathcal{F}}(G^{\bullet k}) \geq (4 \cdot 4.043)^k = 16.17^k$ and $b_{\mathcal{N}}(G^{\bullet k}) \geq (4 \cdot 4.030)^k = 16.12^k$, implying together that $\lambda(G^{\bullet k}) \geq 16.12^k$.

The graph $G^{\bullet k}$ has 49^k vertices, so writing this in terms of n, we have that $\beta(G^{\bullet k}) = 16^k = n^{0.712}$, and that $\lambda(G^{\bullet k}) \geq 16.12^k = n^{0.714}$, giving a multiplicative gap of $n^{0.002}$. □

Open Question 5.2.4. Can one construct a family of BSIP instances in which the scalar linear and vector linear rates are separated by a gap polynomial in the instance size?

It is generally assumed that vector linear coding is much more powerful than scalar linear, but evidence for this remains to be found. The best multiplicative factor known between vector and scalar linear coding is just a small constant: 1.2, achieved by the 5-cycle. It is possible that one could show a separation similar to the technique used here — find an inequality that is satisfied for scalar linear codes and not vector linear codes, use it to find an instance with a small separation and then amplify it using lexicographic products. Alternatively, one could take advantage of the relationship between linear coding and the minrank parameters and amplification using the strong product operation.

5.3 Separation between α and β

In this section we show a class of graphs with a polynomial-sized gap between the trivial lower bound α and the broadcast rate β. This is not only the first large separation between α and β, but it gives the first instance of graphs for which we can show $\alpha < \beta$.

Theorem 5.3.1. *For all $k \in \mathbb{N}$ we have $\beta(C_5^{\bullet k}) = \left(\frac{5}{2}\right)^k$ while $\alpha(C_5^{\bullet k}) = 2^k$, implying that $\beta(C_5^{\bullet k}) \leq \Omega(n^\delta)\alpha(C_5^{\bullet k})$, where $\delta = 1 - 2\log_5(2) \approx 0.139$ and n is the number of messages in $C_5^{\bullet k}$.*

Proof. Theorem 3.2.1 gives $b(C_5) = \beta(C_5) = \frac{5}{2}$. Furthermore, $\alpha(C_5) = 2$, giving us a small separation between α and β.

Now, we can amplify this gap using the super-multiplicativity of lexicographic products (Theorem 4.1.7) just as we did for the gap in Section 5.2.2. In particular, we will transform this small gap on C_5 to a polynomial gap on $C_5^{\bullet k}$.

Applying Theorem 4.1.7 we deduce that for any integer $k \geq 1$ the k-th lexicographic power of C_5 satisfies $\beta(C_5^k) \geq b(C_5^k) \geq \left(\frac{5}{2}\right)^k$. The sub-multiplicativity of β (Theorem 4.1.2) gives $\beta(C_5^k) \leq \beta(C_5)^k = \left(\frac{5}{2}\right)^k$ Furthermore, $\alpha(C_5) = 2$ and it is well known that the independence number is multiplicative on lexicographic products and so $\alpha(C_5^k) = 2^k$. Altogether, C_5^k is a graph on $n = 5^k$ vertices with $\alpha = n^{\log_5(2)}$ and $\beta = n^{1-\log_5(2)}$, implying our result. □

Theorem 5.3.1 also has interesting consequences for general network coding. We can map the BSIP instances that give the large separation between α and β to network coding instances that give a large separation between the network coding rate and iMeagerness, the strongest known cut bound for directed multiple unicast problems. Harvey and Kleinberg [37] show that cut bounds *meagerness* and *vertex sparsity* can be $\Omega(n)$ larger than the coding rate. This motivates them to define iMeagerness, a stronger cut bound. They give an example for which the cut is an $\Omega(\log n)$ factor larger than the coding rate. See Section 1.1.3 for an overview of work on network coding cut bounds. We can apply Theorem 5.3.1 to provide an n-vertex multiple unicast problem for which the cut is an $\Omega(n^{0.138})$ factor larger than the network coding rate.

To this end, we show that there is a network coding instance N with $n + m + 2$ nodes corresponding to every BSIP instance G with n messages and m receivers such that the coding rate of N is equal to $\frac{1}{\beta(G)}$ and the iMeagerness of N equal to $\frac{1}{\alpha(G)}$.

We give a mapping from a BSIP instance to a network coding instance. This formalizes the sketch given in the introduction.

Definition 5.3.2. Given a BSIP instance $G = (V, E)$, the corresponding network coding instance N is given by directed graph with vertex set $V' = \{u_i | i \in V\} \cup \{u'_j | i \in E\} \cup \{w, w'\}$. The edge set E' contains infinite capacity directed edges $\{(u_i, u'_j) | i \in N(j)\} \cup \{(u_i, w) | i \in V\} \cup \{(w', u'_i) | i \in V\}$. E' additionally contains the so-called bottleneck edge (w, w') that has capacity one. There are $|V|$ commodities, and for each commodity a single source

$\mathsf{Src}(i) = \{u_i\}$ and sinks $\mathsf{Snk}(i) = \{u'_j | f(j) = i\}$. See Figure 1.1 for an illustration of the network coding instance.

Observe that if G is a BSIP-G instance then there is only one sink for each commodity and thus corresponds to a multiple unicast network coding instance.

The following theorem establishes that the coding rates of the two instances correspond.

Claim 5.3.3. *Given a BSIP instance $G = (V, E)$, the corresponding network coding instance N has coding rate $\frac{1}{\beta(G)}$.*

Proof. First we show that the coding rate of N is at least $\frac{1}{\beta(G)}$. Let $\varepsilon > 0$ and $\mathcal{E} : \Sigma \to \Sigma_P$ be a broadcasting solution such that $\frac{\log|\Sigma_P|}{\log|\Sigma|} \leq \beta(G) + \varepsilon$. Now, our network coding solution will use source alphabet Σ. Edges $(w, w'), \{(w', u'_i) | i \in V\}$ will have alphabet Σ_P, and the remainder of the edges will have alphabet Σ. The coding function on edges with alphabet Σ_P is $\mathcal{E}$ and the coding function on edges $(u_i, -)$ pulls out the i^{th} source message from the message vector. It is not hard to see that all edge functions can be computed. The sinks receive their desired message because $\mathcal{E}$ is a valid coding function for BSIP. Taking the limit as ε goes to zero, and letting b be such that $\log_b |\Sigma_P| = 1$ gives $\frac{1}{\log|\Sigma|} \leq \beta(G)$. Noticing that we chose b so that $\log_b |\Sigma_e| \leq c(e)$ for all Σ_e giving $\log|\Sigma|$ as a lower bound on the coding rate of this solution.

Now we show the coding rate of N is at most $\frac{1}{\beta(G)}$. Let Σ be an any source alphabet. Without loss of generality we can assume that the coding functions on edges $(u_i, -)$ pull out the i^{th} source message from the message vector because u_i has no incoming edges, is only the source of commodity i, and all of these edges have infinite capacity. For similar reasons, the coding functions of edges $(w, -)$ are equal to the coding function of (w, w') without loss of generality. What remains is to find a coding function on edge (w, w') that exactly captures the BSIP problem. □

Now we show a correspondence between α and iMeagerness.

Definition 5.3.4. Let N be a network coding instance given by a directed acyclic graph $G = (V, E)$, commodities I, and source and sink sets $\mathsf{Src}(i)$ and $\mathsf{Snk}(i)$ for $i \in I$. The *informational Meagerness*, or *iMeagerness* of a subset of edges $A \subseteq E$ is:

$$i\mathcal{M}(A) := \min_{P : A \text{ informationally isolates } P} \frac{\sum_{e \in E'} c(e)}{|P|} \tag{5.1}$$

where A informationally isolates P if A together with the source messages indexed by $[k] \setminus P$ determine the source messages indexed by P for any network coding solution with a strictly positive rate.

The iMeagerness of a network is the minimum iMeagerness of any subset.

$$i\mathcal{M}(N) := \min_{A \subseteq E} i\mathcal{M}(A). \tag{5.2}$$

Claim 5.3.5. *For a BSIP-G instance G, the corresponding network coding instance N has $i\mathcal{M}(N) = \frac{1}{\alpha(G)}$.*

Note that we prove this for BSIP-G rather than the more general BSIP. This is sufficient because $C_5^{\boxtimes k}$ is a BSIP-G instance. It is significantly simpler because α corresponds to the independent set number and not the more complicated expanding sequence number.

Proof. To see that $i\mathcal{M}(N) \leq \frac{1}{\alpha(G)}$ we let A be the bottleneck edge and let P be a maximal independent set in G. It is known that if every path between the $\{\mathsf{Src}(i) | i \in P\}$ and $\{\mathsf{Snk}(i) | i \in P\}$ intersects A then A informationally isolates P (Lemma 7 in [36]). This holds for our choice of A, P because the independence of P in G implies no direct edges between sets $\{u_i | i \in P\}$ and $\{u'_i | i \in P\}$, and all paths go through the bottleneck edge.

To see that $i\mathcal{M}(N) \leq \frac{1}{\alpha(G)}$ we first observe that the set A that minimizes the expression of $i\mathcal{M}(N)$ must be the singleton set containing the bottleneck edge. All other edges have infinite capacity and cannot achieve the minimum. Now, we show that P must be an independent set. Suppose not, then there is some $i, j \in P$ such that $(i, j) \in E(G)$ and thus there is are edges (u_i, u'_j) and (u_j, u'_i) in N. Now we show that A does not informationally isolate P. We give a feasible code in which the source messages not in P and the coding function on A cannot determine the source messages of $i, j \in P$. Let $x = (x_1, \ldots, x_n)$ be the source message tuple. Consider the code that sends $x_i + x_j$ and $x_k, \forall k \neq i, j$ along the bottleneck edge and all other edges (u, v) send a message containing all the information on in-edges of u or sources at u. This code is feasible because all the sinks receive the bottleneck edge message and sink i receives source message j and vice versa as $(i, j) \in E$. But, the messages along edge A and source messages $\overline{P}$ do not determine x_i or x_j. □

5.4 Separating the broadcast rate from clique-cover bound

In this section we show a strong form of separation between β and its upper bound $\overline{\chi}_f$. Not only can we have a family of graphs where $\beta = O(1)$ while $\overline{\chi}_f$ is unbounded, but one can construct such a family where $\overline{\chi}_f$ grows polynomially fast with n.

Theorem 5.4.1. *There exists an explicit family of graphs G on n vertices such that $\beta(G) = \mathrm{minrk}(G) = 3$ whereas the coding schemes based on clique-covers cost at least $\overline{\chi}_f(G) = \Theta(n^{1/4})$ bits.*

The following family of graphs (up to a small modification) was introduced by Erdős and Rényi in [29]. Due to its close connection to the (Sylvester-)Hadamard matrices when the chosen field has characteristic 2 we refer to it as the *projective-Hadamard* graph $H(\mathbb{F}_q)$:

1. Vertices are the non-self-orthogonal vectors in the 2-dimensional projective space over $\mathbb{F}_q$. [1]

[1] vectors in the 2-dimensional projective space over $\mathbb{F}_q$ are equivalence classes of the set $\mathbb{F}_q^3 - \{(0, 0, 0)\}$

2. Two vertices are adjacent iff their corresponding vectors are non-orthogonal.

Observation 5.4.2. *The number of vertices in $H(\mathbb{F}_q)$ is at least $\frac{q^2(q-2)}{(q-1)}$.*

Proof. We need to count the number of non-self orthogonal vectors 2-dimensional projective space over $\mathbb{F}_q$. There are at least $q^2(q-2)$ non-self orthogonal vectors in F_q: pick the first two elements of the vector to be any elements of $\mathbb{F}_q$, then there are at least $q-2$ elements of $\mathbb{F}_q$ to use for the last element so that the vector is not self-orthogonal. At most $q-1$ vectors in $\mathbb{F}_q^3$ map to a single element of 2-dimensional projective space over $\mathbb{F}_q$, completing the proof. □

Proof of Theorem 5.4.1. Let q be prime. We claim that the BSIP-G instance described by the projective-Hadamard graph $H(\mathbb{F}_q)$ on n vertices satisfies $\beta = 3$ while $\overline{\chi}_f = \Theta(n^{1/4})$. The latter is a well-known fact which appears for instance in [6, 59]. Showing that $\overline{\chi}_f \geq (1 - o(1))n^{1/4}$ is straightforward and we include an argument establishing this for completeness.

The fact that $\beta \geq 3$ follows from the fact that the standard basis vectors form an independent set of size 3. A matching upper bound will follow from the $\mathrm{minrk}^{\mathbb{F}}$ parameter (see Definition 1.2.7). Let $\mathbb{F}$ be some finite field and let $\ell = \mathrm{minrk}^{\mathbb{F}}(G)$ be the length of the optimal linear encoding over $\mathbb{F}$ for BSIP-G instance G and messages taking values in $\mathbb{F}$. Broadcasting $\ell\lceil\log_2|\mathbb{F}|\rceil$ bits allows each receiver to recover his required message in $\mathbb{F}$ and so $\beta \leq \ell$. It thus follows that $\lceil\beta(G)\rceil \leq \mathrm{minrk}^{\mathbb{F}}(G)$ for any graph G and finite field $\mathbb{F}$.

Here, dealing with the projective-Hadamard graph H, let B be the Gram matrix over $\mathbb{F}_q$ of the vectors corresponding to the vertices of H. By definition the diagonal entries are nonzero and whenever two vertices u, v are nonadjacent we have $B_{uv} = 0$. In particular B is a representation for H over $\mathbb{F}_q$ which clearly has rank 3 as the standard basis vectors span its entire row space. Altogether we deduce that $\beta(H) = 3$.

modulo the equivalence relation $x \sim kx$, for all $k \in \mathbb{F}_q$, $x \in \mathbb{F}_q^3$.

The fractional clique-cover number is at least as big as the number of vertices divided by the size of the largest clique. Thus, to show that $\overline{\chi}_f(H) \geq (1-o(1))n^{1/4}$ it is sufficient to show that the clique-number of H is at most $(1-o(1))q^{3/2} \leq (1+o(1))n^{3/4}$.

Consider the following multi-graph G which consists of the entire projective space:

1. Vertices are all vectors of the 2-dimensional projective space over $\mathbb{F}_q$.
2. Two (possibly equal) vertices are adjacent iff their corresponding vectors are orthogonal.

Clearly, G contains the complement of the Hadamard graph $H(\mathbb{F}_q)$ as an induced subgraph and it suffices to show that $\alpha(G) \leq (1-o(1))q^{3/2}$.

It is well-known (and easy) that G has $N = q^2+q+1$ vertices and that every vertex of G is adjacent to precisely $q+1$ others. Further observe that for any $u, v \in V(G)$ precisely one vertex of G belongs to $\{u,v\}^{\perp}$ (as u, v are linearly independent vectors). In other words, the codegree of any two vertices in G is 1. We conclude that G is a strongly-regular graph (see e.g. [31] for more details on this special class of graphs) with codegree parameters $\mu = \nu = 1$ (where μ is the codegree of adjacent pairs and ν is the codegree of non-adjacent ones). There are thus precisely 2 nontrivial eigenvalues of G given by $\frac{1}{2}((\mu-\nu) \pm \sqrt{(\mu-\nu)^2+4(q+1-\nu)}) = \pm\sqrt{q}$, and in particular the smallest eigenvalue is $\lambda_N = -\sqrt{q}$. Hoffman's eigenvalue bound (stating that $\alpha \leq \frac{-m\lambda_m}{\lambda_1-\lambda_m}$ for any regular m-vertex graph with largest and smallest eigenvalues λ_1, λ_m resp., see e.g. [31]) now shows

$$\alpha(G) \leq \frac{-N\lambda_N}{(q+1)+\lambda_N} = \frac{(q^2+q+1)\sqrt{q}}{q+\sqrt{q}+1} = q^{3/2} - q + \sqrt{q}\,,$$

as required. □

5.5 Triangle-free Graphs

In addition to demonstrating a large gap between $\overline{\chi}_f$ and β on the projective-Hadamard graphs, we show that even in the extreme cases where G is a triangle-free graph on n vertices, in which case $\overline{\chi}_f(G) \geq n/2$, one can construct coding schemes that significantly outperform $\overline{\chi}_f$.

For triangle-free graphs, where the upper bound $\overline{\chi}_f$ on β is at least $n/2$. The first question in this respect is whether possibly $\beta = \overline{\chi}_f$ in this regime, i.e. for graphs with $\overline{\chi}_f = \theta(n)$ one cannot improve upon the fractional clique-cover approach for broadcasting. This is answered by the following result.

Theorem 5.5.1. *There exists an explicit family of triangle-free graphs on n vertices where $\overline{\chi}_f \geq n/2$ whereas the broadcast rate satisfies $\beta \leq \frac{3}{8}n$.*

The following lemma will be the main ingredient in the construction:

Lemma 5.5.2. *For arbitrarily large integers k there exists a family $\mathcal{F}$ of subsets of $[k]$ whose size is at least $8k/3$ and has the following two properties:*

(i) Every $A \in \mathcal{F}$ has an odd cardinality.

(ii) There are no distinct $A, B, C \in \mathcal{F}$ that have pairwise odd cardinalities of intersections.

Remark 5.5.3. For k even, a simple family $\mathcal{F}$ of size $2k$ with the above properties is obtained by taking all the singletons and all their complements. However, for our application here it is crucial to obtain a family $\mathcal{F}$ of size strictly larger than $2k$.

Remark 5.5.4. The above lemma may be viewed as a higher-dimensional analogue of the Odd-Town theorem: If we consider a graph on the odd subsets with edges between those with an odd cardinality of intersection, the original theorem looks for a maximum independent set while the lemma above looks for a maximum triangle-free graph.

Proof of lemma. It suffices to prove the lemma for $k = 6$ by super-additivity (we can partition a ground-set $[N]$ with $N = 6m$ into disjoint 6-tuples and from each take the original family $\mathcal{F}$).

Let $U_1 = \big\{\{x\} : x \in [5]\big\}$ be all singletons except the last, and $U_2 = \big\{A \cup \{6\} : A \subset [5]\,,\, |A| = 2\big\}$. Clearly all subsets given here are odd.

We first claim that there are no triangles on the graph induced on U_2. Indeed, since all subsets there contain the element 6, two vertices in U_2 are adjacent iff their corresponding 2-element subsets A, A' are disjoint, and there cannot be 3 disjoint 2-element subsets of [5].

The vertices of U_1 form an independent set in the graph, hence the only remaining option for a triangle in the induced subgraph on $U_1 \cup U_2$ is of the form $\{x\}, (A \cup \{6\}), (A' \cup \{6\})$. However, to support edges from $\{x\}$ to the two sets in U_2 we must have that x belongs to both sets, and since $x \neq 6$ by definition we must have $x \in A \cap A'$. However, we must also have $A \cap A' = \emptyset$ for the two vertices in U_2 to be adjacent, contradiction.

To conclude the proof observe that adding the extra set [5] does not introduce any triangles, since U_1 is an independent set while [5] is not adjacent to any vertex in U_2 (its intersection with any set $(A \cup \{6\}) \in U_2$ contains precisely 2 elements). Altogether we have $|\mathcal{F}| = 5 + \binom{5}{2} + 1 = \frac{8}{3}k$. □

Proof of Theorem 5.5.1. Let $\mathcal{F}$ be the family provided by the above lemma and consider the graph G whose n vertices are the elements of $\mathcal{F}$ with edges between A, B whose cardinality of intersection is odd. By definition the graph G is triangle-free and we have $\overline{\chi}_f(G) \geq n/2$.

Next, consider the binary matrix M indexed by the vertices of G where $M_{A,B} = |A \cap B|$ (mod 2). All the diagonal entries of M equal 1 by the fact that $\mathcal{F}$ is comprised of odd subsets only, and clearly M is a representation of G over $GF(2)$. At the same time, M can be written as FF^{T} where F is the $n \times k$ incidence-matrix of the ground-set $[k]$ and subsets of $\mathcal{F}$. In particular we have that $\mathsf{rank}(M) \leq \mathsf{rank}(F) \leq k$ over $GF(2)$. This implies that $\mathrm{minrk}_2(G) \leq k$ and the proof is now concluded by the fact that $\beta(G) \leq \mathrm{minrk}_2(G)$. □

Remark 5.5.5. The construction of the family of subsets $\mathcal{F}$ in Lemma 5.5.2 relied on a triangle-free 15-vertex base graph H which is equivalent to the Petersen graph with 5 extra vertices added to it, each one adjacent to one of the independent sets of size 4 in the Petersen graph.

5.6 Additive Separations

Though in [7] Alon *et al.* show that for BSIP there are instances in which $\beta = 2$ while β^* is unbounded, in the constrained setting of BSIP-G the largest known values of $\beta_1 - \beta$ and $\beta^* - \beta$ were less than one. They are attained by the 5-cycle, where it was known that $\beta_1 = 3, \beta^* \approx 2.68$, and $\beta = 2.5$. These gaps could potentially be attributed to integer-rounding, and we might conjecture that for graphs $\beta_1 = \lceil \beta \rceil$ and $\beta^* < \lceil \beta \rceil$.

The following theorem refutes these suggestions by amplifying both of these gaps to be linear in n. Moreover the construction gives additive separations that are linear in n between most broadcast rates.

Theorem 5.6.1. *There exists a family of graphs G on n vertices for which $\beta(G) = \lambda(G) = \frac{1}{2}n$, $\alpha(G) = \frac{2}{5}n$, $\lambda_1(G) \geq \frac{3}{5}n$, and $\beta_1 \geq \beta^*(G) = (1 - \frac{1}{5}\log_2 5)n \approx 0.54n$.*

Proof of Theorem 5.6.1. Consider the family of graphs on $n = 5k$ vertices given by $G = k \cdot C_5$. It was shown in [7] that $\beta^*(C_5) = 5 - \log_2 5$, and Theorem 3.2.1 gives $\beta(C_5) = \lambda(C_5) = \frac{5}{2}$. It is easy to see that $\lambda_1(C_5) = \mathrm{minrk}(C_5) = 3$. The additivity of β and β^* (Theorem 4.3.1) and the additivity of α gives $\beta^*(G) = (5 - \log_2 5)k$, $\beta(G) = \frac{5}{2}k, \alpha(G) = 2k$. Additionally, Theorem 4.3.2 gives that $\lambda(G) = k\lambda(C_5) = \frac{2}{5}k$ and $\lambda_1(G) = k\lambda_1(C_5) = 3k$. Moreover, $\beta_1(G) \geq \beta^*(G) = (5 - \log_2 5)k$, as required. □

CHAPTER 6

APPROXIMATING THE BROADCAST RATE

This section is devoted to polynomial-time algorithms for approximating β and deciding whether $\beta = 2$ for BSIP. Working in the setting of a general broadcast network is somewhat delicate and we begin by sketching the arguments that will follow.

6.1 Approximating the broadcast rate in general networks

In the simpler case of undirected graphs, a $o(n)$-approximation to β is implied by results of [63, 5, 14] that together give a polynomial time procedure that finds either a small clique-cover or a large independent set (see Remark 6.1.2). To get an approximation for BSIP we will apply a similar technique using analogues of independent sets and clique-covers that give lower and upper bounds respectively on the BSIP broadcasting rate. The analogue of an independent set is an *expanding sequence* — a sequence of receivers where the i^{th} receiver's desired message is unknown to receivers $1, \ldots, i-1$ (see Definition 2.1.1). The clique-cover analogue is a fractional hyperclique-cover (see Definition 2.1.5).

We will prove that there is a polynomial time algorithm that outputs an expanding sequence of size k or reports a fractional hyperclique-cover of size $O\left(kn^{1-1/k}\right)$; the approximation follows by setting k appropriately. We will argue that either we can partition the graph and apply induction or else the side-information map is dense enough to deduce existence of a small fractional hyperclique-cover. The proof of the latter step deviates significantly from the techniques used for graphs, and seems interesting in its own right. We will give a simple procedure to randomly sample hypercliques and use it to produce a valid weight function for the hyperclique-cover by defining the weight of a hyperclique to be proportional to the probability it is sampled by the procedure.

Theorem 6.1.1. *Let G be a broadcasting with side information problem, having n messages* *and m receivers. Then there is a polynomial time algorithm which computes a parameter $\tau = \tau(G)$ such that $1 \leq \frac{\tau(G)}{\beta(G)} \leq O\big(n^{\frac{\log\log n}{\log n}}\big)$.*

Remark 6.1.2. In the setting of undirected graphs a slightly better approximation algorithm for β is a consequence of a result of Boppana and Halldorsson [14], following the work of Wigderson [63]. In [14] the authors showed an algorithm that finds either a "large" clique or a "large" independent set in a graph (where the size guarantee involves the Ramsey number estimate). A simple adaptation of this result (Proposition 2.1 in the Alon-Kahale [5] work on approximating α via the ϑ-function) gives a polynomial-time algorithm for finding an independent set of size $t_k(m) = \max\big\{s : \binom{k+s-2}{k-1} \leq m\big\}$ in any graph satisfying $\overline{\chi}(G) \geq n/k + m$. In particular, taking $m = n/k$ with $k = \frac{1}{2}\log n$ we have that either $\overline{\chi}(G) < \frac{4n}{log(n)}$ or we find an independent set of size $t_k(n/k) = \max\big\{s : \binom{.5\log(n)+s-2}{.5\log(n)-1} \leq \frac{2n}{\log(n)}\big\} \geq .5\log(n)$ for sufficiently large n in polynomial-time.

We now turn our attention to bounding the ratio $\overline{\chi}_f(G)/\alpha(G)$ for a BSIP instance G. Our goal is to show that this ratio is bounded by a function in $o(n)$. To begin with, we need an analogue of the lemma that undirected graphs with small maximum degree have small fractional chromatic number.

Lemma 6.1.3. *If G is a BSIP instance with n vertices, and d is a natural number such that for every receiver j, $|S(j)| + d \geq n$, then $\overline{\chi}_f(G) \leq 4d + 2$.*

Proof. Let us define a procedure for sampling a random subset $T \subseteq [n]$ and a random hyperclique $\mathcal{J}$ as follows. Let π be a uniformly random permutation of $[n + d]$, let i be the least index such that $\pi(i + 1) > n$, and let T be the set $\{\pi(1), \pi(2), \ldots, \pi(i)\}$. (If $\pi(1) > n$ then $i = 0$ and T is the empty set.) Now let $\mathcal{J}$ be the set of all j such that $f(j) \in T \subseteq S(j)$. (Note that $\mathcal{J}$ is indeed a hyperclique.)

For any hyperclique $\mathcal{J}$ let $p(\mathcal{J})$ denote the probability that $\mathcal{J}$ is sampled by this procedure and let $w(\mathcal{J}) = (4d + 2) \cdot p(\mathcal{J})$. We claim that the weights $w(\cdot)$ define a fractional hyperclique-cover of G, or equivalently, that for every receiver j, $\mathbb{P}(f(j) \in T \subseteq S(j)) \geq \frac{1}{4d+2}$. Let $U(j)$ denote the set $[n+d] \setminus N(j)$. The event $\mathcal{E} = \{f(j) \in T \subseteq S(j)\}$ occurs if and only if, in the ordering of $U(j)$ induced by π, the first element of $U(j)$ is $f(j)$ and the next element belongs to $[n+d] \setminus [n]$. Thus,

$$\mathbb{P}(\mathcal{E}) = \frac{1}{|U(j)|} \cdot \frac{d}{|U(j)|-1}.$$

The bound $\mathbb{P}(\mathcal{E}) \geq \frac{1}{4d+2}$ now follows from our assumption $|S(j)| + d \geq n$ which implies that $|U(j)| \leq 2d+1$. □

Lemma 6.1.4. *If G is a BSIP instance and $\alpha(G) \leq k$, then $\overline{\chi}_f(G) \leq 6kn^{1-1/k}$. Moreover, there is a polynomial-time algorithm, whose input is a hypergraph G and a natural number k, that either outputs an expanding sequence of size $k+1$ or reports (correctly) that $\overline{\chi}_f(G) \leq 6kn^{1-1/k}$.*

Proof. The proof is by induction on k. In the base case $k = 1$, either G itself is a hyperclique or there is some pair of receivers j, j' such that $f(j)$ is not in $S(j')$. In that case, the sequence $j_1 = j', j_2 = j$ is an expanding sequence of size 2.

For the induction step, for each hyperedge j define the set $D(j) = [n] \setminus N(j)$ and let j_1 be a hyperedge such that $|D(j)|$ is maximum. If $|D(j_1)| \leq n^{1-1/k} + 1$, then the bound $|S(j)| + n^{1-1/k} \geq n$ is satisfied for every j and Lemma 6.1.3 implies that $\overline{\chi}_f(G) < 4n^{1-1/k} + 2 \leq 6n^{1-1/k}$. Otherwise, partition the vertex set of G into $V_1 = [n] \setminus S(j_1)$ and $V_2 = S(j_1)$, and for $i = 1, 2$ define G_i to be the hypergraph with vertex set V_i and edge set E_i consisting of all pairs $(N(j) \cap V_i, f(j))$ such that $(N(j), f(j))$ is a hyperedge of G with $f(j) \in V_i$. (We will call such a structure the *induced sub-hypergraph of G on vertex set V_i*.) If G_1 contains an expanding sequence $j_2, j_3, \ldots, j_{k+1}$ of size k, then the sequence $j_1, j_2, \ldots, j_{k+1}$ is an expanding sequence of size $k+1$ in G. (Moreover, if an algorithm efficiently finds the sequence $j_2, j_3, \ldots, j_{k+1}$ then it is easy to efficiently construct the sequence $j_1, \ldots, j_{k+1}$.) Otherwise, by the induction hypothesis, G_1 has a fractional hyperclique-cover of weight at most $6(k-1)|V_1|^{1-1/(k-1)} \leq 6(k-1)|V_1|n^{-1/k}$. Continuing to process the induced sub-hypergraph on vertex set V_2 in the same way, we arrive at a partition of $[n]$ into disjoint vertex sets $W_1, W_2, \ldots, W_\ell$ of cardinalities $n_1, \ldots, n_\ell$, respectively, such that for $1 \leq i < \ell$, the induced sub-hypergraph on W_i has a fractional clique-cover of weight at most $6(k-1)n_i n^{-1/k}$, and for $i = \ell$ the induced sub-hypergraph on W_i satisfies the hypothesis of Lemma 6.1.3 with $d = n^{1-1/k}$ and consequently has a fractional hyperclique-cover of weight at most $6n^{1-1/k}$. The lemma follows by summing the weights of these hyperclique-covers. □

Proof of Theorem 6.1.1. We combine Lemmas 2.1.3, 2.1.7, 6.1.4. Run the algorithm described in Lemma 6.1.4 with $k = \frac{\log n}{2 \log\log n}$. If the algorithm outputs an expanding sequence of size $k+1$ then we output $\tau = n$, otherwise, we output $\tau = 6kn^{(1-1/k)}$. In both cases there is a coding scheme of size at most τ (sending all messages for the former and sending the code given by the fractional hyperclique-cover in the latter) and thus $\tau/\beta \geq 1$. If we output $\tau = n$ then we know $\beta \geq \alpha \geq k$ giving $\tau/\beta \leq \frac{2n \log\log n}{\log n}$. In the latter case, we use $\beta \geq 1$ giving $\tau/\beta \leq 6kn^{(1-(1/k))} = \frac{3n}{\log n \log\log n} = O(\frac{n \log\log n}{\log n})$ gives our result. □

6.1.1 Extending the algorithm to networks with variable source rates

The aforementioned approximation algorithm for β naturally extends to the setting where each source in the broadcast network has its own individual rate. Namely, the n message streams are identified with the elements of $[n] = V$, where message stream i has a rate r_i, and the problem input consists of the vector $(r_1, \ldots, r_n)$ and the pairs $\{(N(j), f(j))\}_{j=1}^{m}$. Thus the input is a *weighted directed hypergraph* instance. An index code for a weighted hypergraph consists of the following:

- Alphabets Σ_P and Σ_i for $1 \leq i \leq n$,

- An encoding function $\mathcal{E} : \prod_{i=1}^{n} \Sigma_i \to \Sigma_P$,
- Decoding functions $\mathcal{D}_j : \Sigma_P \times \prod_{i \in N(j)} \Sigma_i \to \Sigma_{f(j)}$.

The encoding and decoding functions are required to satisfy

$$\mathcal{D}_j(\mathcal{E}(\sigma_1, \ldots, \sigma_n), \sigma_{N(j)}) = \sigma_{f(j)}$$

for all $j = 1, \ldots, m$ and all $(\sigma_1, \ldots, \sigma_n) \in \prod_{i=1}^{n} \Sigma_i$. Here the notation $\sigma_{N(j)}$ denotes the tuple obtained from a complete n-tuple $(\sigma_1, \ldots, \sigma_n)$ by retaining only the components indexed by elements of $N(j)$. An index code *achieves* rate $r \geq 0$ if there exists a constant $b > 0$ such that $|\Sigma_i| \geq 2^{b \cdot r_i}$ for $1 \leq i \leq n$ and $|\Sigma_P| \leq 2^{b \cdot r}$. If so, we say that rate r is *achievable.* If G is a weighted hypergraph, we define $\beta(G)$ to be the infimum of the set of achievable rates.

The first step in generalizing the proof given in the previous subsection to the case where the r_i's are non-uniform is to properly extend the notions of hypercliques and expanding sequences. A weak fractional hyperclique-cover of a weighted hypergraph will now assign a weight $w(\mathcal{J})$ to every weak hyperclique $\mathcal{J}$ such that for every receiver j, $\sum_{\mathcal{J} \ni j} w(\mathcal{J}) \geq r_{f(j)}$ (cf. Definition 2.1.5 corresponding to $r_{f(j)} = 1$). As before, the weight of a fractional weak hyperclique-cover is given by $\sum_{\mathcal{J}} w(\mathcal{J})$ and for a weighted hypergraph G we let $\overline{\chi}_f(G)$ denote the minimum weight of a fractional weak hyperclique-cover. An expanding sequence $j_1, \ldots, j_k$ is defined as before (see Eq. 2.1.1) except now we associate such a sequence with the weight $\sum_{\ell=1}^{k} r_{f(j_\ell)}$ and the quantity $\alpha(G)$ will denote the maximum weight of an expanding sequence (rather than the maximum cardinality).

With these extended defintions, the proofs in the previous subsection carry unmodified to the weighted hypergraph setting with the single exception of Lemma 6.1.4, where the assumption that the hypergraph is unweighted was essential to the proof. In what follows we will qualify an application of that lemma via a dyadic partition of the vertices of our weighted hypergraph according to their weights r_i.

Assume without loss of generality that $0 \le r_i \le 1$ for every vertex $i \in [n]$, and partition the vertex set of G into subsets $V_1, V_2, \dots$ such that V_s contains all vertices i such that $2^{-s} < r_i \le 2^{1-s}$. Let G_s denote the induced hypergraph on vertex set V_s. For each of the nonempty hypergraphs G_s, run the algorithm in Lemma 6.1.4 for $k = 1, 2, \dots$ until the smallest value of $k(s)$ for which an expanding sequence of size $k(s)+1$ is not found. If G_s° denotes the unweighted version of G_s, then we know that

$$\begin{aligned}
\alpha(G_s) &\ge 2^{-s}\alpha(G_s^\circ) \ge 2^{-s}k(s) \\
\overline{\chi}_f(G_s) &\le 2^{1-s}\overline{\chi}_f(G_s^\circ) \le 2^{-s} \cdot 12k(s)n^{1-1/k(s)}.
\end{aligned}$$

In addition, for each $i \in V_s$ the set of hyperedges containing i constitutes a hyperclique, which implies the trivial bound

$$\overline{\chi}_f(G_s) \le \sum_{i \in V_s} r_i \le 2^{1-s}|V_s|.$$

Combining these two upper bounds for $\overline{\chi}_f(G_s)$, we obtain an upper bound for $\overline{\chi}_f(G)$:

$$\overline{\chi}_f(G) \le \sum_{s=1}^{\infty} \overline{\chi}_f(G_s) \le \sum_{s=1}^{\infty} 2^{-s} \cdot \min\left\{12k(s)n^{1-1/k(s)},\, 2|V_s|\right\}. \tag{6.1}$$

We define $\tau(G)$ to be the right side of (6.1). We have described a polynomial-time algorithm to compute $\tau(G)$ and have justified the relation $\overline{\chi}_f(G) \le \tau(G)$, so it remains to show that $\tau(G)/\alpha(G) \le cn\left(\frac{\log\log n}{\log n}\right)$ for some constant c.

The bound $\tau(G) \le n$ follows immediately from the definition of τ, so if $\alpha(G) \ge \frac{\log n}{\log\log n}$ there is nothing to prove. Assume henceforth that $\alpha(G) < \frac{\log n}{\log\log n}$, and define w to be the smallest integer such that $2^w \cdot \alpha(G) > \frac{\log n}{2\log\log n}$. We have

$$\begin{aligned}
\tau(G) &\le \sum_{s=1}^{w} 2^{-s} \cdot 12k(s)n^{1-1/k(s)} \;+\; \sum_{s=w+1}^{\infty} 2^{1-s} \cdot |V_s| \\
&\le 12n\sum_{s=1}^{w} 2^{-s}k(s)n^{-1/k(s)} \;+\; 2^{-w} \cdot n \\
&< 12n\alpha(G)\sum_{s=1}^{w} n^{-1/k(s)} \;+\; 2n\alpha(G)\left(\frac{\log\log n}{\log n}\right),
\end{aligned} \tag{6.2}$$

with the last line derived using the relations $2^{-s}k(s) \le \alpha(G_s) \le \alpha(G)$ and $2^{-w} < \alpha(G)\left(\frac{2\log\log n}{\log n}\right)$. Applying once more the fact that $2^{-s}k(s) \le \alpha(G)$, we find that $n^{-1/k(s)} \le n^{-1/(2^s \cdot \alpha(G))}$. Substituting this bound into (6.2) and letting α denote $\alpha(G)$, we have

$$\frac{\tau(G)}{\alpha(G)} \le 2n\left(\frac{\log\log n}{\log n}\right) \;+\; 12n\left(n^{-1/2\alpha} + n^{-1/4\alpha} + \cdots + n^{-1/2^w\alpha}\right).$$

In the sum appearing on the right side, each term is the square of the one following it. It now easily follows that the final term in the sum is less than 1/2, so the entire sum is bounded above by twice its final term. Thus

$$\frac{\tau(G)}{\alpha(G)} \le 2n\left(\frac{\log\log n}{\log n}\right) \;+\; 24n \cdot n^{-1/2^w\alpha}. \tag{6.3}$$

Our choice of w ensures that $2^w\alpha \le \frac{\log n}{\log\log n}$ hence $n^{-2^{-w}a} \le n^{-\log\log n/\log n} = (\log n)^{-1}$. By substituting this bound into (6.3) we obtain

$$\frac{\tau(G)}{\alpha(G)} \le n\left(\frac{2\log\log n}{\log n} + \frac{24}{\log n}\right),$$

as desired.

Open Question 6.1.5. Can this algorithm be extended to give an approximation algorithm for general network coding?

There is no nontrivial approximation known for general network coding. Recently, it was shown that every network coding instance can be reduced to an equivalent BSIP instance [26]. This reduction is not approximation preserving, but perhaps there is a way to use it along with the approximation for the more general weighted case to show an approximation for network coding.

6.2 Determining whether the broadcast rate equals 2

This section is devoted to proving the following theorem.

Theorem 6.2.1. *There is a polynomial time algorithm to decide whether $\beta(G) = 2$.*

We will prove that a structure called an *almost alternating cycle* (AAC) constitutes a minimal obstruction to obtaining a broadcast rate of 2. The proof makes crucial use of the Shannon lower bound, LP $\mathfrak{B}$, calculating the parameter b for AAC's to prove that their broadcast rate is strictly greater than 2. Furthermore, the proof reduces finding an AAC to finding the transitive closure of a particular relation, which is polynomial time computable.

Let G be an undirected graph with independence number $\alpha = 2$. Clearly, if $\overline{G}$ is bipartite then $\overline{\chi}(G) = 2$ and so $\beta(G) = 2$ as well. Conversely, if $\overline{G}$ is not bipartite then it contains an odd cycle, the smallest of which is induced and has at least five vertices since the maximum clique in $\overline{G}$ is $\alpha(G) = 2$. In particular, Theorem 3.2.1 implies that $\beta(G) \geq \beta(\overline{C_n}) = \frac{n}{\lfloor n/2 \rfloor} > 2$. We thus conclude the following:

Corollary 6.2.2. *Let G be an undirected graph on n vertices whose complement $\overline{G}$ is nonempty. Then $\beta(G) = 2$ if and only if $\overline{G}$ is bipartite.*

A polynomial time algorithm for determining whether $\beta = 2$ in BSIP-G follows as an immediate consequence of Corollary 6.2.2. However, for BSIP — or even for the special case of directed graphs (the main setting of [8, 9]) — it is unclear whether such an algorithm exists. In this section we provide such an algorithm, accompanied by a characterization theorem that generalizes the above characterization for undirected graphs. Recall that $S(j)$ denotes the set $N(j) \cup \{f(j)\}$ and $T(j)$ denotes the complement of $S(j)$ in V. We will assume, without loss of generality, that for every $i \in V$ there is an edge j with $f(j) = i$.

Definition 6.2.3. If $G = (V, E)$ is a directed hypergraph and S is a set, a function $F : V \to S$ is said to be *G-compatible* if for every edge $j \in E$, there are two *distinct* elements $t, u \in S$ such that F maps every element of $T(j)$ to t, and it maps $f(j)$ to u.

Definition 6.2.4. If $G = (V, E)$ is a directed hypergraph, an *almost alternating k-cycle* in G is a sequence of k distinct edges $j_1, \ldots, j_k$ and k distinct vertices $v_1, v_2, \ldots, v_k$, such that for $i = 1, \ldots, k-1$ the set $T(j_i)$ contains v_i and v_{i+1}, and the set $T(j_k)$ contains v_k while $f(j_k) = v_1$.

Theorem 6.2.5. *For a directed hypergraph G the following are equivalent:*

(i) $\beta(G) = 2$

(ii) There exists a set S and a G-compatible function $F : V \to S$.

(iii) G contains no almost alternating cycles.

Furthermore there is a polynomial-time algorithm to decide if these equivalent conditions hold.

Proof. **(i)⇒(iii):** The contrapositive statement says that if G contains an almost alternating cycle then $\beta(G) > 2$. Let $j_1, \ldots, j_k$ be the edges of an almost alternating k-cycle with vertices $v_1, \ldots, v_k$. We will use LP $\mathfrak{B}$ to show that $b(G) > 2$. As we did in Section 3.2, we will bring the subscript up from the LP variables and for $S \subseteq V$ we use $z(S)$ in place of z_S. Additionally, we let $\overline{z}(S)$ denote $z(\overline{S})$ and let $S_{i:l}$ denote the set $\{v_i, v_{i+1}, \ldots, v_l\}$.

For $0 < i < k$, we have

$$z(\emptyset) + |V| - 3 \geq \overline{z}(\{f(j_i), v_i, v_{i+1}\}) = \overline{z}(\{v_i, v_{i+1}\}) = \overline{z}(S_{i:i+1})\,, \tag{6.4}$$

which hold by decoding ($v_i, v_{i+1} \notin N(j_i)$).

Summing up (6.4) for $i = 1, \ldots, k-1$ gives

$$(k-1)z(\emptyset) + (k-1)(|V| - 3) \geq \sum_{i=1}^{k-1} \overline{z}(S_{i:i+1}) \tag{6.5}$$

Using submodularity we have that for $1 < i < k$,

$$\overline{z}(S_{1:i}) + \overline{z}(S_{i:i+1}) \geq \overline{z}(S_{1:i+1}) + \overline{z}(\{v_i\}) = \overline{z}(S_{1:i+1}) + z(V) = \overline{z}(S_{1:i+1}) + |V|\,. \tag{6.6}$$

Summing up (6.6) for $i = 2, \ldots, k-1$ and canceling terms that appear on both sides, we obtain

$$\sum_{i=1}^{k-1} \overline{z}(S_{i:i+1}) \geq \overline{z}(S_{1:k}) + (k-2)|V|\,. \tag{6.7}$$

Combining (6.5) with (6.7) we obtain

$$(k-1)z(\emptyset) + (k-1)(|V|-3) \geq \overline{z}(S_{1:k}) + (k-2)|V|\,. \tag{6.8}$$

Now, observe that

$$\overline{z}(S_{1:k}) + k - 2 \geq \overline{z}(\{v_1, v_k\}) \geq \overline{z}(\{v_k\}) \geq z(V) = |V|\,, \tag{6.9}$$

where all the inequalities are due to decoding. The second because $f(j_k) = v_1$ and $v_k \in T(j_k)$, and the third by our assumption that all messages are desired by at least one receiver. Summing (6.8) and (6.9), we obtain

$$(k-1)z(\emptyset) + (k-1)(|V|-3) + k - 2 \geq (k-1)|V|$$

and rearranging we get $z(\emptyset) \geq 2 + (k-1)^{-1}$, from which it follows that $\beta(G) \geq b(G) \geq 2 + (k-1)^{-1}$.

(iii)⇒(ii): Define a binary relation $\sharp$ on the vertex set V by specifying that $v \sharp w$ if there exists an edge j such that $\{v, w\} \subseteq T(j)$. Let $\sim$ denote the transitive closure of $\sharp$. Define F to be the quotient map from V to the set S of equivalence classes of $\sim$. We need to check that F is G-compatible. For every edge $j \in E$, the definition of relation $\sharp$ trivially implies that F maps all of $T(j)$ to a single element of S. The fact that it maps $f(j)$ to a *different* element of S is a consequence of the non-existence of almost alternating cycles. A relation $f(j) \sim v$ for some $v \in T(j)$ would imply the existence of a sequence $v_1, \ldots, v_k$ such that $v_1 = f(j), v_k = v$, and $v_i \sharp v_{i+1}$ for $i = 1, ..., k-1$. Let $v_1, \ldots, v_k$ be the shortest such sequence. If we choose j_i for $0 < i < k$ to be an edge such that $T(j_i)$ contains v_i, v_{i+1} (such an edge exists because $v_i \sharp v_{i+1}$) and we set $j_k = j$, then the vertex sequence $v_1, \ldots, v_k$ and edge sequence $j_1, \ldots, j_k$ constitute an almost alternating cycle in G. It remains to verify that

$j_1, \dots, j_k$ are distinct. If $j_i = j_l$ for $i < l < k$, then $v_i, v_{l+1} \in T(j_i)$ and we have a shorter sequence after removing $j_{i+1}, \dots, j_l$, and if $j_i = j_k$ we have a shorter sequence ending at j_i, both a contradiction to choosing the shortest sequence $v_1, \dots v_k$.

Computing the relation $\sim$ and the function F, as well as testing that F is G-compatible, can easily be done in polynomial time, implying the final sentence of the theorem statement.

(ii)⇒(i): If $F : V \to S$ is G-compatible, we may compose F with a one-to-one mapping from S into a finite field $\mathbb{F}$, to obtain a function $\phi : V \to \mathbb{F}$ that is G-compatible. The public channel broadcasts two elements of $\mathbb{F}$, namely:

$$y = \sum_v x_v$$
$$z = \sum_v \phi(v) x_v, \; x_v \in \mathbb{F}$$

Receiver R_j now decodes message $x(j)$ as follows. Let c denote the unique element of $\mathbb{F}$ such that $\phi(v) = c$ for every v in $T(j)$. Using the pair (y, z) from the public channel, R_j can form the linear combination

$$cy - z = \sum_v [c - \phi(v)] x_v.$$

We know that every $v \in T(j)$ appears with coefficient zero in this sum. For every $v \in N(j)$, receiver R_j knows the value of x_v and can consequently subtract off the term $[c - \phi(v)]x_v$ from the sum. The only remaining term is $[c - \phi(x(j))]x(j)$. The coefficient $c - \phi(x(j))$ is nonzero, because ϕ is G-compatible. Therefore R_j can decode $x(j)$. □

CHAPTER 7

BEYOND BROADCASTING: GRAPH PRODUCTS AND THE NETWORK CODING RATE

We consider the coding analogue of the maximum multicommodity flow problem. The maximum multicommodity flow problem is closely related to the concurrent multicommodity flow problem; it differs only in its objective function. In the maximum multicommodity flow problem there is no notion of fairness between commodities. The objective is simply to maximize the total flow sent between source-sink pairs. Though maximum and concurrent versions of the multicommodity flow problem are different from a practical perspective, the flow-cut and flow-coding gap results discussed in Section 1.1.3 for the concurrent variant still hold in the maximization variant with the correct notion of cut.

For the concurrent variant, the cut bound we use is the integral solution to the dual of the concurrent multicommodity flow linear program. We use the corresponding cut bound here and obtain what we call the *multicut*. The minimum multicut is the minimum size edge set that, when removed, disconnects all source-sink pairs. Just like the concurrent variant, the multicut is not an upper bound on the coding rate in directed networks and can even be a factor k smaller [4].

We study the relationship between multicuts and coding solutions for a special class of networks.

This study has implications not only for network coding, but also for the multicut problem itself. The multicut problem is a fundamental graph partitioning problem and has applications in network robustness where we may want guarantees that the multicut is large implying our network will still be connected even after the failure of many edges. Alternatively, we may want to compute a small multicut in order to determine an efficient way to stop the spread of a contagion in a network.

The multicut problem is known to be NP-hard to compute and even NP-hard to approximate [21, 20]. The best approximation algorithm known for directed graphs is $\tilde{O}(n^{11/23})$ [2]. All of the approximation algorithms [32, 2, 19] to date bound the solution value via the solution to the maximum multicommodity flow problem. This technique is limited by the flow-cut gap, and the gap is known to be $\Omega(k)$ [60] and $\tilde{\Omega}(n^{1/7})$ [20]. Thus, the lower bound given by the maximum multicommodity flow problem isn't strong enough to allow for improved approximation algorithms when parameterized by k.

This chapter considers the possibility of a stronger lower bound via network coding. We introduce a technique to certify when the network coding rate is a lower bound on the multicut, or in other words, when the multicut is an upper bound on the network coding rate. We identify a property of a linear network code that guarantees the code is a lower bound on the multicut. We also show that for the strong graph product of any two networks with such codes, this property is preserved. The following theorem describes one consequence of our main result:

Theorem 7.0.1. *Given a network G in which the optimal multicommodity flow solution consists of a set of node-disjoint paths, there is a product operation in which the optimal network coding rate is equal to the minimum multicut in the k-fold product of G.*

By applying this theorem to a directed path of length n with source and sink at the ends, we give a new lower bound on the multicut in the construction of Saks *et al.* Our proof strengthens Saks's result and provides a tight lower bound on the multicut (see Corollary 7.3.1). Further, it constructs an elegant network coding solution for the construction that has rate equal to the multicut and a $k - o(k)$ factor larger than the multicommodity flow rate. This implies that the construction of Saks *et al.* does not give even give an example where the coding rate can strictly less than the multicut, let alone a factor $\Omega(k)$ smaller.

7.1 Preliminaries

We begin by defining the class of networks for which we analyze the multicut and network coding rates. The definition is tailor-made for taking graph products. All of the definitions in this chapter are self-contained. In particular, we give variations of the definition of a network code, a network coding solution, and the strong graph product. Though essentially the same, these definitions are better adapted for the class of networks we consider.

Definition 7.1.1. A *node-capacitated multicommodity instance* is given by a tuple $N = (G, \mathcal{S}, \mathcal{T}, f)$ where $G = (V, E)$ is an undirected graph, $\mathcal{S}$ and $\mathcal{T}$ are an ordered list of sources and sinks (separate from G) such that the i^{th} source and sink are paired, and $f : \mathcal{S} \cup \mathcal{T} \mapsto 2^V$ is a function that maps each source and sink to a subset of nodes. The instance network can be formed by adding nodes for each element in $\mathcal{S}$ and $\mathcal{T}$ to G and adding directed edges (s, v) for all $s \in \mathcal{S}, v \in f(s)$ and (u, t) for all $t \in \mathcal{T}, u \in f(t)$. We reserve n to denote $|V|$.

It is easier for us to work with node-capacitated networks, but any node-capacitated network can be transformed into an equivalent edge-capacitated network by replacing each node with two nodes with a single directed edge between them. For this reason, even though the graph G is undirected, the network we are considering is far from undirected.

We will show that under certain conditions linear network codes and multicuts in these network instances can be composed under the following product operation.

Definition 7.1.2. The strong product of two instances $N_1 = (G_1, \mathcal{S}_1, \mathcal{T}_1, f_1)$ and $N_2 = (G_2, \mathcal{S}_2, \mathcal{T}_2, f_2)$ is the instance $N_1 \boxtimes N_2 = (G_1 \boxtimes G_2, \mathcal{S}, \mathcal{T}, f)$ where $G_1 \boxtimes G_2$ is the strong graph product of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$:

$$\begin{aligned} V(G_1 \boxtimes G_2) &= V_1 \times V_2 \\ E(G_1 \boxtimes G_2) &= \{((u,v),(u',v')) | (u,v) \neq (u',v') \\ &\qquad u = u' \text{or } (u,u') \in E_1, \\ &\qquad v = v' \text{or } (v,v') \in E_2\}. \end{aligned}$$

The set of sources $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$. The function f is defined by

$$f(s) = \begin{cases} f_1(s) \times V_2 & \text{if } s \in \mathcal{S}_1 \\ V_1 \times f_2(s) & \text{if } s \in \mathcal{S}_2 \end{cases}$$

The sinks $\mathcal{T}$ and function $f(\mathcal{T})$ are defined in the corresponding manner.

Our analysis relies heavily on matrices and we now define the notation and important definitions. Let $A[i,j]$ denote the $(i,j)^{th}$ entry of A, $A[i,-]$ the i^{th} row, and $A[-,j]$ the j^{th} column. Correspondingly, for a vector v, let $a[i]$ denote the i^{th} entry of a.

Definition 7.1.3. The Kronecker product of a $p \times q$ matrix A and $p' \times q'$ matrix B is a $pp' \times qq'$ matrix

$$A \otimes B = \begin{bmatrix} a[1,1]B & \cdots & a[1,q]B \\ \vdots & \ddots & \vdots \\ a[p,1]B & \cdots & a[p,q]B \end{bmatrix}.$$

Definition 7.1.4. The *support* of a vector $v \in \mathbb{F}^{|A|}$, denoted $\mathsf{supp}(v)$, with entries indexed by the set A is the subset $A' \subseteq A$ such that $v[a] \neq 0$ iff $a \in A'$. In other words, $\mathsf{supp}(v)$ is the support of the function $f : A \mapsto \mathbb{F}$ such that $f(a) = v[a]$.

We will overload functions defined on elements of sets to also be defined on subsets. For a function $f : 2^A \mapsto 2^B$ and a subset $A' \subseteq A$, we define $f(A') := \bigcup_{a \in A'} f(a)$. For a function $f : 2^A \mapsto \mathbb{R}$, we define $f(A') := \sum_{a \in A'} f(a)$. Often we will use the additional shorthand of denoting $f(A)$ by f.

7.2 Codes and Cuts

There are some subtleties to defining network coding solutions in graphs with cycles [48]. To avoid these issues we restrict our definition of a network code to include an ordering on nodes that specifies possible dependencies between message vectors.

Definition 7.2.1. A *linear network code* $(\mathbb{F}, r, \pi, L)$ of a node-capacitated multicommodity instance $((V, E), \mathcal{S}, \mathcal{T}, f)$ specifies a finite field $\mathbb{F}$, a function $r(s) : \mathcal{S} \mapsto \mathbb{N}$, an ordering $\pi : V \mapsto [n]$ on nodes in V, and a $n \times r(\mathcal{S})$ coding matrix L. The rows of L are labeled with vertices V and the columns by messages $\mathcal{M} := \bigcup_{s \in \mathcal{S}} \mathcal{M}(s)$, where $\mathcal{M}(s) := \{(s, 1), \ldots, (s, r(s))\}$. Defining $N(v)$ to be $\{v\} \cup \{u \in V | \pi(u) < \pi(v), (u, v) \in E\}$, we have that:

For $v \in V$, $\exists a_v \in \mathbb{F}^{1 \times n}$ such that

1. $\{v\} \subseteq \mathsf{supp}(a_v) \subseteq N(v)$,
2. $\mathsf{supp}(a_v L) \subseteq \mathcal{M}(f^{-1}(v))$.

The v^{th} row of the matrix L describes the linear combination over $\mathbb{F}$ of messages that are sent by node v to all its neighbors in the code. The existence of vector a_v guarantees that v can compute this linear combination using the messages of adjacent nodes that come earlier in the ordering π. In particular, node v can determine its message using $\frac{1}{a_v[v]} \sum_{v' \in N(v) \setminus \{v\}} a_v[v'] L[v', -]$ and the information from the sources entering node v.

Definition 7.2.2. A linear network code $(\mathbb{F}, r, \pi, L)$ of a node-capacitated multicommodity instance $((V, E), \mathcal{S}, \mathcal{T}, f)$ is *decodable* with rate p if there is a subset D of messages $\mathcal{M}$ of L with $|\mathcal{M}| - |D| = p$ such that:

For each message $m = (s_i, j) \in \mathcal{M} \setminus D$, $\exists d_m \in \mathbb{F}^{1 \times n}$ such that

1. $\mathsf{supp}(d_m) \subseteq f(t_i)$

2. $\{m\} \subseteq \mathsf{supp}(d_m L) \subseteq \{m\} \cup D$.

Definition 7.2.2 guarantees that for a message $m \in \mathcal{M}(s_i)$, the sink t_i can decode m assuming that the messages in D are fixed and known to all the receivers. The idea that we can set some messages as fixed is an unusual, but natural, generalization of the standard way to describe a linear code. It will allow us to write the coding matrices in a much nicer form.

Observation 7.2.3. *A network code that sends source messages along p node-disjoint paths is a linear network code that is decodable with rate p.*

Proof. The matrix L has a column for each path that is an indicator vector for the path, and the set $D = \emptyset$. □

Definition 7.2.4. A *multicut* of a node-capacitated multicommodity instance $N = ((V,E), \mathcal{S}, \mathcal{T}, f)$ is a subset of nodes $M \subseteq V$ such that removing the vertices of M from N disconnects all paths between all $s_i - t_i$ pairs.

It will be convenient for us to represent subsets of the vertices of a network in terms of an *indicator matrix*. For a subset $A \subseteq V$, the matrix I_A will be a $n \times |A|$ matrix with rows indexed by nodes $v \in V$ and columns indexed by nodes $w \in A$ where entry $[v,w] = 1$ if $v = w$ and zero otherwise.

Definition 7.2.5. We call a linear network code $C = (\mathbb{F}, r, \pi, L)$ of a node-capacitated multicommodity instance N *ρ-certifiable* if

1. There are cliques $K(v) \subseteq N(v)$, $\forall\ v \in V$ such that C continues to satisfy all of the properties prescribed in the definition of a linear network code (Definition 7.2.1) if we replace all occurrences of $N(v)$ in that definition with $K(v)$ for all $v \in V$.

2. For any multicut M of N, $\mathsf{rank}(L^T I_M) \geq \rho$.

The certifiable property implies that ρ is a lower bound on the size of the multicut: $|M| = \mathsf{rank}(I_M) \geq \mathsf{rank}(L^T I_M) \geq \rho$. The restriction on the coding matrix given by property 1 will allow us to compose together certifiable coding matricies to get a coding matrix that is certifiable for $N_1 \boxtimes N_2$ as well. Notice that we don't need the matrix to be decodable with any rate for it to be certifiable.

Observation 7.2.6. *Any coding solution consisting of r disjoint paths is r-certifiable.*

Proof. Let $(\mathbb{F}, r, \pi, L)$ be the linear code describing the disjoint path solution.

Observe that $(L^T I_M)[i,j] \neq 0$ iff path i intersects node j of M. M is a multicut, so no row $(L^T I_M)[i,-]$ can be the zero vector. Further, the paths are disjoint, so each column $(L^T I_M)[-,j]$ can have at most one non-zero entry. Thus, $\mathsf{rank}(L^T I_M) = r$, the number of rows in L^T. Further, if v belongs to a disjoint path P then v can compute its message using only its predecessor in P, thus Definition 7.2.1 will still hold if we use the subset of $N(v)$ consisting of v and its predecessor in P, a 2-clique. □

7.3 Preserving Properties in Products

Our main theorem shows how to combine linear network codes in two networks to obtain a linear network code in their product, preserving both decodability and certifiability.

Theorem 7.3.1. *Let*

$$N_1 = (G_1 = (V_1, E_1), \mathcal{S}_1, \mathcal{T}_1, f_1) \text{ and}$$

$$N_2 = (G_2 = (V_2, E_2), \mathcal{S}_2, \mathcal{T}_2, f_2)$$

be node-capacitated multicommodity instances with linear coding solutions $C_1 = (\mathbb{F}, r_1, \pi_1, L_1)$ *and* $C_2 = (\mathbb{F}, r_2, \pi_2, L_2)$.

There is a linear network coding solution C *for* $N_1 \boxtimes N_2$ *with coding matrix* $[I_{n_1} \otimes L_2, L_1 \otimes I_{n_2}]$ *such that:*

1. *If* C_1 *and* C_2 *are decodable with rates* p_1, p_2 *respectively then and* C *is decodable with rate* $p := n_1 p_2 + n_2 p_1 - p_1 |f_2(\mathcal{T}_2)|$.
2. *If* C_1 *and* C_2 *are* ρ_1 *and* ρ_2 *certifiable respectively, then* C *is* ρ*-certifiable,* $\rho := (n_1\rho_2 + n_2\rho_1 - \rho_1|f_2(\mathcal{S}_2)|)$, *for* $N_1 \boxtimes N_2$.

Before proving the main theorem we show how it applies to give an improvement to the Saks *et al.* construction. The network in the construction of Saks *et al.* is the k-fold strong product of the network $\mathcal{P}_n = (P_n, \mathcal{S} = \{s\}, \mathcal{T} = \{t\}, f)$ where $P_n = p_1 p_2 \ldots p_n$ is a path of length n and $f(s) = p_1, f(t) = p_n$. Let $\mathcal{P}_n^{\boxtimes k}$ denote the Saks *et al.* graph parameterized by k and n.

Corollary 7.3.1. *The size of the minimum multicut and the rate of the optimal network coding solution of* $\mathcal{P}_n^{\boxtimes k}$ *is* $n^k - (n-1)^k$.

This bound on the multicut is tight and an improvement over the lower bound of $k(n-1)^{k-1}$ given in Saks *et al.* [60].

Proof. From Observations 7.2.3 and 7.2.6 we know that $\mathcal{P}_n$ has a linear network code

$$C = (\mathbb{F}_2, r : r(s) = 1, \pi : \pi(p_i) = i, \mathbf{1}_n)$$

that is decodable with rate 1 and 1-certifiable.

We will fix n and apply Theorem 7.3.1 inductively on k to show that there is a code C_k for $\mathcal{P}_n^{\boxtimes k}$ is ρ_k-certifiable and decodable with rate p_k, where $\rho_k = p_k = n^k - (n-1)^k$. The preceding paragraph establishes that $C_1 = C$ satisfies the base case. Now, assuming true for k, we show for $k+1$:

We apply Theorem 7.3.1 to $N_1 = \mathcal{P}_n$ and $N_2 = \mathcal{P}_n^{\boxtimes k}$. By our inductive hypotheis, we have codes C_1 and C_2 with the required conditions, and now the theorem implies that

$$\begin{aligned}\rho_{k+1} &= \rho_k n + \rho n^k - \rho_k |f(\mathcal{S})| \\ &= \rho_k(n-1) + n^k \quad \text{by } \rho = 1, |f(\mathcal{S})| = 1 \\ &= n^{k+1} - (n-1)^{k+1}\end{aligned}$$

The same proof applies to the coding rate because $p = 1, |f(\mathcal{T})| = 1$.

Further, note that $|f(\mathcal{T}_{\mathcal{P}_n^{\boxtimes k}})| = n^k - (n-1)^k$ as well, because for $A \subset V_1$ and $B \subset V_2$, the set $A \times B$ has cardinality $|A|n_2 + |B|n_1 - |A||B|$, and again the same inductive proof holds because $|f(\mathcal{T})| = 1$. This gives us that $f(\mathcal{T}_{\mathcal{P}_n^{\boxtimes k}})$ is an optimal multicut. Additionally, $f(\mathcal{T}_{\mathcal{P}_n^{\boxtimes k}})$ cuts all sources from all sinks and therefore gives a tight upper bound on the coding rate. □

The same proof also implies the following more general corollary, giving us a large set of graphs where the coding rate is a lower bound on the multicut and better than the flow bound.

Corollary 7.3.2. *If a node-capacitated multicommodity instance $N = (G, \mathcal{S}, \mathcal{T}, f)$ has a flow solution consisting of r disjoint paths, and $|f(\mathcal{S})| = |f(\mathcal{T})| = r$, then $N^{\boxtimes k}$ has an optimal coding rate equal to the size of the optimal multicut equal to $n^k - (n-r)^k$.*

The proof of Theorem 7.3.1 mostly falls out of manipulation of the Kronecker product, in particular, we repeatedly use of the mixed-product property which states that $(A \otimes B)(C \otimes D) = AC \otimes BD$ if the dimensions match correctly.

To aid in the proof of the second part of Theorem 7.3.1, we begin with some definitions and lemmas whose proofs will come later.

Definition 7.3.3. A *lower block triangular matrix* is a block matrix such that the blocks above the main diagonal blocks are identically zero.

Lemma 7.3.4. *If the main diagonal blocks of a lower block triangular matrix have ranks $r_1, r_2, \ldots, r_l$ respectively, then the lower block triangular matrix has rank at least $\sum_{i=1}^{l} r_i$.*

The following lemma is the generalization of a critical Lemma from the Saks *et al.* proof.

Lemma 7.3.5. *For every multicut M of $N_1 \boxtimes N_2$ and every vertex $u \in V_1$ there is a multicut M_u of N_2 such that $K_1(u) \times M_u \subseteq M$.*

Note that by the symmetry of the product operation, Lemma 7.3.5 also implies that the result holds when we switch the roles of N_1 and N_2.

Now we come to proving our main theorem. To avoid confusion, we will reserve u to denote nodes in V_1 and v for V_2.

Proof of Theorem 7.3.1. We define a linear network code $C = (\mathbb{F}, r, \pi, L)$ on $N_1 \boxtimes N_2 = (G_1 \boxtimes G_2, \mathcal{S}, \mathcal{T}, f)$. It has

$$r(s) = \begin{cases} r_1(s)n_2 & \text{if } s \in \mathcal{S}_1 \\ r_2(s)n_1 & \text{if } s \in \mathcal{S}_2 \end{cases}$$

The ordering π will be given by $\pi((u, v)) = n_2(\pi_1(u) - 1) + \pi_2(v)$, which corresponds to a lexicographic ordering of $(\pi_1(u), \pi_2(v))$, and $L = [I_{n_1} \otimes L_2, L_1 \otimes I_{n_2}]$.

In L, the rows are labeled by vertices $(u, v) \in V_1 \times V_2$ and the columns are labeled with messages $\mathcal{M} = (\mathcal{M}_1 \times V_2) \cup (V_1 \times \mathcal{M}_2)$.

C is a linear network code for $\mathbf{N_1 \boxtimes N_2}$

We show that C satisfies Definition 7.2.1. Let a_u and a_v be the vectors that satisfy Definition 7.2.1 for $u \in V_1$ and $v \in V_2$ for C_1 and C_2 respectively. Now, set $a = a_u \otimes a_v$. We claim that a satisfies Definition 7.2.1 for $(u, v) \in V_1 \times V_2$ for $N_1 \boxtimes N_2$.

First, note that $\mathsf{supp}(a) = \mathsf{supp}(a_u) \times \mathsf{supp}(a_v)$, giving us that $\mathsf{supp}(a) \subseteq N(u) \times N(v) \subseteq N((u, v))$ as wanted. Additionally, $a_u[u] \neq 0, a_v[v] \neq 0$ implies that $a[(u, v)] \neq 0$, and $\{(u, v)\} \subseteq \mathsf{supp}(a)$.

The fact that $\mathsf{supp}(aL) \subseteq \mathcal{M}(f^{-1}((u, v)))$ follows from the mixed-product property:

$$\begin{aligned}\mathsf{supp}(aL) &= \mathsf{supp}(a_u \otimes a_v L_2) \cup \mathsf{supp}(a_u L_1 \otimes a_v)\\ &\subseteq (V_1 \times \mathcal{M}_2(f_2^{-1}(v))) \cup (\mathcal{M}_1(f_1^{-1}(u)) \times V_2)\\ &= \mathcal{M}(f^{-1}((u, v)))\end{aligned}$$

C is decodable with rate p

Let $D_1 \subset \mathcal{M}_1$, $D_2 \subset \mathcal{M}_2$, and $d^1_c, d^2_{c'}$ for $c \in \mathcal{M}_1 \setminus D_1, c' \in \mathcal{M}_2 \setminus D_2$ be the subsets and vectors showing that C_1 and C_2 satisfy Definition 7.2.2.

We will show that C is p-decodable with

$$D = (D_1 \times V_2) \cup (V_1 \times D_2) \cup (\mathcal{M}_1 \times f_2(\mathcal{T}_2)).$$

Note that $|D| = |D_1|n_2 + |D_2|n_1 + (p_1 - |D_1|)|f_2(\mathcal{T}_2)|$, and thus $|\mathcal{M}| - |D| = p$ as needed.

We first consider message $m = (u, m_2) = (u, (s'_i, j)) \in (V_1 \times \mathcal{M}_2) \setminus D$. Let $d_m = \mathbb{1}_u \otimes d^2_{m_2}$. We have that $\mathsf{supp}(d_m) \subseteq \{u\} \times f_2(t'_i) \subseteq f(t'_i)$.

Additionally,

$$\begin{aligned}\mathsf{supp}(d_m L) &= \mathsf{supp}\left(\left[\mathbb{1}_u \otimes d^2_{m_2} L_2, \mathbb{1}_u L_1 \otimes d^2_{m_2}\right]\right)\\ &\subseteq (\{u\} \times (\{m_2\} \cup D_2)) \cup (\mathcal{M}_1 \times f_2(t_i))\\ &\subseteq \{m\} \cup D\end{aligned}$$

Finally, because $\{m_2\} \subseteq \mathsf{supp}(d^2_{m_2} L_2)$, we also have $\{m\} \subseteq \mathsf{supp}(d_m L)$, as needed.

Now we consider message $m = (m_1, v) = ((s_i, j), v) \in \mathcal{M}_1 \times V_2$. Similar to the previous case, we define $d_m = d^1_{m_1} \otimes \mathbb{1}_v$ and by parallel arguments, we have that $\mathsf{supp}(d_m) \subseteq f(t_i)$ and $\{m\} \subseteq \mathsf{supp}(d_m L)$.

To determine the set that contains the support of $d_m L$ we can write down the same set as before, but because D is not symmetric, we can't come to our desired conclusion.

$$\begin{aligned}\mathsf{supp}(d_m L) &= \mathsf{supp}\left(\left[d^1_{m_1} \otimes \mathbb{1}_v L_2, d^1_{m_1} L_1 \otimes \mathbb{1}_v\right]\right)\\ &\subseteq (f_1(t_i) \times \mathcal{M}_2) \cup ((m_1 \times D_1) \times \{v\}) .\end{aligned}$$

Instead, we will need to modify d_m to eliminate the component of the support in $f_1(t_i) \times \mathcal{M}_2$. In the previous case we showed that the vector $d_{(u,m_2)}$ has $\{(u, m_2)\} \subseteq \mathsf{supp}(d_{(u,m_2)} L) \subseteq \{(u, m_2)\} \cup D$. Thus, we can set d'_m to be d_m minus an appropriate linear combination of vectors in $Q = \{d_{(u,m_2)} | u \in f_1(t_i), m_2 \in \mathcal{M}_2\}$ to obtain the desired support for $d'_m L$. Vectors in Q have support in $f_1(t_i) \times \mathcal{M}_2 = f(t_i)$, as needed.

C is ρ-certifiable

First, showing that Definition 7.2.1 goes through if $N(u)$ is replaced with clique $K(u)$ is identical to the proof above along with the observation that if K_1 and K_2 are cliques in N_1 and N_2 then $K_1 \times K_2$ is a clique in $N_1 \boxtimes N_2$.

It remains to show that $\mathsf{rank}(L^T I_M) \geq \rho$ for all multicuts M of $N_1 \boxtimes N_2$.

Notice that we can view the matrix L^T as having a block of rows for each $w \in V_1 \cup V_2$; the block of rows associated to $u \in V_1$ is $\mathbb{1}_u \otimes L_2^T$, and to $v \in V_2$ is $L_1^T \otimes \mathbb{1}_v$ (where $\mathbb{1}_u$ is the indicator row vector of u).

We will show that $\mathsf{rank}(L^T B) \geq \rho$ for a matrix B that is in the column space of I_M. This is sufficient because there is some linear transformation T such that $I_M T = B$, implying $\mathsf{rank}(L^T I_M) \geq \mathsf{rank}(L^T I_M T) = \mathsf{rank}(L^T B) \geq \rho$.

The matrix B will have r_1 columns for each $v \in V_2$ and r_2 columns for each $u \in V_1$. Let M_u, $u \in V_1$ be the multicut of $\{u\} \times V_2$ satisfying the conditions of Lemma 7.3.5 using the clique $K_1(u)$ that shows certifiability, and similarly for $M_v, v \in V_2$. The matrix B has a block of columns equal to $a_u^T \otimes I_{M_u}$ for each $u \in V_1$, and $I_{M_v} \otimes a_v^T$ for $v \in V_2 \setminus f_2(\mathcal{S}_2)$ where a_u and a_v are the vectors satisfying Definition 7.2.1 with cliques $K_1(u)$ and $K_2(v)$. The matrix B lies in the column space of I_M because a_u and a_v have support within their corresponding cliques and $K_1(u) \times M_u \subseteq M$, $M_v \times K_2(v) \subseteq M$.

We will show that the matrix $L^T B$ is lower block triangular with n_1 diagonal blocks of rank at least ρ_2 and $n_2 - |f_2(\mathcal{S}_2)|$ diagonal blocks of rank at least ρ_1. Row blocks of $L^T B$ are indexed by $w \in V_1 \cup V_2$ and column blocks are indexed by $w \in V_1 \cup V_2 \setminus f_2(\mathcal{S}_2)$. We will assume that the blocks are ordered according to $-\pi_1$ and $-\pi_2$ and blocks associated to elements of V_1 precede those of V_2.

The analysis of the blocks in the product matrix can be split into four cases. We only need to analyze three of the cases for purposes of showing the matrix is lower block triangular because the blocks that fall into the last case only appear in the lower right of the product matrix.

Block $[u, u']$, $u, u' \in V_1$:

$$\begin{aligned} L^T B[u, u'] &= (\mathbb{1}_u \otimes L_2^T)(a_{u'}^T \otimes I_{M_{u'}}) \\ &= \mathbb{1}_u a_{u'}^T \otimes L_2^T I_{M_{u'}} \end{aligned}$$

Thus, block $[u', u]$ has rank at least ρ_2 if $u \in \mathsf{supp}(a_{u'}) \subseteq K_1(u')$ and is identically zero otherwise. In particular, it is zero whenever $\pi_1(u) > \pi_1(u')$ because $u \in K_1(u') \implies \pi_1(u) \leq \pi_1(u')$.

Block $[v, v']$, $v, v' \in V_2 \setminus f_2(\mathcal{S}_2)$:

$$\begin{aligned} L^T B[v, v'] &= (L_1^T \otimes \mathbb{1}_v)(I_{M_{v'}} \otimes a_{v'}^T) \\ &= L_1^T I_{M_{v'}} \otimes \mathbb{1}_v a_{v'}^T \end{aligned}$$

Just as for block $[u, u']$, block $[v, v']$ has rank at least ρ_1 if $v \in \mathsf{supp}(a_{v'})$ and is zero otherwise.

Block $[u, v]$, $u \in V_1, v \in V_2 \setminus f_2(\mathcal{S}_2)$:

$$\begin{aligned} L^T B[u, v] &= (\mathbb{1}_u \otimes L_2^T)(I_{M_v} \otimes a_v^T) \\ &= \mathbb{1}_u I_{M_v} \otimes L_2^T a_v^T = 0 \end{aligned}$$

The last equality holds because $v \notin f_2(\mathcal{S}_2)$ implies $f_2^{-1}(v) = \emptyset$ and thus $\mathcal{M}(f_2^{-1}(v)) = \emptyset$, giving $L_2^T a_v = 0$.

The first two cases above, along with the ordering of blocks so that larger π values are on the top left, implies that the top left and lower right quadrants of the matrix $L^T B$ are lower block triangular with the required ranks on the diagonal blocks. The final case implies that the top right quadrant is all zero, as wanted. □

Proof of Lemma 7.3.4. Let $D_1, \ldots, D_l$ be the diagonal blocks of the matrix with ranks $r_1, \ldots, r_l$ respectively. We can convert the matrix to the identity matrix starting with the top left diagonal block D_1. First we apply steps of Gaussian elimination that convert D_1 to the identity of size r_1, possibly with additional rows or columns of all zeros. We delete the zero rows and columns of D_1 from the entire matrix. Then we subtract rows of $D_1 = I_{r_1}$ from the rest of the matrix so that the only non-zero terms in the first r_1 columns are contained in D_1. Notice that the lower block triangular property implies that all of the preceding row operations only change the first r_1 columns. We continue in this fashion for $D_2, \ldots, D_l$. At the end we are left with an identity matrix of size $\sum_{i=1}^{l} r_i$, implying that our original matrix has a submatrix of rank at least $\sum_{i=1}^{l} r_i$. □

Proof of Lemma 7.3.5. Suppose for contradiction that there is a multicut M of $N_1 \boxtimes N_2$

and some $u \in V_1$ such that for any multicut M_2 of N_2 there is at least one vertex $(a, b) \in K_1(u) \times M_2, (a, b) \notin M$. Let $C = \{v \in V_2 | K_1(u) \times v \subseteq M\}$. By assumption, C is not a multicut of N_2, and there exists a source-sink path in N_2 that does not intersect with C. Let $p_1 \ldots p_l$ be such a path. For each vertex $v \in V_2 \setminus C$, let $g(v) = (a, v)$ such that $a \in K_1(u), (a, v) \notin M$. Such a vertex must exist by definition of C. The path $g(p_1) \ldots g(p_l)$ is a source-sink path in $N_1 \boxtimes N_2$ that does not intersect M, a contradiction. □

7.4 Open Questions

In this work we give a class of network codes that provide lower bounds on the multicut. There are many potential directions to expand this class. For example, it may be possible to allow for edge-capacitated graphs or arbitrary capacities, or relax the condition of certifiability by strengthening Lemma 7.3.5. In networks of Saks *et al.* we show the coding rate exactly matches the multicut, despite the flow being a factor k smaller. We know a simple example where the network coding rate is less than the multicut, but we have no example eliminating the possibility that just two times the network coding rate is always at least the multicut. In general, does there exist some parameter κ that is $o(k)$ such that the coding rate scaled up by κ is always at least the size of the minimum multicut? This work focused on the multicut problem and maximum multicommodity flow variant of network coding but did not touch upon the related sparsest cut and concurrent multicommodity flow. Do similar results hold in that regime?

APPENDIX A

FANO AND NON-FANO INEQUALITIES

This appendix is devoted to deriving the Fano inequality that holds for any subspaces of a vector space over $\mathbb{F}$ such that $\text{char}(\mathbb{F}) = 2$, specified in Definition 2.3.15 and the non-Fano inequality that holds for any subspaces of a vector space over $\mathbb{F}$ such that $\text{char}(\mathbb{F}) \neq 2$, specified in Definition 2.3.16.

We derive the inequalities using the Fano matroid, $\mathcal{F}$, and non-Fano matorid as defined in Definition 5.2.2. We index the elements with $\mathcal{U} = \{100, 010, 001, 110, 101, 011, 111\}$ to index the elements of the two matroids. Further let $\mathcal{O} \subset \mathcal{U}$ be the vectors with odd Hamming weight, let $\mathcal{B}$ be the vectors with Hamming weight one and let $i+j$ for $i, j \in \mathcal{U}$ be the bitwise addition of i, j.

Before explaining how we derive these constraints, we introduce a bit of notation. If $\{V_i\}_{i\in I}$ are subspaces of a vector space V, let the span of V_i and V_j be denoted $V_i + V_j$ and let $\dim(\{V_i\}_{i\in I})$ be the dimension of the span of $\{V_i\}_{i\in I}$. Also, let $\vec{\mathbf{d}}(\{V_i\}_{i\in I})$ be a $2^{|I|}$ dimensional vector indexed by the subsets of I such that the coordinate indexed by S is $\dim(\{V_i\}_{i\in S})$. We let $V_1 \oplus \cdots \oplus V_k$ denote the sum of mutually complementary subspaces $V_1, \ldots, V_k$. If $V = V_1 \oplus \cdots \oplus V_k$ then V is isomorphic to the vector space $\prod_{i=1}^{k} V_i$ via the mapping $(v_1, \ldots, v_k) \mapsto v_1 + \cdots + v_k$. In this case, for an index set $S \subseteq \{1, \ldots, k\}$, we will use π_S to denote the projection function $V \to \oplus_{i\in S} V_i$, i.e. the function that maps an element $v = \sum_{i=1}^{k} v_i$ to the element $\pi_S(v) = \sum_{i\in S} v_i$.

Now, we derive our inequalities in a sequence of four steps. We will go through the steps by giving a series of lemmas and theorems. The proofs of these results are in the sections that follow.

The fact that the Fano matroid can be represented over $\mathbb{F}_2$ and the non-Fano matroid cannot tells us something about dimension dependencies that can occur in $\mathbb{F}_2$. In the first step we extract the critical dimension relations that distinguish vector spaces over $\mathbb{F}$ with $\operatorname{char}(\mathbb{F}) = 2$, as described in the following lemma.

Lemma A.0.1. *Let $V = V_1 \oplus V_2 \oplus V_3$ be a vector space over a field $\mathbb{F}$, and suppose $W \subset V$ is a linear subspace that is complementary to each of $V_1 \oplus V_2, V_1 \oplus V_3, V_2 \oplus V_3$. Then*

$$\dim\left(\pi_{12}(W), \pi_{13}(W), \pi_{23}(W)\right) = \begin{cases} 2\dim(W) & \textit{if } \operatorname{char}(\mathbb{F}) = 2 \\ 3\dim(W) & \textit{if } \operatorname{char}(\mathbb{F}) \neq 2. \end{cases} \tag{A.1}$$

Next, using Lemma A.0.1 we derive two dimension inequalities, one for even characteristic that will become our Fano inequality, and one for odd characteristic that will become our non-Fano inequality. But rather than being universally valid for any dimension vector over a certain field, these inequalities only hold if some conditions hold on the dimensions of certain subsets.

Lemma A.0.2 (Conditional Even Characteristic Inequality)**.** *Suppose $\{V_i\}_{i \in \mathcal{U}}$ are 7 subspaces of a vector space over $\mathbb{F}$ such that $\operatorname{char}(\mathbb{F}) = 2$ and*

(i) $\dim(\{V_i\}_{i\in\mathcal{O}}) = \dim(\{V_i\}_{i\in\mathcal{B}})$

(ii) $\dim(V_i, V_j, V_k) = \dim(V_i) + \dim(V_j) + \dim(V_k)\ \forall i, j, k \in \mathcal{O}$

(iii) $\dim(V_i, V_j, V_{i+j}) = \dim(V_i, V_j)\ \forall i, j \in \mathcal{O}$

Then $\dim(V_{110}, V_{101}, V_{011}) \leq 2\dim(V_{111})$.

Lemma A.0.3 (Conditional Odd Characteristic Inequality)**.** *Suppose $\{V_i\}_{i \in \mathcal{U}}$ are 7 subspaces of a vector space over $\mathbb{F}$ such that $\operatorname{char}(\mathbb{F}) \neq 2$ and*

(i) $\dim(\{V_i\}_{i\in\mathcal{O}}) = \dim(\{V_i\}_{i\in\mathcal{B}})$

(ii) $\dim(V_i, V_j, V_k) = \dim(V_i) + \dim(V_j) + \dim(V_k)\ \forall i, j, k \in \mathcal{O}$

(iii) $\dim(V_i, V_j, V_{i+j}) = \dim(V_i, V_j)\ \forall i, j \in \mathcal{B}$

(iv) $\dim(V_i, V_j, V_{111}) = \dim(V_i, V_j)\ \forall i, j : i + j = 111$

Then $\dim(V_{110}, V_{101}, V_{011}) \geq 3\dim(V_{111})$.

The third step is to transform the conditional inequalities given in the lemmas above to general inequalities that apply to any 7 subspaces of a vector space over a field of even (resp. odd) characteristic by using the following approach. We will start with arbitrary subspaces and then repeatedly modify them until they satisfy the conditions of Lemma A.0.2. At that point the result in the conditional lemma will imply an inequality involving the dimensions of the modified subspaces, which we will express in terms of the dimensions of the original subspaces.

Theorem A.0.4 (Even Characteristic Inequality)**.** *There exists a* 2^7*-dimensional vector* Λ_{even} *such that for any 7 subspaces* $\{V_i\}_{i\in\mathcal{U}}$ *of a vector space over* $\mathbb{F}$ *with* $\text{char}(\mathbb{F}) = 2$*,*

$$\Lambda_{\text{even}} \cdot \vec{\mathbf{d}}(\{V_i\}_{i\in\mathcal{U}}) \geq 0 \ \textit{and}\ \Lambda_{\text{even}} \cdot \vec{\mathbf{r}}(\mathcal{N}) < 0.$$

Theorem A.0.5 (Odd Characteristic Inequality)**.** *There exists a* 2^7*-dimensional vector* Λ_{odd} *such that for any 7 subspaces* $\{V_i\}_{i\in\mathcal{U}}$ *of a vector space over* $\mathbb{F}$ *with* $\text{char}(\mathbb{F}) \neq 2$*,*

$$\Lambda_{\text{odd}} \cdot \vec{\mathbf{d}}(\{V_i\}_{i\in\mathcal{U}}) \geq 0 \ \textit{and}\ \Lambda_{\text{odd}} \cdot \vec{\mathbf{r}}(\mathcal{F}) < 0\,.$$

Finally, we convert the inequalities so that the corresponding constraint schema are tight and thus can be part of a LP that is super-multiplicative. The following lemma shows how to take a single linear dimension inequality, such as one of those whose existence is asserted by Theorems A.0.4 and A.0.5, and transform it into a tight inequality. Recall that $\Upsilon_n^{\mathbb{F}} \subset \mathbb{R}^{2^n}$ for any index set K of size n and field $\mathbb{F}$, is the set of all vectors $\vec{\mathbf{d}}(\{V_k\}_{k\in K})$, where $\{V_k\}_{k\in K}$ runs through all K-indexed tuples of finite-dimensional vector spaces over $\mathbb{F}$.

Lemma A.0.6 (Tightening Modification)**.** *Suppose I is any index set, e is an element not in I, and $J = I \cup \{e\}$. There exists an explicit linear transformation from $\mathbb{R}^{\mathcal{P}(J)}$ to $\mathbb{R}^{\mathcal{P}(I)}$, represented by a matrix B, such that:*

(i) $B \cdot \Upsilon^{\mathbb{F}}_{|J|} \subseteq \Upsilon^{\mathbb{F}}_{|I|}$ *for every field* $\mathbb{F}$.

(ii) $B\mathbf{1} = B\mathbf{1}_j = 0$ *for all* $j \in J$.

(iii) If M is a matroid with ground set I and the intersection of all matroid bases of M is the empty set, then $B\vec{\mathbf{r}}(M + e) = \vec{\mathbf{r}}(M)$, where $M + e$ denotes the matroid obtained by adjoining a rank-zero element to M.

Now, applying the tightening modification lemma to the inequalities Λ_{odd} and Λ_{even} gives our final theorem the Fano and non-Fano inequalities we use to get LP bounds.

Theorem A.0.7. *Let $\vec{\alpha}_{\mathcal{F}}$ be the Fano inequality given in Definition 2.3.15. For $\mathbb{F}$ such that* $\operatorname{char}(\mathbb{F}) = 2$, *$\vec{\alpha}_{\mathcal{F}} \cdot \vec{d} \geq 0$ for $d \in \Upsilon^{\mathbb{F}}_8$. Moreover, $\vec{\alpha}^T_{\mathcal{F}}\mathbf{1} = \vec{\alpha}^T_{\mathcal{F}}\mathbf{1}_j = 0$ for all $i \in \{1, \ldots, 8\}$ and the rank vector of the matroid $\mathcal{N}$ violates the inequality: $\vec{\alpha}_{\mathcal{F}} \cdot \vec{\mathbf{r}}(\mathcal{N} + e) < 0$, where $\mathcal{N} + e$ is the non-Fano matroid adjoined to the rank zero element e.*

Theorem A.0.8. *Let $\vec{\alpha}_{\mathcal{N}}$ be the non-Fano inequality given in Definition 2.3.16. For $\mathbb{F}$ such that* $\operatorname{char}(\mathbb{F}) \neq 2$, *$\vec{\alpha}^T_{\mathcal{N}} \cdot \vec{d} \geq 0$ for $d \in \Upsilon^{\mathbb{F}}_8$. Moreover, $\vec{\alpha}^T_{\mathcal{N}}\mathbf{1} = \vec{\alpha}_{\mathcal{F}}\mathbf{1}_j = 0$ for all $i \in \{1, \ldots, 8\}$ and the rank vector of the matroid $\mathcal{F}$ violates the inequality: $\vec{\alpha}_{\mathcal{N}} \cdot \vec{\mathbf{r}}(\mathcal{F} + e) < 0$, where $\mathcal{F} + e$ is the non-Fano matroid adjoined to the rank zero element e.*

A.1 Proof of Lemma A.0.1

Proof of Lemma A.0.1. Recalling that V is isomorphic to $\prod_{i=1}^{3} V_i$, we will write elements of V as ordered triples. Our assumption that W is complementary to each of $V_1 \oplus V_2, V_1 \oplus$

$V_3, V_2 \oplus V_3$ implies that a nonzero element of W has three nonzero coordinates, a fact that we will use in both cases of the lemma.

If $\mathrm{char}(\mathbb{F}) = 2$, then every vector $(x, y, z) \in V$ satisfies

$$\pi_{12}(x, y, z) + \pi_{13}(x, y, z) = (x, y, 0) + (x, 0, z) = (0, y, z) = \pi_{23}(x, y, z)$$

hence $\pi_{12}(W) + \pi_{13}(W) = \pi_{23}(W)$. Consequently

$$\dim\left(\pi_{12}(W), \pi_{13}(W), \pi_{23}(W)\right) = \dim\left(\pi_{12}(W), \pi_{13}(W)\right) \leq 2\dim(W).$$

To prove the reverse inequality we observe that $\pi_{12}(W)$ and $\pi_{13}(W)$ are complementary, since every nonzero element of $\pi_{12}(W)$ is of the form $(x, y, 0)$ with $x, y \neq 0$, whereas every nonzero element of $\pi_{13}(W)$ is of the form $(x, 0, z)$ with $x, z \neq 0$, and hence $\pi_{12}(W) \cap \pi_{13}(W) = \{0\}$.

When $\mathrm{char}(\mathbb{F}) \neq 2$, we prove Equation (A.1) by showing that $\pi_{12}(W), \pi_{13}(W), \pi_{23}(W)$ are mutually complementary. Consider any three vectors $w_1 = (x_1, y_1, z_1)$, $w_2 = (x_2, y_2, z_2)$, and $w_3 = (x_3, y_3, z_3)$, all belonging to W, such that

$$0 = \pi_{23}(x_1, y_1, z_1) + \pi_{13}(x_2, y_2, z_2) + \pi_{12}(x_3, y_3, z_3) = (x_2 + x_3, y_1 + y_3, z_1 + z_2)\,.$$

This implies that $x_2 + x_3 = 0$, so the first coordinate of $w_2 + w_3$ is zero. However, the zero vector is the only vector in W whose first coordinate is zero, hence $w_2 + w_3 = 0$. Similarly, $w_1 + w_3 = 0$ and $w_1 + w_2 = 0$. Now using the fact that 2 is invertible in $\mathbb{F}$, we deduce that $w_1 = \frac{1}{2}[(w_1 + w_2) + (w_1 + w_3) - (w_2 + w_3)] = 0$, and similarly $w_2 = 0$ and $w_3 = 0$. Thus, the only way to express the zero vector as a sum of vectors in $\pi_{12}(W), \pi_{13}(W), \pi_{23}(W)$ is if all three summands are zero, i.e. those three subspaces are mutually complementary as claimed. □

A.2 Proofs of Conditional Inequalities

Proof of Lemma A.0.2. Hypotheses (i) and (iii) of the lemma imply that all 7 subspaces are contained in the span of $V_{100}, V_{010}, V_{001}$. Moreover, hypothesis (ii) implies that $V_{100}, V_{010}, V_{001}$ are mutually complementary and that V_{111} is complementary to each of $V_{100} + V_{010}$, $V_{100} + V_{001}$, $V_{010} + V_{001}$. Thus, we can apply Lemma A.0.1 with $V = V_{100} \oplus V_{010} \oplus V_{001}$ and $W = V_{111}$, yielding the equation $\dim(\pi_{12}(V_{111}), \pi_{23}(V_{111}), \pi_{13}(V_{111})) = 2\dim(V_{111})$.

We claim that $\pi_{12}(V_{111}) = (V_{001} + V_{111}) \cap (V_{100} + V_{010})$. To see this, take an arbitrary element $w \in V_{111}$ having a unique representation of the form $x + y + z$ with $x \in V_{100}, y \in V_{010}, z \in V_{001}$. By definition $\pi_{12}(w) = x + y = w - z$, from which it can be seen at once that $\pi_{12}(w)$ belongs to both $V_{100} + V_{010}$ and $V_{001} + V_{111}$. Conversely, any element $v \in (V_{001} + V_{111}) \cap (V_{100} + V_{010})$ can be expressed as $v = w - z$ where $w \in V_{111}, z \in V_{001}$ but it can also be expressed as $v = x + y$ where $x \in V_{100}, y \in V_{010}$. Consequently, $w = x + y + z$ and $v = \pi_{12}(w)$.

Hypothesis (iii) implies that V_{110} is contained in both $V_{001} + V_{111}$ and $V_{100} + V_{010}$, hence $V_{110} \subseteq \pi_{12}(V_{111})$. Similarly $V_{101} \subseteq \pi_{13}(V_{111})$ and $V_{011} \subseteq \pi_{23}(V_{111})$. Hence $\dim(V_{110}, V_{101}, V_{011}) \leq \dim(\pi_{12}(V_{111}), \pi_{23}(V_{111}), \pi_{13}(V_{111})) = 2\dim(V_{111})$, as desired. □

Proof of Lemma A.0.3. Just as in the proof of Lemma A.0.2 we apply the result of Lemma A.0.1, but now with $\mathrm{char}(\mathbb{F}) \neq 2$. Hypotheses (i) and (iii) imply that all 7 subspaces are contained in the span of $V_{100}, V_{010}, V_{001}$, and hypothesis (ii) implies that those three subspaces are mutually complementary, and that V_{111} is complementary to the sum of any two of them. Thus, Lemma A.0.1 implies that $\dim(\pi_{12}(W), \pi_{23}(W), \pi_{13}(W)) = 3\dim(W)$. Now we aim to show that hypotheses (iii) and (iv) imply that V_{110} contains $\pi_{12}(V_{111})$, and similarly for V_{101}, V_{011}. This will imply that $\dim(V_{110}, V_{101}, V_{011}) \geq \dim(\pi_{12}(W), \pi_{23}(W), \pi_{13}(W)) = 3\dim(W)$ as desired.

It remains for us to justify the claim that V_{110} contains $\pi_{12}(V_{111})$. Suppose (x, y, z) belongs to V_{111}, where we use (x, y, z) as an alternate notation for $x + y + z$ such that x belongs to V_{100}, y belongs to V_{010}, z belongs to V_{001}. We know from hypothesis (iv) that V_{111} is contained in $V_{001} + V_{110}$. So write $x + y + z = a + b$ where a is in V_{001} and b is in V_{110}. We know from hypothesis (iii) that V_{110} is contained in $V_{100}+V_{010}$, so write $b = c+d$ where c is in V_{100} and d is in V_{010}. Then $x+y+z = c+d+a$, and both sides are a sum of three vectors, the first belonging to V_{100}, the second to V_{010}, the third to V_{001}. Since those three vector spaces are mutually complementary, the representation of another vector as a sum of vectors from each of them is unique. So $x = c, y = d, z = a$. This means that $x + y = c + d = \pi_{12}(x, y, z)$. Recall that $c + d$ is in V_{110}. As (x, y, z) was an arbitrary element of V_{111}, we have shown that V_{110} contains $\pi_{12}(V_{111})$. □

A.3 Proofs of Unconditional Inequalities

Proof of Theorem A.0.4. As mentioned above, the proof will proceed by repeatedly modifying the input subspaces until they satisfy the requirements of Lemma A.0.2. The modifications we make to a vector space are of one type: we *delete* a vector w from a subspace V that contains w, by letting B be a basis of V containing w and then replacing V with the span of $B \setminus w$.

Let $\{V_i\}_{i\in\mathcal{U}}$ be seven subspaces of a vector space V over $\mathbb{F}$ such that $\text{char}(\mathbb{F}) = 2$. We will modify the subspaces $\{V_i\}_{i\in\mathcal{U}}$ into $\{V_i'\}_{i\in\mathcal{U}}$ that satisfy the conditions of Lemma A.0.2. To start, we set $\{V_i'\}_{i\in\mathcal{U}} = \{V_i\}_{i\in\mathcal{U}}$. We then update $\{V_i'\}_{i\in\mathcal{U}}$ in three steps, each of which deletes vectors of a certain type in an iterative fashion. The order of the deletions within each step is arbitrary.

Step 1: Vectors in V_{111}' but not in $\sum_{i\in\mathcal{B}} V_i'$ from V_{111}'.

Step 2: (a) Vectors in $V'_{100} \cap V'_{010}$ from V'_{010}.

(b) Vectors in $V'_{001} \cap (V'_{100} + V'_{010})$ from V'_{001}.

(c) Vectors in $V'_{111} \cap (V'_{100} + V'_{010})$ from V'_{111}.

(d) Vectors in $V'_{111} \cap (V'_{010} + V'_{001})$ from V'_{111}.

(e) Vectors in $V'_{111} \cap (V'_{100} + V'_{001})$ from V'_{111}.

Step 3: Vectors in V'_{i+j} but not in $V'_i + V'_j$ for $i, j \in \mathcal{O}$ from V'_{i+j}.

First, we argue that $\{V'_i\}_{i \in \mathcal{U}}$ satisfy the conditions of Lemma A.0.2. The deletions in step (1) ensure that V'_{111} is contained in $\sum_{i \in \mathcal{B}} V'_i$, thus satisfying condition (i). The deletions in steps (2a)–(2b) ensure that $V'_{100}, V'_{010}, V'_{001}$ are mutually complementary, and steps (2c)–(2d) ensure that V'_{111} is complementary to the sum of any two of them, thus satisfying condition (ii). Furthermore, step (2) does not change $\sum_{i \in \mathcal{B}} V'_i$ because we only delete a vector from one of $\{V'_i\}_{i \in \mathcal{B}}$ when it belongs to the span of the other two. Thus condition (i) is still satisfied at the end of step (2). Step (3) ensures that V'_{i+j} is contained in $V'_i + V'_j$, thus satisfying condition (iii). Furthermore, it does not modify $V'_i, i \in \mathcal{O}$, and thus conditions (i) and (ii) remain satisfied after step (3).

Now, by Lemma A.0.2 we have that

$$\dim(V'_{110}, V'_{101}, V'_{011}) \leq 2 \dim(V'_{111}). \tag{A.2}$$

Let

$$\begin{aligned}
\delta &= \dim(V_{111}, \{V_i\}_{i \in \mathcal{B}}) - \dim(\{V_i\}_{i \in \mathcal{B}}) \\
\delta[i|j,k] &= \dim(V_i, V_j, V_k) - \dim(V_j, V_k) \\
\delta[i;j] &= \dim(V_i \cap V_j) = \dim(V_i) + \dim(V_j) - \dim(V_i, V_j) \\
\delta[i;j,k] &= \dim(V_i \cap (V_j + V_k)) = \dim(V_i) + \dim(V_j, V_k) - \dim(V_i, V_j, V_k)
\end{aligned}$$

Observe that after step (1) $\dim(V'_{111}) = \dim(V_{111}) - \delta$, and steps (2) and (3) only delete more vectors from V'_{111}, so we have $\dim(V'_{111}) \leq \dim(V_{111}) - \delta$.

It remains to get a lower bound on $\dim(V'_{110}, V'_{101}, V'_{011})$ in terms of dimensions of subsets of $\{V_i\}_{i \in \mathcal{U}}$. We do this by giving an upper bound on the total number of vectors deleted from $E = V'_{110} + V'_{101} + V'_{011}$ in terms of the δ terms we defined above. In steps (1) and (2) we delete nothing from E, but we delete some vectors from $V'_i, i \in \mathcal{O}$. Specifically, $\delta[100; 010]$ vectors are deleted from V'_{010}, $\delta[001; 100, 010]$ vectors are deleted from V'_{001}, and no vectors are deleted from V_{100}. As already noted, step (1) deletes δ vectors from V'_{111}, while step (2) deletes at most $\sum_{i,j \in \mathcal{B}} \delta[111; i, j]$ vectors from V'_{111}. To summarize, the dimensions of $V'_i, i \in \mathcal{O}$, after steps (1) and (2), satisfy:

$$\dim(V'_{100}) = \dim(V_{100}) \tag{A.3}$$

$$\dim(V'_{010}) = \dim(V_{010}) - \delta[100; 010] \tag{A.4}$$

$$\dim(V'_{001}) = \dim(V_{001}) - \delta[001; 100, 010] \tag{A.5}$$

$$\dim(V'_{111}) \geq \dim(V_{111}) - \delta - \sum_{i,j \in \mathcal{B}} \delta[111; i, j]. \tag{A.6}$$

In step (3), when we delete vectors in V'_{i+j} but not in $V'_i + V'_j$; if no deletions had taken place in prior steps then the number of vectors deleted from V'_{i+j} would be $\delta[i + j|i, j]$. However, the deletions that took place in steps (1) and (2) have the effect of reducing the dimension of $V'_i + V'_j$, and we must adjust our upper bound on the number of vectors deleted from V'_{i+j} to account for the potential difference in dimension between $V_i + V_j$ and $V'_i + V'_j$. When $i = 100$, $j = 010$, there is no difference between $V_i + V_j$ and $V'_i + V'_j$, because the only time vectors are deleted from either one of these subspaces is in step (2a), when vectors in $V'_{100} \cap V'_{010}$ are deleted from V'_{010} without changing the dimension of $V'_{100} + V'_{010}$. For all other pairs $i, j \in \mathcal{O}$, we use the upper bound

$$\dim(V_i + V_j) - \dim(V'_i + V'_j) \leq [\dim(V_i) - \dim(V'_i)] + \left[\dim(V_j) - \dim(V'_j)\right],$$

which is valid for any four subspaces V_i, V_j, V'_i, V'_j satisfying $V'_i \subseteq V_i$, $V'_j \subseteq V_j$. Let $\Delta\dim(V_i)$

denote the difference $\dim(V_i) - \dim(V_i')$. Combining these upper bounds, we find that the number of extra vectors deleted from E in step (3) because of differences in dimension between $V_i' + V_j'$ and $V_i + V_j$ is at most

$$\begin{aligned}\left(\sum_{i,j\in\mathcal{O}} \Delta\dim(V_i) + \Delta\dim(V_j)\right) &- \Delta\dim(V_{100}) - \Delta\dim(V_{010}) \\ &= 2\left(\sum_{i\in\{100,010\}} \Delta\dim(V_i)\right) + 3\left(\sum_{i\in\{001,111\}} \Delta\dim(V_i)\right) \\ &\leq 2\delta[100;010] + 3\delta[001;100,010] + 3\delta + 3\sum_{i,j\in\mathcal{B}} \delta[111;i,j]\end{aligned}$$

where the last inequality follows by combining equations (A.3)–(A.6).

We now sum up our upper bounds on the number of vectors deleted from E in step (3), to find that

$$\dim(E) \geq \dim(V_{110}, V_{101}, V_{011}) - \sum_{i,j\in\mathcal{O}} \delta[i+j|i,j] - 2\delta[100;010] - 3\delta[001;100,010] - 3\delta - 3\sum_{i,j\in\mathcal{B}} \delta[111;i,j]. \tag{A.7}$$

Expanding out all the δ terms, combining with the upper bound $\dim(V_{111}') \leq \dim(V_{111}) - \delta$, and plugging these into Equation (A.2) gives us $\Lambda_{\text{even}} \cdot \vec{\mathbf{d}}(\{V_i\}_{i\in\mathcal{U}}) \geq 0$ for some 2^7-dimensional vector Λ_{even}, as desired; after applying these steps one obtains Equation (A.8) below. When $\{V_i\}_{i\in\mathcal{U}}$ are one-dimensional subspaces constituting a representation of the non-Fano matroid over a field of characteristic $\neq 2$, it is easy to check that all of the δ terms appearing in (A.7) are zero. So, the inequality states that $\dim(V_{110}, V_{101}, V_{011}) \leq 2\dim(V_{111})$, whereas we know that $\dim(V_{110}, V_{101}, V_{011}) = 3\dim(V_{111})$ for the non-Fano matroid. Consequently $\Lambda_{\text{odd}} \cdot \vec{\mathbf{r}}(\mathcal{F}) < 0$.

For completeness, the inequality $\Lambda_{\text{even}} \cdot \vec{\mathbf{d}}(\{V_i\}_{i\in\mathcal{U}}) \geq 0$ is written explicitly as follows.

$$\begin{aligned}
&2\dim(V_{100}) + 2\dim(V_{010}) + 3\dim(V_{001}) + 11\dim(V_{111}) \\
&+3\dim(V_{100}, V_{010}) + 2\dim(V_{100}, V_{001}) + 2\dim(V_{010}, V_{001}) \\
&-\dim(V_{100}, V_{111}) - \dim(V_{010}, V_{111}) - \dim(V_{001}, V_{111}) - 4\dim(V_{100}, V_{010}, V_{001}) \\
&-3\dim(V_{111}, V_{100}, V_{010}) - 3\dim(V_{111}, V_{100}, V_{001}) - 3\dim(V_{111}, V_{010}, V_{001}) \\
&+\dim(V_{110}, V_{100}, V_{010}) + \dim(V_{101}, V_{100}, V_{001}) + \dim(V_{011}, V_{010}, V_{001}) \\
&+\dim(V_{110}, V_{111}, V_{001}) + \dim(V_{101}, V_{111}, V_{010}) + \dim(V_{011}, V_{111}, V_{100}) \\
&-\dim(V_{110}, V_{101}, V_{011}) + \dim(V_{111}, V_{100}, V_{010}, V_{001}) \geq 0\,. \qquad (A.8)
\end{aligned}$$

This concludes the proof of the theorem. □

Proof of Theorem A.0.5. Let $\{V_i\}_{i\in\mathcal{U}}$ be seven subspaces of a vector space V over $\mathbb{F}$ such that $\text{char}(\mathbb{F}) \neq 2$. Just as in the proof Theorem A.0.4, we will modify the subspaces $\{V_i\}_{i\in\mathcal{U}}$ into $\{V_i'\}_{i\in\mathcal{U}}$ that satisfy the conditions of Lemma A.0.3, starting with $\{V_i'\}_{i\in\mathcal{U}} = \{V_i\}_{i\in\mathcal{U}}$. We again delete vectors of a certain type in an iterative fashion. The order of the deletions within each step is arbitrary.

Step 1: Vectors in V'_{111} but not in $\sum_{i\in\mathcal{B}} V_i'$ from V'_{111}.

Step 2: (a) Vectors in $V'_{100} \cap V'_{010}$ from V'_{010}.

(b) Vectors in $V'_{001} \cap (V'_{100} + V'_{010})$ from V'_{001}.

(c) Vectors in $V'_{111} \cap (V'_{100} + V'_{010})$ from V'_{111}.

(d) Vectors in $V'_{111} \cap (V'_{010} + V'_{001})$ from V'_{111}.

(e) Vectors in $V'_{111} \cap (V'_{100} + V'_{001})$ from V'_{111}.

Step 3: Vectors in V'_{i+j} but not in $V'_i + V'_j$ for $i, j \in \mathcal{B}$ from V'_{i+j}.

Step 4: Vectors in V'_{111} but not in $V'_i + V'_j$ for $i, j : i + j = 111$ from V'_{111}.

The first two steps in this sequence of deletions, along with the first two conditions in Lemma A.0.3 are identical to those in the even characteristic case. Thus, by arguments from the proof Theorem A.0.4 we have that by the end of step (2) conditions (i), (ii) are satisfied. Step (3) is almost identical to the same step in the even characteristic case; the difference is that now we only perform the step for pairs $i, j \in \mathcal{B}$ rather than all pairs $i, j \in \mathcal{O}$. As before, at the end of step (3) condition (iii) is satisfied, and since the step does not modify V_i' for any $i \in \mathcal{O}$, it does not cause either of conditions (i), (ii) to become violated. Step (4) ensures condition (iv), so it remains to show that step (4) preserves conditions (i)–(iii). Step (4) only modifies V_{111}' so it doesn't change $\sum_{i\in\mathcal{B}} V_i'$, therefore preserving (i). It preserves (ii) because if three subspaces are mutually complementary, they remain mutually complementary after deleting a vector from one of them. It preserves (iii) because (iii) does not involve V_{111}', which is the only subspace that changes during step (4).

Now, by Lemma A.0.2 we have that

$$3\dim(V_{111}') \leq \dim(V_{110}', V_{101}', V_{011}'). \tag{A.9}$$

As in the proof of Theorem A.0.4, let

$$\begin{aligned}
\delta &= \dim(V_{111}, \{V_i\}_{i\in\mathcal{B}}) - \dim(\{V_i\}_{i\in\mathcal{B}}) \\
\delta[i|j,k] &= \dim(V_i, V_j, V_k) - \dim(V_j, V_k) \\
\delta[i;j] &= \dim(V_i \cap V_j) = \dim(V_i) + \dim(V_j) - \dim(V_i, V_j) \\
\delta[i;j,k] &= \dim(V_i \cap (V_j + V_k)) = \dim(V_i) + \dim(V_j, V_k) - \dim(V_i, V_j, V_k)
\end{aligned}$$

Observe that we only reduce the size of subspaces, so $\dim(V_{110}', V_{101}', V_{011}') \leq \dim(V_{110}, V_{101}, V_{011})$.

It remains to get a lower bound on $\dim(V_{111}')$ in terms of dimensions of subsets of $\{V_i\}_{i\in\mathcal{U}}$. We do this by giving an upper bound on the number of vectors we delete from V_{111}' in terms of the δ terms we defined above. Step (1) deletes δ vectors. Steps (2a) and (2b) delete nothing from V'_{111}, and at the end of (2a)–(2b) we have

$$\dim(V'_{100}) = \dim(V_{100}) \tag{A.10}$$

$$\dim(V'_{010}) = \dim(V_{010}) - \delta[100; 010] \tag{A.11}$$

$$\dim(V'_{001}) = \dim(V_{001}) - \delta[001; 100, 010] \tag{A.12}$$

Steps (2c)–(2e) delete at most $\sum_{i,j\in\mathcal{B}} \delta[111; i, j]$ vectors from V'_{111}, and they do not change any of the other subspaces.

In step (3) no vectors are deleted from V'_{111}, but we will still need an upper bound on the number of vectors deleted in this step since it will influence our upper bound on the number of vectors deleted from V'_{111} in step (4). If no deletions took place prior to step (3), then for all $i, j \in \mathcal{B}$ exactly $\delta[i + j|i, j]$ vectors would be deleted from V'_{i+j} during step (3). However, if $\dim(V'_i, V'_j) < \dim(V_i, V_j)$, then we must adjust our estimate of the number of deleted vectors to account for this difference. Steps (1) and (2a) cannot change $\dim(V'_i, V'_j)$ for any $i, j \in \mathcal{B}$, but step (2b) reduces each of $\dim(V'_{001}, V'_{100})$ and $\dim(V'_{001}, V'_{010})$ by at most $\delta[001; 100, 010]$. Therefore, at the end of step (3) we have

$$\dim(V'_{110}) = \dim(V_{110}) - \delta[110|100, 010] \tag{A.13}$$

$$\dim(V'_{101}) \geq \dim(V_{101}) - \delta[101|100, 001] - \delta[001; 100, 010] \tag{A.14}$$

$$\dim(V'_{011}) \geq \dim(V_{011}) - \delta[011|010, 001] - \delta[001; 100, 010] \tag{A.15}$$

If no deletions took place prior to step (4), then the number of vectors we would need to delete from V'_{111}, to make it a subspace of $V'_i + V'_j$, would be at most $\delta[111|i, j]$. As before, we need to adjust this bound to account for the potential difference in dimension between $V_i + V_j$ and $V'_i + V'_j$. Using the upper bound

$$\dim(V_i + V_j) - \dim(V'_i + V'_j) \leq [\dim(V_i) - \dim(V'_i)] + \left[\dim(V_j) - \dim(V'_j)\right],$$

which is valid for any four subspaces V_i, V_j, V_i', V_j' satisfying $V_i' \subseteq V_i$, $V_j' \subseteq V_j$, we find that the number of extra vectors deleted from V_{111}' in step (4) because of differences in dimension between $V_i' + V_j'$ and $V_i + V_j$ (for some $i, j \in \mathcal{U}$, $i + j = 111$), is at most

$$\sum_{i \in \mathcal{U} \setminus \{111\}} \dim(V_i) - \dim(V_i') \leq \delta[100; 010] + 3\delta[001; 100, 010] + \sum_{i,j \in \mathcal{B}} \delta[i + j | i, j],$$

where the first inequality follows by combining equations (A.10)–(A.15).

We now sum up our upper bounds on the number of vectors deleted from V_{111}' in steps (1)–(4) combined, to find that

$$\dim(V_{111}') \geq \dim(V_{111}) - \delta - \sum_{i,j \in \mathcal{B}} \delta[111; i, j] - \delta[100; 010] - 3\delta[001; 100, 010] - \sum_{i,j \in \mathcal{B}} \delta[i + j | i, j]. \tag{A.16}$$

Expanding out all of the δ terms, combining with the upper bound on $\dim(V_{111}')$, and plugging these into Equation (A.9) gives us $\Lambda_{\text{odd}} \cdot \vec{\mathbf{d}}(\{V_i\}_{i \in \mathcal{U}}) \geq 0$ for some 2^7-dimensional vector Λ_{odd}, as desired; after applying these steps one obtains Equation (A.17) below. When $\{V_i\}_{i \in \mathcal{U}}$ are one-dimensional subspaces constituting a representation of the Fano matroid over a field of characteristic 2, it is easy to check that all of the δ terms appearing in (A.16) are zero. So, the inequality states that $\dim(V_{110}, V_{101}, V_{011}) \geq 3 \dim(V_{111})$, whereas we know that $\dim(V_{110}, V_{101}, V_{011}) = 2 \dim(V_{111})$ for the Fano matroid. Consequently $\Lambda_{\text{odd}} \cdot \vec{\mathbf{r}}(\mathcal{F}) < 0$.

For completeness, the inequality $\Lambda_{\text{odd}} \cdot \vec{\mathbf{d}}(\{V_i\}_{i \in \mathcal{U}}) \geq 0$ is written explicitly as follows.

$$\begin{aligned}
&3\dim(V_{100}) + 3\dim(V_{010}) + 9\dim(V_{001}) + 6\dim(V_{111}) + 6\dim(V_{100}, V_{010}) - 12\dim(V_{100}, V_{010}, V_{001})\\
&+3\dim(V_{110}, V_{100}, V_{010}) + 3\dim(V_{101}, V_{100}, V_{001}) + 3\dim(V_{011}, V_{010}, V_{001})\\
&-3\dim(V_{111}, V_{100}, V_{010}) - 3\dim(V_{111}, V_{100}, V_{001}) - 3\dim(V_{111}, V_{010}, V_{001})\\
&+3\dim(V_{111}, V_{100}, V_{010}, V_{001}) + \dim(V_{110}, V_{101}, V_{011}) \geq 0\,.
\end{aligned} \tag{A.17}$$

This completes the proof of the theorem. □

A.4 Tightening Modification

Proof of Lemma A.0.6. If U is any vector space with a J-tuple of subspaces $\{U_j\}_{j\in J}$, then there is a quotient map π from U to $V = U/U_e$, and we can form an I-tuple of subspaces $\{V_i\}_{i\in I}$ by specifying that $V_i = \pi(U_i)$ for all $i \in I$. The dimension vectors $\vec{\mathbf{u}} = \vec{\mathbf{d}}(\{U_j\})$ and $\vec{\mathbf{v}} = \vec{\mathbf{d}}(\{V_i\})$ are related by an explicit linear transformation. In fact, for any subset $S \subseteq I$, if we let U_S, V_S denote the subspaces of U, V spanned by $\{U_i\}_{i\in S}$ and $\{V_i\}_{i\in S}$, respectively, then π maps $U_S + U_e$ onto V_S with kernel U_e, and this justifies the formula

$$\mathbf{v}_S = \mathbf{u}_{S\cup\{e\}} - \mathbf{u}_{\{e\}}.$$

Thus, $\mathbf{v} = B_0\mathbf{u}$, where B_0 is the matrix

$$(B_0)_{ST} = \begin{cases} 1 & \text{if } T = S \cup \{e\} \\ -1 & \text{if } T = \{e\} \\ 0 & \text{otherwise,} \end{cases} \tag{A.18}$$

and therefore $B_0 \cdot \Upsilon^{\mathbb{F}}_{|J|} \subseteq \Upsilon^{\mathbb{F}}_{|I|}$.

Similarly, if U is any vector space with an I-tuple of subspaces $\{U_i\}_{i\in I}$ and k is any element of I, we can define $U_{-k} \subseteq U$ to be the linear subspace spanned by $\{U_i\}_{i\neq k}$, and we can let $\pi : U \to U_{-k}$ be any linear transformation whose restriction to U_{-k} is the identity and $\pi(U_k) = U_k \cap U_{-k}$. The restriction of π to U_k has kernel W_k of dimension $\dim(W_k) = \dim(\{U_i\}_{i\in I}) - \dim(\{U_i\}_{i\in I, i\neq k})$. As before, let $V_i = \pi(U_i)$ for all $i \in I$, let U_S, V_S denote the subspaces of U, V spanned by $\{U_i\}_{i\in S}$ and $\{V_i\}_{i\in S}$, and let $\vec{\mathbf{u}} = \vec{\mathbf{d}}(\{U_i\}), \vec{\mathbf{v}} = \vec{\mathbf{d}}(\{V_i\})$. If $k \notin S$ then $V_S = U_S$ and $\mathbf{v}_S = \mathbf{u}_S$, while if $k \in S$ then U_S contains W_k, the linear transformation π maps U_S onto V_S with kernel W_k, and $\mathbf{v}_S = \mathbf{u}_S - \dim(W_k) = \mathbf{u}_S - \mathbf{u}_I + \mathbf{u}_{I\setminus\{k\}}$.

Thus, $\mathbf{v} = B_k\mathbf{u}$, where B_k is the matrix

$$(B_k)_{ST} = \begin{cases} 1 & \text{if } T = S \\ 1 & \text{if } k \in S \text{ and } T = I \setminus \{k\} \\ -1 & \text{if } k \in S \text{ and } T = I \\ 0 & \text{otherwise.} \end{cases} \tag{A.19}$$

and therefore $B_k \cdot \Upsilon^{\mathbb{F}}_{|I|} \subseteq \Upsilon^{\mathbb{F}}_{|I|}$.

Now assume without loss of generality that $I = \{1, 2, \ldots, n\}$ and let $B = B_n B_{n-1} \cdots B_1 B_0$. We have seen that $B \cdot \Upsilon^{\mathbb{F}}_{|J|} \subseteq \Upsilon^{\mathbb{F}}_{|I|}$. From (A.18) one can see that $B_0\mathbf{1} = B_0\mathbf{1}_e = 0$ and that for every $k \in I$, $B_0\mathbf{1}_k = \mathbf{1}_k$. (Here, it is important to note that $\mathbf{1}_k$ on the left side refers to a vector in $\mathbb{R}^{\mathcal{P}(J)}$ and on the right side it refers to a vector in $\mathbb{R}^{\mathcal{P}(I)}$.) Furthermore, from (A.19) one can see that $B_k\mathbf{1}_k = 0$ and that $B_k\mathbf{1}_i = \mathbf{1}_i$ for all $i \neq k$. Thus, when we left-multiply a vector $\vec{\mathbf{w}} \in \{\mathbf{1}\} \cup \{\mathbf{1}_j\}_{j \in J}$ by the matrix B, one of the following things happens. If $\vec{\mathbf{w}}$ is equal to $\mathbf{1}$ or $\mathbf{1}_e$ then $B_0\vec{\mathbf{w}} = 0$ hence $B\vec{\mathbf{w}} = 0$. Otherwise, $\vec{\mathbf{w}} = \mathbf{1}_k \in \mathbb{R}^{\mathcal{P}(J)}$ for some $k \in I$, $B_0\vec{\mathbf{w}} = \mathbf{1}_k \in \mathbb{R}^{\mathcal{P}(I)}$, and as we proceed to left-multiply $\mathbf{1}_k$ by $B_1, B_2, \ldots$, it is fixed by B_i $(i < k)$ and annihilated by B_k, so once again $B\vec{\mathbf{w}} = 0$. This confirms assertion (ii) of the lemma.

Finally, if $M, M + e$ are matroids satisfying the hypotheses of assertion (iii), then for every set $S \subseteq I$ we have $r(S \cup \{e\}) - r(\{e\}) = r(S)$ and hence $B_0\vec{\mathbf{r}}(M + e) = \vec{\mathbf{r}}(M)$. For any $k \in I$ our assumption on M implies that it has a matroid basis disjoint from $\{k\}$, and hence that $r(I \setminus \{k\}) = r(I)$. Inspecting (A.19), we see that this implies $B_k\vec{\mathbf{r}}(M) = \vec{\mathbf{r}}(M)$ for all $k \in I$, and hence $B\vec{\mathbf{r}}(M + e) = \vec{\mathbf{r}}(M)$ as desired. □

Proof of Theorems A.0.7 and A.0.8. Let $\mathbb{F}$ be a finite field. When $\mathrm{char}(\mathbb{F}) = 2$ our proof applies to Theorem A.0.7 and when $\mathrm{char}(\mathbb{F}) \neq 2$ it applies to A.0.8. Let M denote the matroid $\mathcal{N}$ if $\mathrm{char}(\mathbb{F}) = 2$, and let $M = \mathcal{F}$ if $\mathrm{char}(\mathbb{F}) \neq 2$. In both cases, we will let $M + e$

denote the matroid obtained by adjoining a rank-zero element to M, and we will denote the ground sets of M, $M+e$ by I, J, respectively. Recall the vectors $\Lambda_{\text{even}}, \Lambda_{\text{odd}} \in \mathbb{R}^{\mathcal{P}(I)}$ from Theorems A.0.4 and A.0.5. Let $\Lambda = \Lambda_{\text{even}}$ if $\text{char}(\mathbb{F}) = 2$, $\Lambda = \Lambda_{\text{odd}}$ if $\text{char}(\mathbb{F}) \neq 2$. By Theorems A.0.4 and A.0.5, $\Lambda \cdot \vec{\mathbf{r}}(M) < 0$, a fact that we will be using later.

Recall the linear transformation $B : \mathbb{R}^{\mathcal{P}(J)} \to \mathbb{R}^{\mathcal{P}(I)}$ from Lemma A.0.6, and let

$$\vec{\alpha} = B^{\mathsf{T}} \Lambda.$$

Observe that this transformation gives us the desired inequalities defined in Definitions 2.3.15 and 2.3.16.

For any $\vec{\mathbf{d}} \in \Upsilon^{\mathbb{F}}_{|J|}$ we have $\vec{\alpha}^{\mathsf{T}} \vec{\mathbf{d}} = \Lambda^{\mathsf{T}} B \vec{\mathbf{d}} \geq 0$, since $B\vec{\mathbf{d}} \in \Upsilon^{\mathbb{F}}_{|I|}$ and $\Lambda \cdot \vec{\mathbf{v}} \geq 0$ for all $\vec{\mathbf{v}} \in \Upsilon^{\mathbb{F}}_{|I|}$. The equations $B\mathbf{1} = B\mathbf{1}_j = 0$ for all $j \in J$ imply that $\vec{\alpha}^{\mathsf{T}}\mathbf{1} = \vec{\alpha}^{\mathsf{T}}\mathbf{1}_j = 0$.

Finally, we also have that $\vec{\alpha}^{\mathsf{T}} \vec{\mathbf{r}}(M+e) = \Lambda^{\mathsf{T}} B \vec{\mathbf{r}}(M+e) = \Lambda^{\mathsf{T}} \vec{\mathbf{r}}(M) < 0$, as needed. □

BIBLIOGRAPHY

[1] Micah Adler, Nicholas J.A. Harvey, Kamal Jain, Robert Kleinberg, and April Rasala Lehman. On the capacity of information networks. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 241–250. ACM, 2006.

[2] Amit Agarwal, Noga Alon, and Moses S. Charikar. Improved approximation for directed cut problems. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, STOC 2007, pages 671–680, New York, NY, USA, 2007. ACM.

[3] Amit Agarwal and Moses Charikar. On the advantage of network coding for improving network throughput. In *Information Theory Workshop, 2004. IEEE*, pages 247–249. IEEE, 2004.

[4] Rudolf Ahlswede, Ning Cai, Shuo-Yen Robert Li, and Raymond W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, 000.

[5] Noga Alon and Nabil Kahale. Approximating the independence number via theϑ-function. *Mathematical Programming*, 80(3):253–264, 1998.

[6] Noga Alon and Michael Krivelevich. Constructive bounds for a ramsey-type problem. *Graphs and Combinatorics-an Asian Journal*, 13(3):217–226, 1997.

[7] Noga Alon, Eyal Lubetzky, Uri Stav, Amit Weinstein, and Avinatan Hassidim. Broadcasting with side information. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 823–832, 2008.

[8] Ziv Bar-Yossef, Yitzhak Birk, T. S. Jayram, and Tomer Kol. Index coding with side information. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 197–206, 2006.

[9] Yitzhak Birk and Tomer Kol. Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients. *IEEE/ACM Transactions on Networking (TON)*, 14(SI):2825–2830, 2006.

[10] Anna Blasiak. Multicut lower bounds via network coding. In *2013 International Symposium on Network Coding (NetCod)*. IEEE, 2013.

[11] Anna Blasiak and Robert Kleinberg. The serializability of network codes. In *Automata, Languages and Programming*, pages 100–114. Springer, 2010.

[12] Anna Blasiak, Robert Kleinberg, and Eyal Lubetzky. Lexicographic products and the power of non-linear network coding. In *52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2011)*, pages 609–618. IEEE, 2011.

[13] Anna Blasiak, Robert Kleinberg, and Eyal Lubetzky. Broadcasting with side information: Bounding and approximating the broadcast rate. *IEEE Transactions on Information Theory*, 2013. To Appear.

[14] Ravi Boppana and Magnús M Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT Numerical Mathematics*, 32(2):180–196, 1992.

[15] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoss, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM (JACM)*, 60(1):6, 2013.

[16] T. H. Chan and Raymond W. Yeung. On a relation between information inequalities and group theory. *IEEE Transactions on Information Theory*, 48:1992–1995, 2002.

[17] Terence Chan and Alex Grant. Dualities between entropy functions and network codes. *IEEE Transactions on Information Theory*, 54(10):4470–4487, 2008.

[18] Terence H Chan. Balanced information inequalities. *IEEE Transactions on Information Theory*, 49(12):3261–3267, 2003.

[19] Joseph Cheriyan, Howard Karloff, and Yuval Rabani. Approximating directed multicuts. In *42nd IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 320–328. IEEE, 2001.

[20] Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *Journal of the ACM (JACM)*, 56(2):6, 2009.

[21] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.

[22] Randall Dougherty, Chris Freiling, and Kenneth Zeger. Networks, matroids, and non-Shannon information inequalities. *IEEE Transactions on Information Theory*, 53(6):1949–1969, 2007.

[23] Randall Dougherty, Chris Freiling, and Kenneth Zeger. Linear rank inequalities on five or more variables. *arXiv preprint arXiv:0910.0284*, 2009.

[24] Randall Dougherty, Christopher Freiling, and Kenneth Zeger. Insufficiency of linear coding in network information flow. *IEEE Transactions on Information Theory*, 51(8):2745–2759, 2005.

[25] Randall Dougherty, Christopher Freiling, and Kenneth Zeger. Six new non-Shannon information inequalities. In *2006 IEEE International Symposium on Information Theory*, pages 233–236. IEEE, 2006.

[26] Michelle Effros, Salim El Rouayheb, and Michael Langberg. An equivalence between network coding and index coding. In *IEEE International Symposium on Information Theory (ISIT 2013)*, 2013. To appear.

[27] Salim El Rouayheb, Alex Sprintson, and Costas Georghiades. On the relation between the index coding and the network coding problems. In *IEEE International Symposium on Information Theory (ISIT 2008)*, pages 1823–1827. IEEE, 2008.

[28] Salim El Rouayheb, Alex Sprintson, and Costas Georghiades. A new construction method for networks from matroids. In *IEEE International Symposium on Information Theory (ISIT 2009)*, pages 2872–2876, 2009.

[29] P. Erdős and A. Rényi. On a problem in the theory of graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 7:623–641 (1963), 1962.

[30] Uriel Feige. Randomized graph products, chromatic numbers, and the Lovász ϑ-function. *Combinatorica*, 17(1):79–90, 1997. An earlier version appeared in Proc. of the 27th Annual ACM Symposium on Theory of computing (STOC 1995), pp. 635–640.

[31] Chris Godsil and Gordon Royle. *Algebraic graph theory*, volume 207 of *Graduate Texts in Mathematics*. Springer-Verlag, 2001.

[32] Anupam Gupta. Improved results for directed multicut. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 454–455. Society for Industrial and Applied Mathematics, 2003.

[33] Willem Haemers. An upper bound for the Shannon capacity of a graph. In *Colloq. Math. Soc. János Bolyai*, volume 25, pages 267–272, 1978.

[34] Willem Haemers. On some problems of Lovász concerning the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(2):231–232, 1979.

[35] Nicholas J Harvey, Robert D Kleinberg, and April Rasala Lehman. Comparing network coding with multicommodity flow for the k-pairs communication problem. 2004.

[36] Nicholas J. A. Harvey, Robert Kleinberg, and April Rasala Lehman. On the capacity of information networks. *IEEE Transactions on Information Theory*, 52(6):2345–2364, 2006.

[37] Nicholas J.A. Harvey and Robert Kleinberg. Tighter cut-based bounds for k-pairs communication problems. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, 2005.

[38] Nicholas J.A. Harvey, Robert Kleinberg, Chandra Nair, and Yunnan Wu. A "chicken & egg" network coding problem. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 131–135, 2007.

[39] Tracey Ho, Muriel Médard, Ralf Koetter, David R Karger, Michelle Effros, Jun Shi, and Ben Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52(10):4413–4430, 2006.

[40] AW Ingleton. Representation of matroids. *Combinatorial mathematics and its applications*, 23, 1971.

[41] Sidharth Jaggi, Peter Sanders, Philip A Chou, Michelle Effros, Sebastian Egner, Kamal Jain, and Ludo MGM Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.

[42] Kamal Jain, Mohammad Mahdian, and Mohammad R Salavatipour. Packing Steiner trees. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 266–274. Society for Industrial and Applied Mathematics, 2003.

[43] Kamal Jain, Vijay V Vazirani, Raymond Yeung, and Gideon Yuval. On the capacity of multiple unicast sessions in undirected graphs. In *IEEE International Symposium on Information Theory (ISIT 2005)*, pages 563–567. IEEE, 2005.

[44] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. XORs in the air: practical wireless network coding. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 243–254. ACM, 2006.

[45] Ryan Kinser. New inequalities for subspace arrangements. *Journal of Combinatorial Theory, Series A*, 118(1):152–161, 2011.

[46] Gerhard Kramer and Serap Savari. Edge-cut bounds on network coding rates. *Journal of Network and Systems Management*, 14(1):49–67, 2006.

[47] Michael Langberg and Alexander Sprintson. On the hardness of approximating the

network coding capacity. *IEEE Transactions on Information Theory*, 57(2):1008–1014, 2011.

[48] April Rasala Lehman. *Network Coding.* PhD thesis, MIT, 2005.

[49] April Rasala Lehman and Eric Lehman. Complexity classification of network information flow problems. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms (SODA 2004)*, pages 142–150, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[50] S-YR Li, Raymond W. Yeung, and Ning Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.

[51] Zongpeng Li and Baochun Li. Network coding: the case of multiple unicast sessions. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, 2004.

[52] Nati Linial and Umesh Vazirani. Graph products and chromatic numbers. In *30th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1989)*, pages 124–128. IEEE, 1989.

[53] László Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25(1):1–7, 1979.

[54] Eyal Lubetzky and Uri Stav. Nonlinear index coding outperforming the linear optimum. *IEEE Trans. Inf. Theor.*, 55(8):3544–3551, August 2009. An earlier version appeared in Proc. of the 48th Annual IEEE Symposium of Foundations of Computer Science (FOCS 2007), pp.161-167.

[55] Konstantin Makarychev, Yuri Makarychev, Andrei Romashchenko, and Nikolai Vereshchagin. A new class of non Shannon type inequalities for entropies. *Communications in Information and Systems*, 2(2):147–166, 2002.

[56] Hamed Maleki, Viveck Cadambe, and Syed Jafar. Index coding: An interference alignment perspective. In *IEEE International Symposium on Information Theory (ISIT 2012)*, pages 2236–2240. IEEE, 2012.

[57] Frantisek Matus. Infinitely many information inequalities. In *IEEE International Symposium on Information Theory (ISIT 2007)*, pages 41–44. IEEE, 2007.

[58] Muriel Médard, Michelle Effros, David Karger, and Tracey Ho. On coding for non-multicast networks. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, volume 41, pages 21–29. The University; 1998, 2003.

[59] Dhruv Mubayi and Jason Williford. On the independence number of the erdos-rényi and projective norm graphs and a related hypergraph. *J. Graph Theory*, 56(2):113–127, 2007.

[60] Michael Saks, Alex Samorodnitsky, and Leonid Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24:525–530, 2004.

[61] Lihua Song, Richard W. Yeung, and Ning Cai. Zero-error network coding for acyclic networks. *IEEE Transactions on Information Theory*, 49(12):3129–3139, 2003.

[62] Hua Sun and Syed A Jafar. Index coding capacity: How far can one go with only Shannon inequalities? *arXiv preprint arXiv:1303.7000*, 2013.

[63] Avi Wigderson. Improving the performance guarantee for approximate graph coloring. *J. Assoc. Comput. Mach.*, 30(4):729–735, 1983.

[64] Raymond W. Yeung. *A First Course in Information Theory*. Springer, 2002.

[65] Raymond W. Yeung and Zhen Zhang. Distributed source coding for satellite communication. *IEEE Transactions on Information Theory*, 45(4):1111–1120, 1999.

[66] Zhen Zhang and Raymond W Yeung. A non-Shannon-type conditional inequality of information quantities. *IEEE Transactions on Information Theory*, 43(6):1982–1986, 1997.

[67] Zhen Zhang and Raymond W. Yeung. On characterization of entropy function via information inequalities. *IEEE Transactions on Information Theory*, 44(4):1440–1452, 1998.