Guessing Games on Undirected Graphs

Anh Nhat Dang

Queen Mary
University of London

A thesis submitted for the degree of

Doctor of Philosophy

London 2015

Statement of Originality

I, Anh Nhat Dang, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged and my contribution indicated. Previously published material is also acknowledged in the next page.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Anh Nhat Dang
Aug 3rd , 2015

Publications

Parts of this thesis have been published; Chapters 3, 4, and the second section of 5 have been published or under submission as the papers listed below, in chronological order. Chapter 6 is currently under preparation for submission.

[23] Peter J. Cameron, **Anh Nhat Dang**, and Søren Riis. Guessing Games on Triangle-free Graphs. *CoRR*, abs/1410.2405, 2014.

[5] Rahil Baber, Demetres Christofides, **Anh Nhat Dang**, Søren Riis, and Emil Vaughan. Graph Guessing Games and non-Shannon Information Inequalities. *CoRR*, abs/1410.8349, 2014.

[4] Rahil Baber, Demetres Christofides, **Anh Nhat Dang**, Søren Riis, and Emil Vaughan. Multiple unicasts, graph guessing games, and non-Shannon inequalities. In *International Symposium on Network Coding, NetCode 2013, Calgary, AB, Canada, June 7-9, 2013*, pages 1–6, 2013.

Abstract

Guessing games for directed graphs were introduced by Riis for studying multiple unicast network coding problems. In a guessing game, the players toss generalised die and can see some of the other outcomes depending on the structure of an underlying digraph. They later simultaneously guess the outcome of their own die. Their objective is to find a strategy that maximises the probability that they all guess correctly. The performance of the optimal strategy for a digraph is measured by the guessing number.

In general, the existence of an algorithm for computing guessing numbers of a graph is unknown. In the case of undirected graphs, Christofides and Markström defined a strategy that they conjectured to be optimal. One of the main results of this thesis is a disproof of this conjecture. In particular, we illustrate an undirected graph on 10 vertices having guessing number which is strictly larger than the lower-bound provided by Christofides and Markström's method. Moreover, even in case the undirected graph is triangle-free, we establish counter examples to this conjecture based on combinatorial objects known as Steiner systems.

The main tool thus far for computing guessing numbers of graphs has been information theoretic inequalities. Using this method, we are able to derive the guessing numbers of new families of undirected graphs, which in general cannot be computed directly using a computer. A new result of the thesis is that Shannon's information inequalities, which work particularly well for a wide range of graph classes, are not sufficient for computing the guessing number.

Another contribution of this thesis is a firm answer to the question concerning irreversible guessing games. In particular, we construct a directed graph G with Shannon upper-bound that is larger than the same bound obtained when we reverse all edges of G.

Finally, we initialize a study on noisy guessing game, which is a generalization of noiseless guessing game defined by Riis.

We pose a few more interesting questions, some of which we can answer and some which we leave as open problems.

TO MY FAMILY

Acknowledgements

First and foremost I would like to thank my advisor, Dr. Søren Riis, for his patient guidance, generosity and friendship during these many years. He has always been available for discussions and advice, not only on the ideas presented in this thesis but also on general issues when I was lost within my research and future plans. I also thank him for his works on guessing games which are the fundamental pillar for the results developed in this thesis.

Secondly, I wish to thank Professor Peter Cameron for his advice, encouragement, and kindness. Indeed, Professor Cameron has been my de facto co-supervisor and the results on guessing games on triangle-free graphs would not exist without our discussions on strongly regular graphs. I also want to thank Professor Cameron for helping me shape my knowledge and determination to work in mathematics.

I am deeply grateful to my second advisor, Professor Peter Keevash, for his guidance, generosity, and especially his feedback on my progress. Without his advice, the results presented here would not be in optimal form.

I am thankful to my co-authors and former colleges Dr. Demetres Christofides, Dr. Emil Vaughan, Dr. Maximilien Gadouleau and especially Dr. Rahil Baber. I deeply appreciate their support and encouragement to extend the results of our papers. They also deepened my knowledge in many areas of mathematics, especially graph theory and group theory. Dr. Baber has always been a great mentor and friend who not only contributed a large part in our first paper, but also gave me general advice about PhD life.

I thank Dr. Mark Wildon and Prof. Klas Markström for their positive feedback on the early version of this thesis. Many changes were made to improve the illustration of this thesis are based on their carefully review and suggestions.

I also benefited greatly from joining several research study groups including the QMUL Combinatoric Study Group and the London Number Theory Study Group. Therefore, I am deeply indebted to professor Fred Diamond. He was the one who introduced me to the London Number Theory Study Group. Thanks to him, I was able to clearly define my mathematical research.

I also wish to thank my friends for their help and friendship. Their encouragement plays an important role for me to complete this thesis. I especially want to thank Ms. Marcela Schaefer for her careful edits on the language presented in this dissertation.

Most importantly, I would like to thank my family; their love has helped me overcome all difficulties. Without the encouragement from my mother, Nguyen Thi Thu Phuong, and my younger sister, Dang My Linh, I might not have been able to start and continue until the very end of this long road. I thank my father, Dang Huy Binh, for his support and advice all these years. I also thank members of my extended family for their moral support.

I am grateful for the scholarship of the School of Electronic Engineering and Computer Science, which covered my tuition fees and stipends to live in London. Finally, I thank Melissa Yeo for her understanding and her help on my paper works.

Contents

Chapter 1

Introduction

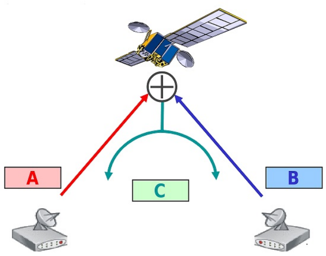
1.1 Motivation

Let us consider the following scenario: Suppose we have two ground stations, A and B, in nonline-of-sight that wish to communicate with each other via a satellite C. In this setting, each ground station can send a message that it wishes to deliver to satellite C and C will broadcast this message to both stations. For simplicity, we would assume that all the communication channels are noiseless, hence, each node in this network can receive the exact data being sent. The best data scheme for this problem using the traditional store-and-forward networking guarantees that the total number of messages being sent by both station equals the number of messages being broadcasted by satellite C. A question arises: Can we do better?

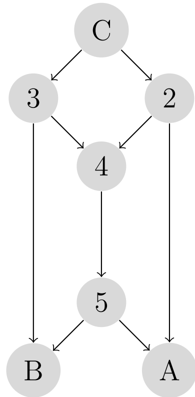
Ahlswede, Cai, Li, and Yeung [2] approached this problem from a new perspective. In their paper "Network Information Flow," a new data scheme is illustrated. Instead of letting nodes forward messages, vertices in the network are allowed to send encoded messages, which are obtained by applying algebraic operations such as addition, subtraction, and XOR on incoming packages. As opposed to the traditional store-and-forward point of view, where information is treated as commodity and data communication is seen as sending packages through networks without changing the content, the new approach by Ahlswede et al. sees data as liquid, which can be mixed and manipulated, hence the name information flow.

Back to our example, a new data scheme can be adopted:

Suppose the ground station A wants to send message a, and ground station B wishes to send b. When both of these messages reached C, a new message $a \,\mathrm{XOR}\, b$ is then



(a) Satellite problem



(b) Multicast reformulation

Figure 1.1: Satellite communication problem and the multicast reformulation.

computed and broadcasted. At the terminal side, the original message can be recovered by applying the XOR operation on received data and sending data, e.g. $a\,\mathrm{XOR}(a\,\mathrm{XOR}\,b) = b$. This new scheme allows us to reduce the number of messages being broadcasted significantly. In the optimal case where the number of sent messages by both stations are equals, the number of broastcasting messages by satellite C will be half of the total number of sent messages.

Our previous example is an instance of a problem known as a multicast problem. This problem concerns raises a question: is it possible to deliver messages from one source to all of its sinks simultaneously? The case of one source – one sink is answered by the max-flow min-cut theorem stating that the amount of information that can be delivered equals the capacity of the minimum cut separating the source and sink. Moreover, the optimal value can be computed efficiently. However, when there are more than one sink, as in our example, the problem changes significantly. Following the same mindset of the max-flow min-cut theorem, the answer to the question in this case requires us to compute a fractional packing of Steiner trees. Compared to the fractional packing of paths needed for only one sink, the Steiner tree packing problem is NP-hard [57]. Moreover, the "optimal" solution does not match the cut upper bound – the minimum of all minimum source-sink cuts. In particular, the fractional packing solution is a multiplicative factor of $\Omega((\log n/\log\log n)^2)$ smaller than the cut in directed graphs and lies between 36/31 and 1.55 the cut in undirected graphs [1, 20].

With the information flow approach in mind, Ahlswede et al. [2] were able to prove that the optimal coding rate is equal to the cut upper bound, hence we gain a huge throughput in comparison with the traditional routing method. The work of Li et al. [93] further shows that this optimality can be achieved by using a very simple family of codes – the linear codes. Moreover, an optimal linear code can be found in polynomial time [54]. Therefore, we do not have to worry about the NP-hard packing problem in the classical setting. Surprisingly, this optimal solution can also be achieved by randomized algorithms. In particular, with high probability, the solution in which each node outputs a random linear combination of its input is optimal [53]. These results motivate practical applications of network information flow. An example of successful application in wireless networks is [59], where the new communication framework provides faster transmission rates compared to traditional routing. The term "network coding" is coined for this paradigm of encompassing coding and retransmitting of messages at the intermediate nodes of the network.

The great starting point of multicast network coding motivates studies of this method in other general settings, e.g. multiple-unicast network, multiple-multicast network. A formal description of a network coding problem is as follows [13]:

Definition 1. *[13] A network coding problem* $(G, I, \mathrm{Src}, \mathrm{Snk})$ *is specified by the following data:*

- $G = (V, E)$ *is a directed acyclic graph.*
- *Each edge* $e \in E$ *is assigned a non-negative integer* $c(e)$ *called channel capacity.*
- I *is the set of* k *messages.*
- $\mathrm{Src} : I \to \mathcal{P}(V)$ *is called the source function, where* $\mathcal{P}(V)$ *is the power-set of* V.
- $\mathrm{Snk} : I \to \mathcal{P}(V)$ *is called the sink map.*

In particular, a multicast problem is when $|I| = |\mathrm{Src}(I)| = 1$.

For simplicity, we will always assume that messages in I are arbitrary elements of a fixed finite alphabet Σ_i. We also choose a finite alphabet Σ_e, called the edge alphabet. At any node $v \in V$, we define encoding function f_{e_v} for every out-edge of v which maps the symbols in Σ_e which are carried on the in-edges to v, or a symbol in Σ_i if v is a source, to elements in Σ_e. Also each sink $s \in V$ is associated with a decoding function f_s which maps elements which are carried on are carried on the in-edges to s to elements in Σ_i.

Definition 2. *[13] A network coding solution is a tuple* $(\Sigma_i, \Sigma_e, (f_e)_{e\in E}, (f_s)_{s\in \mathrm{Snk}(I)})$ *where:*

- *Σ_i is called a source alphabet.*
- *Σ_e is called an edge alphabet.*
- *Encoding functions $(f_e)_{e \in E}$ and decoding functions $(f_s)_{s \in \mathrm{Snk}(I)}$ such that for every k-tuple of messages $x = (x_1, x_2, \ldots, x_k) \in \Sigma_i^k$:*

Edge-Source-Constraints *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. By computable we mean that $f_{(u,v)}$ is a well-defined deterministic map.*

Edge-Sink-Constraints *For every sink s that requires message i, the functions on in-edges to s together with the messages for which v is a source are sufficient to determine the value of x_i.*

A network is said to be solvable if there exists a coding solution for it.

In a non-acyclic graph, the *Edge-Source-Constraints* is not a sufficient characterization of soluvability. 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 from the functions on in-edges to u preceding (u, v) in the ordering. See [56, 50, 51, 14] for more discussions on how to define coding for graphs with cycles.

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 messages 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, where each source has a unique corresponding sink, i.e. the case where $|\mathrm{Src}(i)| = |\mathrm{Snk}(i)| = 1$ in Definition 1.

Regarding this general setting, questions arise: Are linear codes sufficient to provide the maximum throughput? Can we compute the coding rate efficiently?

A large portion of literature has been dedicated to answering these questions for various optimization problems. A wide range of mathematical tools have been adopted to solve open problems and also to create new ones. To name a few: abstract algebraical objects, such as commutative rings or finite fields; combinatorial objects such as matroids or graphs; geometry, and optimization among others. Furthermore, recent progress in information theory plays a leading role in answering many compelling questions about the coding capacity of an information flow networks. The rapid growth of the field filled with many interesting studies indicates that network coding theory and its application have been under the spotlight.

We now give a short overview of the previous work addressing our guiding questions in relation to the general network coding problem and the multiple unicast problem. This is my attempt to provide a bird's-eye view on theoretical results in network coding theory focusing on multiple-unicast setting and its interaction with information theory. For a more encyclopaedic survey, we recommend the readers refer to [9], which contains a comprehensive list of active areas in network coding and an extensive bibliography. Otherwise, readers can refer to [41, 42, 31, 84] for theoretical surveys and [33, 73] for applications.

1.1.1 Are linear codes sufficient to provide the maximum throughput?

There are two classifications of linear codes in the literature: scalar linear codes, in which messages are elements of a finite commutative ring such as $\mathbb{Z}/n\mathbb{Z}$ or a finite field $\mathbb{F}_q$ of order q, and vector linear codes, in which messages are elements of a finite-dimensional vector space defined over some finite field. The work of Li, Yeung, and Cai [93] showed that if a multicast network M is solvable, then by enlarging the side of the alphabet (if needed), we can find a scalar linear solution for M. It was Riis [80] who showed that we can find a linear solution over a vector space defined over $\mathbb{F}_2$ for every solvable multicast network. A series of articles [76, 68, 55] investigate the properties of vector linear code, which proved to be more powerful than scalar linear version for network coding in more general setting. In particular, there are networks that have vector linear codes, but no scalar linear one. It is therefore natural to hope that linear codes also provide at least a good approximation to an optimal solution for coding. This conjecture was raised in [76] and was disproved in [34]. A counter example was established by Dougherty, Freiling, and Zeger where a non-linear code that exceeds the vector linear code by a factor of 11/10 [34]. The intuition behind the construction of the network was explained in [36] and for the first time, a new technique of building information networks from matroids was introduced. This technique turned out to be influential and many variations of this core idea were implemented in other settings such as [40, 15]. In fact, based on the technique introduced by Dougherty et. al. combined with properties of graph product, Blasiak et. al. established an example of index coding instance where an $\Omega(n^{1/2-\epsilon})$ multiplicative gap between vector linear and non-linear coding exists.

1.1.2 Can we compute the coding rate efficiently?

The existence of an algorithm for computing the coding rate of a given network is possibly the most important unanswered question in network coding. Despite a significant body of work devoted to address this problem, we have achieved almost no progress. It is suggested that there might be the case that no such algorithm exist [67]. The potential undecidable property of this problem arises from the fact that the size of an alphabet can be arbitrarily large, which implies that the optimal encoding/decoding functions live in an infinite search space. Furthermore, even for a fixed alphabet size, defining a good approximation algorithm is not a trivial task. To the best of my knowledge, using a method of code graph developed in [45, 28] (which we will recall in Section 3.3 Chapter 3), finding an optimal coding function is equivalent to computing the independent number of a non-trivial graph of order k^n, where n is approximately the size of the multiple unicast networks and k is the size of a fixed alphabet.

The potential undecidability of the problem implies that we have no result that relates finding an optimal network coding solution to problems in P or NP classes. In fact, it cannot be ruled out that there exists a linear time algorithm that computes coding rates in the most general setting of the problem with arbitrary coding functions over arbitrarily large alphabet sizes. However, when restricting the space of coding functions to be linear codes over a fixed alphabet, Lehman and Lehman [68] showed that finding the optimal scalar linear code of a multiple unicast network is equivalent to a 3-SAT problem, which is known to be NP-hard. Moreover, even finding a constant approximation of the optimal coding function in the space of all coding function defined over a fixed alphabet size is hard, assuming the unique games conjecture [66].

As an alternative to computing exact coding capacities, there has been a significant body of work devoted to determining bounds on the coding and linear coding capacity of a network, since a good upper-bound provides us with a value that can be compared to a lower bound derived by some well-known method. If in the best case, these two values agree, we achieve an optimal solution. Otherwise, we know that the maximum gap between the upper bound and the optimal solution cannot be too large, since this gap sets a limit on the approximation ratio.

The most common strategy to derive an upper-bound is to use information-theoretic arguments. The intuitive idea is that we can consider flows along edges in our network as jointly distributed random variables over some probability space. Given a probability distribution, we associate each subset of the edge set to an information measure known as Shannon entropy of the jointly distributed random variables. A collection of all entropy value with respect to the given distribution returns us a vector of non-negative numbers regarded as an entropic vector of the network code. It was proved in [95] that the closure of the set of all entropic vectors of network codes is a convex set. Therefore, finding an upper-bound of a network coding problem is equivalent to an optimization problems over the mentioned space. There are two constraints that characterize this space of entropic vectors:

The first constraint type is imposed by the combinatorial structure of the network [97]. In our case of directed acyclic graphs, this constraint stands for the computability property of encoding/decoding functions given its input. In particular, it says that at vertex v, the conditional entropy of a random variable which corresponds to an out-edge of v given all random variables of in-edges of v equals 0. For the case where the network contains a cycle, various constraints have been introduced over the years [56, 50, 65, 51] yet a complete classification is still missing.

The second constraint type comes purely from information theory. These constraints are information inequalities. Information inequalities are independent of the network topology as they hold universally for all n-tuples of random variables. Those inequalities are referred to as the law of information theory. Given a positive integer $n \geq 2$, we can consider the collection of all possible jointly distribution of n random variables $X_1, \ldots, X_n$. For each joint distribution, we can form a corresponding entropic vector of length $2^n - 1$. The set of all such vectors is denoted by Γ_n^* with closure denoted by $\bar{\Gamma}_n^*$. A complete characterization of $\bar{\Gamma}_n^*$ requires finding all possible information inequalities [95]. A first list of information inequalities, known as Shannon inequalities, are of the following form:

$$H(AC) + H(BC) - H(ABC) - H(C) = I(A;B|C) \geq 0$$

A, B, C are disjoint collections of random variables. Using these inequalities, an outer-bound of Γ_n^* can be computed and we denote the set of all non-negative vectors to satisfy these inequalities as Γ_n.

The Shannon inequalities can be rewritten in a form of polymatroidal axioms:

Monotonicity $(H(A) \leq H(AB))$,

Non-negativity $(H(A) \geq 0)$,

Submodularity $(H(A) + H(B) \geq H(A \cup B) + H(A \cap B))$.

For $n \leq 3$, the set $\bar{\Gamma}_n^*$ and Γ_n coincide [102]. However, for $n > 3$, $\bar{\Gamma}_n^* \subsetneq \Gamma_n$ [98]. In particular, this means that there exist additional inequalities other than Shannon inequalities. These mysterious inequalities are called non-Shannon inequalities, and deriving them is the main topic of a significant amount of work, e.g. [98, 72, 69, 99, 35, 75, 91, 26, 37].

One important observation from these works is that a satisfactory characterization of $\bar{\Gamma}_n^*$ is still far from complete. Even in case $n = 4$, there are infinitely many non-Shannon inequalities coming in families [75, 91, 37].

Back to our story of network coding, a question was raised concerning whether a set of all Shannon information inequalities combined with the network topological constrains provides us an exact bound for optimal network coding solution. The answer to this question is negative, and the first counterexample was established by Dougherty et al. in their influential paper [36] where a connection between non-representable matroid and possible requirement for non-Shannon inequalities in computing network code was illustrated. Furthermore, [25] shows that a complete description of the capacity of network coding implies a depiction of Γ_n.

Other attempts to bounding the capacity regions by combining the sparsest cut, an upper bound on the flow rate in the multicommodity flow problem, with information inequalities. For example, it is known that the capacity of a cut that disconnects all sinks from all sources is an upper bound on the network coding rate. Improving this trivial bound requires complicated information-theoretic arguments [87, 48, 49, 65, 14]. Unfortunately, each of these bound can be larger than the coding rate by a factor of the order of the underlying graph; therefore, these are not very good approximations.

Considering all previous works, it is crystal clear that outside the multicast setting, network coding is a hydra. The difficulty of the general problem has motivated works defined over simplified settings. One such setting was thought of by Riis [82] and its dual problem was introduced independently by Birk and Kol [11]. We will discuss the class of problems introduced by Riis in the next section as it is more natural from the our perspective.

1.1.3 Guessing numbers

Let us put the following restrictions on a multiple-unicast network coding problem $(G, I, \mathrm{Src}, \mathrm{Snk})$:

- G is a directed acyclic graph.
- $c(e) = 1$ for every $e \in E$.
- For every $i \in I$, $|\mathrm{Src}(i)| = 1$, and $\mathrm{Src}(i) \neq \mathrm{Src}(j)$ for every $i \neq j$.
- For every $i, j \in I$, $|\mathrm{Snk}(i)| = |\mathrm{Snk}(j)| = 1$ and $\mathrm{Snk}(i) \neq \mathrm{Snk}(j)$ for every $i \neq j$.
- The source alphabet Σ_i and the edge alphabet Σ_e are identical.
- Every node in G will broadcast its information via its out-edges.

In other words, we consider a multiple-unicast problem where edge capacity is 1 and for every message $i \in I$, i is generated by a unique source node $\mathrm{Src}(i)$ and is demanded by a unique sink node $\mathrm{Snk}(i)$, all using the same alphabets.

In this setting, Riis converted the multiple-unicast problem into a combinatorial problem defined on directed graphs, which is called the guessing game. He then introduced a concept called "the guessing number of a digraph" [82] which connects graph theory, network coding, and circuit complexity theory. This definition of guessing numbers is particularly important because of the following theorem:

Theorem 1. *[82] A multiple unicast network of n sources and n sinks defined on an acyclic digraph D is solvable over a given alphabet if and only if the guessing game played on the digraph D' and the given alphabet has guessing number n.*

The digraph D' in the theorem is obtained from D by a deterministic process. In particular, if D is the butterfly network, then D' is a clique of order 3 (Figure 1.2).

Combined with results in [82] and [39], which show that any network coding instance can be converted to an equivalent multiple-unicast problem, we are guaranteed that the guessing number is a direct criterion on the solvability of network coding. Similar to linear code of multiple-unicast network, we can define the linear guessing number, which evaluates the solvability of a given network instance restricted to the class of linear maps. Comparing these two quantities, we can evaluate the performance of linear network coding in comparison with that of general network coding. Beside this connection to network coding, the guessing number also found an application in

(a) The Butterfly network (b) guessing game on K_3

Figure 1.2: The Butterfly network and its guessing game equivalence.

disproving a long-standing open conjecture in circuit complexity concerning with the optimal Boolean circuit for a Boolean function [89].

Progression on bounding the value of the guessing number and the linear guessing number of digraphs has been made. In particular, the guessing number and linear guessing number for some families of digraphs have been completely characterized [82, 90, 28]. Moreover, just like the case of general network coding, we can define an entropic measure for a digraph. This value is called the entropy of that graph and it was proved that the guessing number and the entropy of the same graph agree[1] [82]. Following the works [102], [98] and [36], it indicates that it might be the case that the entropy of a digraph cannot be derived by using the Shannon inequalities alone.

A complementary measure of the guessing number of a digraph is the so-called public entropy [81]. This entropy measure the information defect which is known to be equal to the length of a minimal index code induced on the graph D [81]. (We refer the reader to [40] for the concept of index coding and its relation to network coding.)

Just like the case of general network coding, little is known about the guessing number of a digraph, and even for undirected graphs (From now on, we treat an undirected graph as a special type of digraph where an undirected edge represents two directed edge going in opposite directions). In particular, we do not have an algorithm for computing the guessing number, and our best attempt to provide a good upper-bound for the guessing number is via linear programming using graph constraints and information inequalities. Fortunately, this work can be done by computer. However, the program can only deal with graphs of small orders due to its limited computation

[1] We should not confuse this notion of graph entropy mentioned in this thesis and the graph entropy for undirected graph introduced by Körner in [64]. These two quantities are fundamentally different.

power. Despite this, progress has been made towards understanding the guessing number. In the most general case of digraph, Gadouleau and Riis constructed an algorithm for computing optimal coding functions under the constraint that the alphabet size is fixed [44]. As for finding a good approximation of a guessing number, the paper [90] showed that the complement of the minimal rank over $\mathbb{F}_2$ of a matrix representing D is a lower-bound. For the case where the underlying graph is undirected, an explicit description of an lower bound for the guessing number was introduced by Christofides and Markström [28]. This lower-bound provides an exact bound for guessing numbers in a number of cases.

Concerning the index coding problem, new classes of codes have been developed, such as the scalar linear codes hinging on a greedy clique-cover [11], or the matrix rank linear coding schemes over $\mathbb{F}_2$ [7]. This class of codes turns out to be a generalization of the result proved in [90]. Since the alphabet is not limited in $\mathbb{F}_2$, Lubetzky and Stav [71] considered the problem of linear codes over general finite fields $\mathbb{F}_q$. Combined with arguments from Ramsey theory, they show that for every positive real number ϵ there is a family of graphs on $n(\epsilon)$ vertices for which we can find a good linear code over some finite field of odd characteristics, but not over $\mathbb{F}_2$. Moreover, by defining index coding in a more general setting, Blasiak et. al. [15] were able to construct a hypergraph in which the optimal coding function is non-linear and the separation between non-linear coding and its vector linear counterpart is $\Omega(n^{1/2-\epsilon})$. This construction combined the matroid construction in Dougherty et. al. [36] and graph product. Applications of matroid theory in constructing guessing game/index coding instance can be found in [88, 40, 15].

Regarding the current development of the field, there are few questions that we would like to address in this work. Firstly, we consider the problem of finding a good lower-bound for guessing numbers. Solving this problem seems to be out of our reach when our graphs are general digraphs. However, if we are restricted to the case of undirected graphs, the lower-bound constructed by Christofides and Markström [28] seems to be a good approximation, since at least it uses information about the symmetry of the underlying undirected graph. Moreover, it is a generalisation of the linear code introduced in [11]. Can we say this lower-bound is also the exact bound of guessing numbers for every undirected graph? In addition, even in case the answer to the previous question is negative, can we still find families of undirected graphs for which the equality holds? The second problem that we wish to tackle is about the sufficiency of Shannon information inequalities for bounding the guessing number of a digraph.

It is notable that if we use the construction introduced in [36, 88, 40, 15], what we receive is generally a hypergraph. Therefore, even in case there are hypergraphs with non-trivial broadcasting rate where in order to approximate this value, non-Shannon inequalities must be involved, this construction might not be easily translated into the language of guessing games on digraphs or undirected graphs.

1.2 Contributions

Chapter 4 and Section 5.2 of Chapter 5 are joint work with Rahil Baber, Demetres Christofides, Peter J. Cameron, Emil Vaughan, and Søren Riis. These works appear in [4], [5], [23]. The author is one of the main investigators of results in [4, 5], and is the main researcher of [23].

Our contribution to guessing games over undirected graphs encompass the following topics: exhibiting gaps between bounds on guessing numbers of undirected graphs, reversible and irreversible guessing games, the improvement of guessing numbers when additional directed edges are introduced, and different bounds of information inequalities.

The first part of Chapter 5 and all of Chapter 6 are an attempt by the author to address the following problems:

- matching between vector-linear bounds and guessing numbers of undirected graphs
- the generalization of guessing numbers with the introduction of noises.

We now give more details about all of our results and their locations in this thesis.

Different bounds of guessing numbers (Chapter 4)

The fractional clique cover bound introduced by Christofides and Markström [28] is important as it indicates when an undirected graph has a vector linear optimal guessing strategy over every finite field. That is, it shows that if a graph has its guessing number matching its guessing number by fractional clique cover, then for every finite field, we can find an optimal guessing strategy, which is a vector-linear. We also have other bounds on guessing numbers such as the minimum rank of a matrix representation of a guessing game over a fixed finite field. We can also generalise the minimum rank of matrix representation to be the minimum rank of vector-matrix representation defined over a fixed finite field.

It is important to note that we have not seen any example of a guessing game where its guessing number can only be achieved by non-linear guessing strategies.

Moreover, a theorem by Riis [83] shows that if a multiple-unicast instance is weakly irreversible [2], in which case the network must only have solutions that use a non-linear coding method as its optimal code.

These results strengthen the belief that the vector-linear strategy at its most general form is sufficient for all guessing games played on an undirected graph.

We know that the guessing number can be bounded from above via linear programming with information-theoretic constrains. This upper bound is, in theory, larger than the actual guessing number, but in practice, these two quantities agree for almost every testable instance when our graph is undirected. In fact, this bound matches the lower-bound provided by the fractional clique cover bound. Therefore, it was conjectured that the fractional clique cover bound is in fact also the guessing number of an undirected graph.

We disproved this conjecture by showing that there are undirected graphs where optimal guessing strategies out-perform the fractional clique cover strategies established in [28]. The first counterexample is a graph on 10 nodes, where its guessing number can only be obtained by using vector-matrix guessing strategy in its most general form. This is Theorem 11 in Section 4.1. Notice that the vector-matrix guessing strategy depends on the characteristic of the finite field. Based on this counterexample, we constructed an undirected graph where its guessing number is obtained by the minimum rank of a matrix representation, which is essentially a linear code. Moreover, this guessing number cannot be obtained from the fractional clique cover method. This is Theorem 12 in Section 4.1.

We show that even in case our graph is triangle-free, that is, we forbid the appearance on any clique of order greater than 2, there are many graphs with guessing numbers that are significantly higher than the bound provided by the fractional clique cover

[2] A multiple-unicast network is said to be weakly irreversible if the number of messages defined over a given alphabet that can be send in the network does not equal the number of messages defined over the same alphabet that can be sent in a reversed network which is obtained by reversing all the edges in the underlying digraph.

method. We construct these results based on combinatorial structures known as Steiner systems. The statements and proofs can be found in Section 4.3.

It is noticeable that all of our counterexamples to the fractional clique cover conjecture are highly non-trivial and we can amplify the difference between these bounds by various graph products discussed in the work of Blasiak et. al.

Effect of additional directed edges on guessing number of an undirected graph (Chapter 4)

A challenging problem in the application of network coding is the variation of network topology along time. A simplified question is: how much does the coding capacity change when new communication channels are introduced to the already existing topology? Back to the case of mutiple-unicast networks, can we ask for a network in which its coding capacity is significantly changed when we add or remove a single channel? An equivalent question is asking for a guessing game played on some digraph such that the guessing number changes when a directed edge is introduced or removed from the graph. An obvious example to this problem is by letting D be a directed acyclic graph. Multiple authors have noticed that the guessing number of an acyclic graph is 0. If we introduce a new directed edge to D such that the new digraph contains a directed cycle, then it is easy to show that the guessing number of the new digraph is exactly 1.

Beside these results, however, can we obtain the same effect when our underlying graph is an undirected graph? The answer to this question is positive. It is interesting to note that the construction was based on our counter-example to fractional clique cover strategy. The statement of the problem and the proof can be found in Section 4.1.

Application of non-Shannon information inequalities (Chapter 5)

Just like the situation concerning the general network information flow problem, we progressed very little towards finding an algorithm for computing the guessing number of a given digraph. The closest answer to an algorithm is in the work of Gadouleau and Riis [45], and Christofides and Markström [28]. In these works, putting a constraint on the size of the alphabet, a guessing number over the given alphabet can be read off as an logarithm of an independent number of a vertex-transitive undirected graph.

Unfortunately, the order of this non-trivial graph is exponentially large. In particular, if the guessing game is played on a graph of order n over alphabet of size k, the order of the associated vertex-transitive undirected graph is k^n. Finding the independent number for such graph is intractable with our current knowledge.

As an alternative to computing guessing numbers, we can adopt the method of using information-theoretic arguments to compute an upper bound for guessing numbers and match it up with some well-known lower bounds introduced above. As the digraph is obtained by converting a multiple-unicast network coding instance with directed acyclic topology, we can easily convert the constraints imposed on network instance discussed above into the information constraints defined for the guessing game. Therefore, these information constraints also consist of two types. The first type reflects the combinatorial structure of the digraph, while the second type comes purely from information theory. Constraints of the second type are essentially information inequalities. Naturally, a question arises: does the set of all Shannon information inequalities combined with the graph topological constrains provide us with an exact bound for guessing numbers?

Prior to this work, the answer to this question was open. Specifically, we only know one core example of network (and some variations of it) in the literature where non-Shannon information inequalities provide better bounds on coding capacity compared to Shannon's. The construction of the network, which was developed by Dougherty et al. [36], was involved properties of the Vámos matroid, which is known to be non-representable. The resulting network is a multiple-multicast instance; i.e., each source has more than one sink. Variations of the matroidal method are developed for index coding where the obtained graphs are generally hypergraphs. Therefore, we cannot adopt the method introduced in [36] because the networks obtained from this construction generally cannot be converted into an undirected graph.

Our new result shows that there exists a graph where non-Shannon inequalities provide a better approximation of the guessing number compared to using the Shannon information inequalities alone. In fact, we show that there are gaps between the bounds provided by the Zhang-Yeung inequalities and its alternative provided by the Dougherty-Freiling-Zeger inequalities.

Following the general strategy set out in [15], it might be possible to extend these gaps by using different type of graph products.

One important note in our result is the the example is an undirected graph of small order. Therefore, it is possible to compute and check the calculation by computer. In fact, the process of deriving these bounds is achieved by computer and data the file can be obtained upon request. The result appears in Section 5.2.

Irreversible guessing games (Chapter 5)

The existence of an irreversible multiple-unicast network information flow instance is still open. We say a multiple-unicast network D is irreversible if that the coding capacity of D does not equal the coding capacity of the network Reverse(D) obtained by reversing all edges of D[3]. Back to our setting of guessing game on digraph, we can ask for an instance of a digraph where its guessing number is different from the guessing number of the digraph obtained by reversing all the directed edges. We were not be able to find an answer to this problem. However, we can provide an example for a weaker question concerning the existence of a digraph where its Shannon-bound is different from the Shannon-bound of its reverse. We show a related result, the answer to this question is positive. The construction and the statement of the result is demonstrated in Section 5.2.

Exact bound for the guessing numbers of families of undirected graphs (Chapter 5)

As it is unknown to generally compute the guessing number even in cases of using non-Shannon information inequalities, we may try to find graphs where we can verify the upper-bound of guessing numbers provided by information inequalities and the lower-bound by some well-known method match. Very few examples for families of graphs can be found in the literature. Before this work, the guessing number of these following families of undirected graphs were known:

- perfect graphs,
- C_n – the n cycle,
- C_n^c – the complement of the n cycle,
- 3-regular Cayley graphs of $\mathbb{Z}/n\mathbb{Z}$ of special type,

[3]An example of a multiple-unicast problem where non-linear network code is involved was established in [36]; however, this network does not satisfy our criteria.

- certain Circulant graphs.

The guessing numbers of perfect graphs are well understood. The calculation of the guessing number of the n-cycle and its complement are completed independently by Christofides and Markström [28], and Blasiak, Kleinberg, and Lubetzky [12]. The first computation for C_5 was established in [3, 8, 81]. The cases of 3-regular Cayley graphs of $\mathbb{Z}/n\,\mathbb{Z}$ of special type and certain Circulant graphs, the coding rate are computed instead [12], but we can easily transfer their result into the setting a guessing game.

In this work, we calculated guessing numbers for many new families of undirected graphs. None of these graphs are included in any of the families above. The list of new families and their calculations can be found in Section 5.1.

Noisy guessing games (Chapter 6)

Thus far, the benefits achieved by network coding (efficiency, safety) are based on one important assumption: the communication channels are error-free. What if some channels are error-prone?

In this situation, we would have to face the problem of error propagation: a single corrupted packet potentially corrupts all packets received at the terminals. One way to deal with this is to apply the classical theory of error correcting codes which adds redundancy to the transmission in the time domain and hence increases the success rate of each communication channel. However, this method leaves out the information about network topology, which essentially is the source of all properties of network coding. In order to gain the full benefits of network coding, we need some methods to add redundancy representing the spatial domain; that is, an existence of network error correction framework.

In the case of multicast network, a theory proposed by Cai and Yeung [21, 96, 22] extends all the knowledge of classical coding theory, including code distance, weight measures, and coding bounds. A special attribute of this framework is that the topological properties of a given network is encoded within the error correcting mechanism to improve the noisy coding capacity. Based on this framework, various algorithmic constructions of network error correcting codes have been studied in [54, 92, 6, 74, 47, 53, 62, 17, 43, 46, 86, 100, 63, 101].

In the case of general network information flow problems, in contrast, a convinced framework for network error correction remains elusive. The obstacle exists due to the enigmatic interaction between coding function and noises in the network. In this chapter, we initialize studies of interaction between coding functions and noises restricted to our setting of guessing games. We proposed a definition of noisy guessing numbers, which is a generalized version of the noiseless guessing number introduced by Riis [82]. First few properties of this quantity together with show cases on undirected graphs of small order are also established. The definition and properties of noisy guessing numbers are proposed and proved in Section 6.2. Section 6.3 demonstrates the computation of this quantity for graphs K_2, P_3, and K_3.

1.3 Thesis structure

The organization of the thesis is as follows:

Chapter 2

In this chapter, we recall basic terminologies in discrete probability theory and information theory. The aim is to provide the reader a rigorous yet gentle introduction to the existing development of information theory. We aim to gather fundamental concepts of information measures and their properties together with the newest progressions in characterizing the space of all entropic vectors of n jointly distributed random variables in one place. Specifically, we hope that this chapter will give the reader a clearer view about the method of deriving different families of information inequality.

Chapter 3

Chapter 3 introduces the formal language of guessing games and provides rigorous proofs of various properties of a guessing number. In particular, we show that the definition of guessing number has asymptotic behaviour. We also recall different known approaches to calculate lower-bounds of a guessing number including the code graph [45, 28], and the fractional clique cover strategy defined in [28], which provides a feasible computational method for calculating lower bounds of guessing numbers for undirected graphs. A method for calculating upper bounds of guessing numbers by making use of entropic arguments is also introduced in this chapter. The theory presented in this chapter mainly follows the paper by Christofides and Markström [28].

Chapter 4, Chapter 5, and Chapter 6

These chapters present the details of our results discussed in Section 1.2.

Chapter 7

We conclude the thesis with some open problems and future research directions.

Chapter 2

Information Measures and Information Inequalities

In order to make this thesis self-contained, this chapter recalls some basic terminologies in discrete probability theory and information theory. It consists of two parts. The first part is a summary of fundamental concepts in information theory and their properties. Based on the information given in the first part, we will build up a small survey on recent progress in characterizing the space of all entropic vectors of n jointly distributed random variables, especially, we will focus on constructing different families of information inequality.

The content of this chapter is synthesized from several sources, e.g. *None of the results in this chapter were discovered by the author of this thesis*; a couple of proofs for well-known results, e.g. Copy Lemma, Zhang-Yeung Inequality, are selected from literatures.

2.1 Entropy and entropic functions

2.1.1 Shannon's information measures

Throughout this thesis, we will only consider entropy of discrete random variables, i.e. random variables with co-domains that are discrete sets. Let X be a random variable with codomain $\mathcal{X}$. We write p_X for $\{p_X(x) = \mathbf{P}[X = x], x \in \mathcal{X}\}$ – the probability distribution of X. When there is no ambiguity, we abbreviate $p_X(x)$ as $p(x)$.

We write Supp(X) for the support of X. Supp(X) is the set of all $x \in \mathcal{X}$ with positive probability, i.e. $p(x) > 0$. If Supp(X) $= \mathcal{X}$, we say that p_X is strictly positive.

Definition 3. *The entropy $H(X)$ of a random variable X is defined by*

$$H(X) = -\sum_{x} p(x) \log_s p(x),$$

where $s > 1$ is a chosen real number.

Note that in the above definition, in the case $p(x) = 0$ the value of $p(x) \log p(x)$ is undefined. However, due to the continuity property of the entropy map (see [95, Chapter 1, Section 1.3]), it is safe to adopt the convention $p(x) \log(x) = 0$ if $p(x) = 0$, or equivalently, the summation is taken over the support of X.

We observe that the entropy $H(X)$ of a random variable X is a function of the probability distribution $p(x)$ and so it depends only on $p(x)$ but not on the range of $\mathcal{X}$. Intuitively, $H(X)$ measures our uncertain of the value of random variable X.

We recall the definition of other Shannon's information measures, which are basically linear combinations of entropy.

Definition 4. *Let $X_1, \ldots, X_n$ be n jointly distributed random variables. The joint entropy $H(X_1, \ldots, X_n)$ is defined by*

$$H(X_1, \ldots, X_n) = -\sum_{x_1, \ldots, x_n} p(x_1, \ldots, x_n) \log p(x_1, \ldots, x_n).$$

Definition 5. *For random variables X and Y, the conditional entropy of X given Y is defined by*

$$H(X|Y) = -\sum_{x,y} p(x, y) \log p(x|y).$$

Straightforward calculation shows that

Proposition 1.

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y).$$

For two random variables X and Y, we can define the mutual information $I(X, Y)$ which measures the amount of information of one random variable by provided the other.

Definition 6. *For random variables X and Y, the mutual information between X and Y is defined by*

$$I(X; Y) = \sum_{x,y} \log \frac{p(x, y)}{p(x)p(y)}.$$

Proposition 2. • *$I(X;Y) = I(Y;X)$, i.e. mutual information is symmetrical in X and Y.*

- *$I(X;X) = H(X)$.*

and

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y). \end{aligned}$$

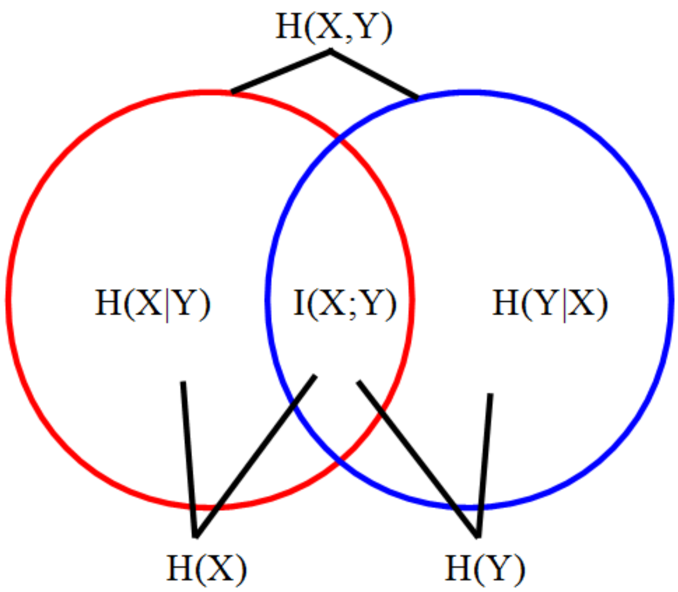


Figure 2.1: Relationship between entropies and mutual information for two random variables [95].

Similar to conditional entropy, one can define conditional mutual information of random variables X and Y given random variable Z.

Definition 7. *For random variables X, Y and Z, the mutual information between X and Y conditioning on Z is defined by*

$$I(X;Y|Z) = \sum_{x,y,z} p(x,y,z) \log \frac{p(x,y|z)}{p(x|z)p(y|z)}.$$

As one might expect, conditional mutual information satisfies the same set of relations for mutual information except that all the terms are now conditioned on a random variable Z.

Proposition 3.

$$
\begin{aligned}
I(X;X|Z) &= H(X|Z),\\
I(X;Y|Z) &= H(X|Z) - H(X|Y,Z)\\
&= H(Y|Z) - H(Y|X,Z)\\
&= H(X|Z) + H(Y|Z) - H(X,Y|Z).
\end{aligned}
$$

There is another important information measure known as the Kullback-Leibler divergence which indicates the difference between two probability distributions P and Q. It is a non-symmetric measure which establishes the amount of information lost when approximating P by Q. The definition of this measure is not recalled here as it is irrelevant to the content of this thesis.

2.1.2 Entropic vectors

Let $[n] = \{1, 2, \ldots, n\}$ where $n \geq 2$ be an index set. For each non-empty subset $\alpha = \{i_1, \ldots, i_m\} \subseteq [n]$, we write $X_{\{\alpha\}}$ for the set of jointly distributed random variables $\{X_{i_1}, \ldots, X_{i_m}\}$, and we write the joint entropy of $\{X_{i_1}, \ldots, X_{i_m}\}$ as $H(X_\alpha)$.

Definition 8. *Let $[n] = \{1, 2, \ldots, n\}$ where $n \geq 2$ be a finite set. We treat the set of all non-empty subsets of $[n]$ as an index set. A vector $\epsilon = [\epsilon_\alpha | \alpha \subseteq [n]] \in \mathbb{R}^{2^n - 1}$ is an entropic vector if there exist a set of jointly distributed random variables $X_{[n]} := \{X_{i_1}, \ldots, X_{i_m}\}$ such that $\epsilon_\alpha = H(X_\alpha)$ for all $\alpha \subseteq [n]$.*

We write Γ_n^* for the collection of all entropic vectors of n random variables and its closure is denoted as $\bar{\Gamma}_n^*$. Elements of Γ_n^* are also referred to as entropic functions.

Example 1. *For $n = 2$, $\epsilon = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$ is an entropic vector as the following probability distribution over binary alphabet could be given*

	$X_2 = 0$	$X_2 = 1$
$X_1 = 0$	0.5	0
$X_1 = 1$	0	0.5

One of the main goals in information theory is to characterize the region of Γ_n^* for positive integer n. Even though many aspects of Γ_n^* for $n \geq 4$ are far from completely understood, substantial progress has been made. These results will be presented in the next section.

2.2 The region of Γ_n^*

We first point out a few basic properties of Γ_n^* which are direct consequences of its definition.

- Let $X_1, \ldots, X_n$ be n degenerate random variables taking constant values, hence $H(X_\alpha) = 0$ for all α. This implies that Γ_n^* contains the origin.
- By definition, $H(X_\alpha)$ is always non-negative, hence we have $\Gamma_n^* \subseteq \{\epsilon | \epsilon_\alpha \geq 0 \text{ for all non-empty } \alpha \in 2^{[n]}\} \subseteq \mathbb{R}^{2^n-1}$.

In order to further characterize Γ_n^*, an approach of constructing outer bounds of Γ_n^* via unconstrained information inequalities is generally adopted. Therefore, generating such information inequalities is the main topic of this section.

Given an information expression – a linear combination of Shannon's information measures which involves a finite number of random variables, e.g. $H(ABC)+H(C)-H(AB)+H(AC)$, we can ask if such an expression satisfies some inequalities, e.g. $H(ABC)+H(C)-H(AB)+H(AC) \geq 0$. If an inequality exists and it holds for any joint distributions of random variables appeared in the given expression, then we regard such inequality as an information inequality.

Recall from the previous section that any Shannon's information measures other than joint entropy can be expressed as a linear combination of joint entropies by applying the following identity:

$$I(X;Y|Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z).$$

Thus, any information expression involving n random variables can be written as a linear combination of $2^n - 1$ associated joint entropies. We refer the latter expression as the *canonical form* of an information expression. We might wonder whether a canonical form of an information expression is unique. The answer to this question is yes, and it is a corollary of the following theorem:

Theorem 2. *Let f be an information expression. The unconstrained information identity $f = 0$ always holds if and only if f is the zero function.*

The idea of the proof is that assuming f is a non-zero function, the set $f = 0$ is a hyperplane in $\mathbb{R}^{2^n-1}$ which has Lebesgue measure 0. Now we can show that the volume of Γ_n^* has non-zero Lebesgue measure, hence Γ_n^* cannot be contained in the hyperplane $f = 0$. Therefore f must be a zero function. We omit detail of the proof, and recommend readers to refer to [95, Chapter 12, Section 12.2].

Corollary 1. *The canonical form of an information expression is unique.*

Thanks to this corollary, it is justified to make the following definition

Definition 9. *Let n be a positive integer, and let $\alpha_1, \ldots, \alpha_k$ be the non-empty subsets of $\{1, \ldots, n\}$. Let $a_i \in \mathbb{R}$ for $1 \leq i \leq k$. An inequality of the form*

$$a_1 H(\{X_i : i \in \alpha_1\}) + \ldots + a_k H(\{X_i : i \in \alpha_k\}) \geq 0$$

is called an information inequality if it holds for all jointly distributed random variables $X_1, \ldots X_n$.

Shannon proved the basic inequalities[85]:

$$H(A) \geq 0 \tag{2.1}$$

$$H(A|B) \geq 0 \tag{2.2}$$

$$I(A; B) \geq 0. \tag{2.3}$$

These inequalities are all special cases of the inequality

$$I(A; B|C) = H(AC) + H(BC) - H(C) - H(ABC) \geq 0 \tag{2.4}$$

For example, to obtain $I(A; B) \geq 0$ we set C to be a degenerate random variable taking constant value, hence

$$\begin{aligned} 0 &\leq H(AC) + H(BC) - H(C) - H(ABC) \\ &= H(A) + H(B) - 0 - H(AB) \\ &= I(A; B) \end{aligned}$$

We can generate new inequalities by combining instances of these basic inequalities. For example,

$$\frac{I(A; B|C)}{2} + I(A; C) \geq 0$$

is a valid information inequality which holds for any jointly distributed random variables A, B, C. We can rewrite this inequality in its canonical form:

$$\begin{aligned} 0 &\leq \frac{H(AC) + H(BC) - H(ABC) - H(C)}{2} + H(A) + H(C) - H(AC) \\ &= \frac{2H(A) + H(C) + H(BC) - H(ABC) - H(AC)}{2} \end{aligned}$$

Also we can permute the variables, e.g. $A \to B$, $B \to C$, $C \to A$ to obtain a "new" information inequality:

$$0 \leq 2H(B) + H(C) + H(AC) - H(ABC) - H(AB)$$

For the rest of this section, we will identify inequalities that can be obtained from another using information identity, permutation of random variables, and a combination of existing inequalities by algebraic operators.

The inequality $H(AB) + H(BC) - H(ABC) - H(C) \geq 0$ proved by Shannon in his monument paper [85] is historically important. This inequality stood alone for almost half a century as the fundamental inequality. All information inequalities known in the literature are consequences of this inequality [95]. We call information inequalities which can be derived by adding special cases of inequality 2.4 as Shannon's type inequality.

Definition 10. *A Shannon information inequality is any information inequality of the form* X_i

$$\sum_i \alpha_i I(A_i; B_i | C_i) \geq 0 \tag{2.5}$$

where each $\alpha_i \geq 0$.

From the previous discussion, we see that every information inequality can be expressed in its canonical form

$$\sum_{\alpha \subseteq [n]} a_\alpha H(X_\alpha) = \mathbf{a}^\top \mathbf{H} \geq 0,$$

and the set of all vectors in $\mathbb{R}^{2^n-1}$ satisfying such inequality corresponds to the half-space

$$\mathcal{H}_{n,\mathbf{a}} := \{\mathbf{u} \in \mathbb{R}^{2^n-1} | \mathbf{a}^\top \mathbf{h} \geq 0\}.$$

We denote $\Gamma_n := \cap_{\mathbf{a}} \mathcal{H}_{n,\mathbf{a}}$ the set of vectors $u \in \mathbb{R}^{2^n-1}$ for which u satisfies all Shannon's type inequalities. We can check if an unconstrained inequality $\mathbf{a}^\top \mathbf{h} \geq 0$ is a Shannon-type inequality by examining whether Γ_n is a subset of $\{\mathbf{u} \in \mathbb{R}^{2^n-1} | \mathbf{a}^\top \mathbf{h} \geq 0\}$. The seminal work of Yeung [95] showed that this verification procedure can be formulated as a linear programming problem if the number of random variables is fixed.

Theorem 3. *[95] An expression* $\mathbf{a}^\top \mathbf{h} \geq 0$ *is a Shannon-type inequality if and only if the minimum of the problem*

$$\textit{Minimize } \mathbf{a}^{\top}\mathbf{h} \textit{ subjected to } \mathbf{Gh} \geq 0,$$

where $\mathbf{G}$ *is a* m *by* $2^n - 1$ *matrix representing all elementary inequalities in canonical forms, is zero. In this case, the minimum occurs at the origin.*

Theorem 3 enables a machine-proving approach to all Shannon-type inequalities. Implementations of this algorithm are made freely available online, e.g. the Information Theory Inequality Prover (ITIP) by Yeung and Yan [94], or Xitip developed by Pulikkoonattu, Perron and Diggavi [79].

We refer any information inequality that cannot be expressed in the form 2.4 as a non-Shannon information inequality.

We have the following results concerning the region of Γ_n^* for $n < 4$.

Theorem 4.

$$\Gamma_2^* = \Gamma_2.$$

Theorem 5.

$$\Gamma_3^* \subsetneq \bar{\Gamma}_3^* = \Gamma_3.$$

A corollary of these theorems is that all information inequalities involving fewer than four random variables are Shannon inequalities.

Unlike cases $n = 2$ or $n = 3$, less is known about the region of Γ_n^* when the number of random variables is at least 4. However, we have the following characteristic:

Theorem 6. $\bar{\Gamma}_n^*$ *is convex.*

The first non-Shannon-type information inequality was discovered by Zhang and Yeung in their seminal work [98], [95, Theorem 14.7 on p.310].

Theorem 7. *The following is a* 4*-variables non-Shannon information inequality:*

$$2I(C; D) \leq I(A; B) + I(A; C, D) + 3I(C; D|A) + I(C; D|B). \tag{2.6}$$

The proof of Theorem 7 consists of two parts. The first part proves that the inequality is valid for every joint distribution on 4-random variables, and the second part illustrates that it cannot be expressed in the form of 2.4.

To prove 2.6 is a valid information inequality, Zhang and Yeung introduced a basic technique which until now is the only known approach to come up with new information inequalities. Their method can be summarized as follows:

1 Start with a set of arbitrary random variables.

2 Add auxiliary random variables with special properties.

3 Apply known information inequalities to the enlarged set of random variables.

The most important part is Step 2 which is encapsulated in the Copy Lemma [98].

Lemma 1 (Copy Lemma)**.** *Let A, B, C, D be jointly distributed random variables. There is another random variable R, jointly distributed with A, B, C, D with the following properties.*

1. *The marginal distributions of (A, B, C) and (A, B, R) are the same with R replacing C.*

2. *$I(CD; R|AB) = 0$.*

In this case we say that R is a D-copy of C over (A, B).

The proof is taken from [38].

Proof. Let $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, denote the alphabets of the random variables A, B, C, D resp. Let a, b, c, d denote arbitrary elements of $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\mathcal{D}$, resp. with probability $p(a, b, c, d)$. Let R be a new random variable and let r denote an arbitrary element of its alphabet, which is $\mathcal{C}$. Define the joint probability distribution of A, B, C, D, R by

$$p'(a, b, c, d, r) = \frac{p(a, b, c, d) \sum_d p(a, b, r, d)}{\sum_{c,d} p(a, b, c, d)}.$$

It is clear that these are nonnegative. Summing over r we get

$$\begin{aligned}\sum_r p'(a, b, c, d, r) &= \frac{p(a, b, c, d) \sum_{r,d} p(a, b, r, d)}{\sum_{c,d} p(a, b, c, d)} \\ &= p(a, b, c, d).\end{aligned}$$

so that p' is an extension of p, which also implies that the sum of all of the probabilities p' is 1. Similarly, the marginal distribution of (A, B, R) is given by

$$\begin{aligned}\sum_{c,d} p'(a, b, c, d, r) &= \frac{\sum_{c,d} p(a, b, c, d) \sum_d p(a, b, r, d)}{\sum_{c,d} p(a, b, c, d)} \\ &= \sum_d p(a, b, r, d)\end{aligned}$$

while the marginal distribution of (A, B, C) is given by $\sum_d p(a, b, c, d)$, demonstrating Condition 1.

If we write Condition 2 in terms of entropies, we get

$$H(ABCD) + H(ABR) - H(AB) - H(ABCDR) = 0.$$

But $H(A, B, R) = H(A, B, C)$ by Condition 1, so it remains to show that

$$H(ABCDR) = H(ABCD) + H(ABC) - H(AB).$$

We compute $H(ABCDR)$ as

$$\begin{aligned} H(ABCDR) &= \sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log p'(a,b,c,d,r) \\ &= \sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log p(a,b,c,d) \\ &+ \sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log \sum_{d} p(a,b,r,d) \\ &- \sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log \sum_{c,d} p(a,b,c,d). \end{aligned}$$

But

$$\begin{aligned} &\sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log p(a,b,c,d) \\ &\qquad = \sum_{a,b,c,d} -p(a,b,c,d) \log p(a,b,c,d) \\ &\qquad = H(ABCD) \end{aligned}$$

$$\begin{aligned} &\sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log \sum_{d} p(a,b,r,d) \\ &\qquad = \sum_{a,b,r} \sum_{c,d} -p'(a,b,c,d,r) \log \sum_{d} p(a,b,r,d) \\ &\qquad = \sum_{a,b,r} \sum_{d} -p(a,b,r,d) \log \sum_{d} p(a,b,r,d) \\ &\qquad = H(ABC) \end{aligned}$$

and

$$\begin{aligned} &\sum_{a,b,c,d,r} -p'(a,b,c,d,r) \log \sum_{c,d} p(a,b,c,d) \\ &\qquad = \sum_{a,b,} \sum_{c,d,r} -p'(a,b,c,d,r) \log \sum_{c,d} p(a,b,c,d) \\ &\qquad - \sum_{a,b,} \sum_{d,r} -p(a,b,r,d) \log \sum_{c,d} p(a,b,c,d) \\ &\qquad = H(AB). \end{aligned}$$

Therefore, $H(ABCDR) = H(ABCD) + H(ABC) - H(AB)$ as desired. $\square$

Proof of Theorem 7. By expanding mutual informations into entropies and cancelling terms, one can verify the following identities

$$\begin{aligned} I(A;B) &+ I(C;R|A) + I(C;D|R) + I(AB;R|CD) \\ &+ I(D;R|B) + I(A;B|RD) + I(D;R|A) \\ &+ I(R;C|B) + I(A;B|CR) + I(C;R|ABD) \\ &= 2I(A;B|C) + I(A;C|B) + I(B;C|A) \\ &+ I(A;B|D) + I(C;D) \\ &+ 2I(CD;R|AB) \\ &+ I(A;B|R) - I(A;B|C) \\ &+ I(A;R|B) - I(A;C|B) \\ &+ I(B;R|A) - I(B;C|A). \end{aligned}$$

Each of the conditional mutual information terms on the left-hand-side are non-negative by Shannon's inequalities. Thus, if these terms except $I(A;B)$ are erased and the "=" is replaced by "$\leq$", then we obtain a 5-variable Shannon-type inequality.

By the Copy Lemma we may choose R to be a D-copy of C over AB. Then, the term $2I(CD;R|AB)$ is zero by Condition 2, and each pair

$$I(A;B|R) - I(A;B|C) \tag{2.7}$$

$$I(A;R|B) - I(A;C|B) \tag{2.8}$$

$$I(B;R|A) - I(B;C|A). \tag{2.9}$$

vanishes by Condition 1.

Hence 2.6 is a valid inequality for 4-random variables.

To show that 2.6 is not a Shannon's type inequality, we can use the ITIP package discussed above. □

Corollary 2.

$$\Gamma_4^* \subset \bar{\Gamma}_4^* \subsetneq \Gamma_4.$$

Since the seminal work [98], many other non-Shannon information inequalities have been discovered. To name a few: Lněnička [69], Makarychev, Makarychev, Romashchenko, and Vereshchagin [72], Zhang [99], Zhang and Yeung [102], Mátus [75], Chan and Grant [26], Xu, Wang, and Sun [91], Dougherty, Freiling, and Zeger [35] [38].

Mátus constructed the first two infinite families of non-Shannon inequalities in [75]. He indexed elements in each of these families by the positive integers. A clever manipulation of elements in the first list was used to show that no finite collection of linear inequalities will ever be able to describe Γ_n^* completely. Based on the list provided in [75], Chan and Grant derived the first non-trivial non-linear information inequality [26].

Xu-Wang-Sun added new information inequalities and a third infinite list in [91].

Dougherty, Freiling, and Zeger [38] combined exhaustive computer search and Zhang-Yeung method to derive a new set of information inequalities which can be used to derive information inequalities known in the literatures. They also presented a set of rules for combining new inequalities from old ones which can be used iteratively to generate uncountable collections of information inequalities. As an example, they showed that the first list of inequalities provided by Matus and the third list produced by Xu-Wang-Sun can be derived from this process.

Chapter 3

Guessing Games

3.1 Definitions

We start out by recalling some formal terminologies in graph theory that will be used throughout the rest of this thesis. We follow standard graph theoretic language introduced in standard text books such as [16] and [32].

Definition 11. *A* directed graph*, or* digraph *for short, is a pair* $G = (V(G), E(G))$*, where* $V(G)$ *is the set of* vertices *of* G *and* $E(G)$ *is a set of ordered pairs of vertices of* G *called the* directed edges *of* G*. Given a directed edge* $e = (u, v)$*, which we also denote by* $\overrightarrow{uv}$*, we call* u *the* tail *and* v *the* head *of* e *and say that* e *goes from* u *to* v*.*

For the purposes of guessing games we will assume throughout that our digraphs are loopless, i.e. they contain no edges of the form $\overrightarrow{uu}$ for $u \in V(G)$.

Given a digraph G and a vertex $v \in V(G)$, the *in-neighbourhood* of v is the set of all vertices u in $V(G)$ such that $\overrightarrow{uv}$ is an edge of G, i.e. $\Gamma^-(v) = \{u : \overrightarrow{uv} \in E(G)\}$. Similarly, we define the *out-neighbourhood* of a vertex v to be the set of all vertices u of G such that $\overrightarrow{vu}$ is a directed edge, i.e. $\Gamma^+(v) = \{u : \overrightarrow{vu} \in E(G)\}$.

In this thesis, our main results will primarily be on *undirected graphs* which are naturally treated as a special type of digraph G where $\overrightarrow{uv} \in E(G)$ if and only if $\overrightarrow{vu} \in E(G)$. We call the pair of directed edges $\overrightarrow{uv}$ and $\overrightarrow{vu}$, the *undirected edge* uv.

A major role in our guessing strategies will be played by *cliques* i.e. subgraphs in which every pair of vertices are joined by an undirected edge. We will use standard notations for the maximum degree and the independent, clique, chromatic, and fractional chromatic numbers of an undirected graph as $\alpha(G)$, $\Delta(G)$, $\omega(G)$, $\chi(G)$, and

$\chi_f(G)$, respectively. (see [32] for definitions of these parameters).

Given a digraph G and an integer $t \geq 1$, the *t-uniform blowup of G* which we will write as $G(t)$ is a digraph formed by replacing each vertex v in G with a class of t vertices $v_1, \ldots, v_t$ with $\overrightarrow{u_i v_j} \in E(G(t))$ if and only if $\overrightarrow{uv} \in E(G)$.

Definition 12. *A* guessing game (G, s) *is a game played on a digraph G and the alphabet $A_s = \{0, 1, \ldots, s-1\}$. There are $|V(G)|$ players working as a team. Each player corresponds to one of the vertices of the digraph. Each player v is assigned an integer x_v from A_s uniformly and independently at random. Each player will be given a list of the players in its in-neighbourhood with their corresponding values. Using just this information each player must guess their own value. If all players guess correctly they will all win, but if just one player guesses incorrectly they will all lose.*

Throughout this work we will be freely speaking about the player v instead of the player corresponding to the vertex $v \in V(G)$.

Definition 13. *Given a guessing game (G, s), for $v \in V(G)$ a* strategy for player v *is formally a function $f_v : A_s^{|\Gamma^-(v)|} \to A_s$ which maps the values of the in-neighbours of v to an elements of A_s, which will be the guess of v. A* strategy $\mathcal{F}$ *for a guessing game is a sequence of such functions $(f_v)_{v \in V(G)}$ where f_v is a strategy for player v.*

We denote by $\mathrm{Win}(G, s, \mathcal{F})$ *the event that all the players guess correctly when playing (G, s) with strategy $\mathcal{F}$.*

Obviously, it is impossible for players to guess correctly every round. However, from probabilistic point of view, we can restrict the objective into finding a strategy $\mathcal{F}$ that maximises $\mathbf{P}[\mathrm{Win}(G, s, \mathcal{F})]$.

Example 2. *As an example we consider the guessing game (K_n, s), where K_n is the* complete (undirected) graph *of order n, i.e. $|V(K_n)| = n$ and $E(K_n) = \{uv : u, v \in V(G), u \neq v\}$. Naively we may think that since each player receives no information about their own value that each player may as well guess randomly, meaning that the probability they win is s^{-n}. This however is not optimal. Certainly the probability that any given player guesses correctly is $1/s$, but Riis [82] noticed that by discussing their strategies beforehand the players can in fact coordinate the moments where they guess correctly, and therefore increase their chance of winning. For example before the game begins they can agree that they will all play under the assumption that*

$$\sum_{v \in V(K_n)} a_v \equiv 0 \bmod s. \tag{3.1}$$

Player u can see all the values except its own, and assuming (3.1) is true it knows that

$$a_u \equiv - \sum_{\substack{v \in V(K_n) \\ v \neq u}} a_v \bmod s.$$

Consequently player u will guess that its value is $-\sum_{v \in V(K_n), v \neq u} a_v \mod s$. *Hence if (3.1) is true every player will guess correctly and if (3.1) is false every player will guess incorrectly. So the probability they all guess correctly is simply the probability that (3.1) is true which is* $1/s$. *This is clearly optimal as, irrespective of the strategy, the probability that a single player guesses correctly is* $1/s$ *and so we can not hope to do better.*

We note that the optimal strategy given in the example was a *pure strategy* i.e. there is no randomness involved in the guess each player makes given the values it sees. The alternative is a *mixed strategy* in which the players randomly choose a strategy to play from a set of pure strategies. The winning probability of the mixed strategy is the average of the winning probabilities of the pure strategies weighted according to the probabilities that they are chosen. This however is at most the the winning probability of an optimal pure strategy as proved by the following lemma.

Lemma 2. *Every randomised strategy for the guessing game* (G, s) *has winning probability at most* $\mathbf{P}[\mathrm{Win}(G, s, \mathcal{F}_{\mathrm{opt}})]$, *where* $\mathcal{F}_{\mathrm{opt}}$ *is an optimal pure guessing strategy.*

Proof. Following our previous paragraph, a randomised strategy $\mathcal{G}$ can be described by assigning a probability $\mathbf{P}[\mathcal{G} = \mathcal{F}])$ to each deterministic strategy $\mathcal{F}$. The winning probability of such a strategy is

$$\begin{aligned} \mathbf{P}[\mathrm{Win}(G, s, \mathcal{G})] &= \sum_{\mathcal{F}} \mathbf{P}[\mathcal{G} = \mathcal{F}]\mathbf{P}[\mathrm{Win}(G, s, \mathcal{F})] \\ &\leq \max_{\mathcal{F}} \mathbf{P}[\mathrm{Win}(G, s, \mathcal{F})] = \mathbf{P}[\mathrm{Win}(G, s, \mathcal{F}_{\mathrm{opt}})] \end{aligned}$$

□

Therefore we gain no advantage by playing a mixed strategy. As such throughout this thesis we will only ever consider pure strategies.

We now define the guessing number of guessing game (G, s) which will be our measure of the winning probability obtained by an optimal strategy for (G, s).

Definition 14. *The guessing number* $\text{gn}(G, s)$ *of a guessing game* (G, s) *is the largest* β *such that there exists a strategy* $\mathcal{F}$ *for* (G, s) *satisfies that every player* v *guesses its own value* x_v *correctly with probability* $\frac{1}{s^{n-\beta}}$. *In other words,*

$$\text{gn}(G, s) = |V(G)| + \log_s \left(\max_{\mathcal{F}} \mathbf{P}[\text{Win}(G, s, \mathcal{F})] \right).$$

Although this looks like a cumbersome property to work with we can think of it as a measure of how much better the optimal strategy is over the strategy of just making random guesses, as

$$\max_{\mathcal{F}} \mathbf{P}[\text{Win}(G, s, \mathcal{F})] = \frac{s^{\text{gn}(G,s)}}{s^{|V(G)|}}.$$

Later we will look at information entropy inequalities as a way of analyzing the guessing game and in this context the definition of the guessing number will appear more natural.

Example 3. *In our previous example of guessing game* (K_n, s), *the guessing number is* $\text{gn}(K_n, s) = n - 1$.

3.2 The asymptotic guessing number

Note that the guessing number of the example (K_n, s) we discussed earlier is represented by $\text{gn}(K_n, s) = n - 1$ which does not depend on s. In general $\text{gn}(G, s)$ will depend on s and it is often extremely difficult to determine the guessing number exactly. Consequently we will instead concentrate our efforts on evaluating the *asymptotic guessing number* $\text{gn}(G)$ which we define to be the limit of $\text{gn}(G, s)$ as s tends to infinity. To prove the limit exists we first need to consider the guessing number on the blowup of G.

Lemma 3. *Given a digraph* G, *and integers* $s, t \geq 1$,

$$\max_{\mathcal{F}} \mathbf{P}[\text{Win}(G(t), s, \mathcal{F})] \geq \left(\max_{\mathcal{F}} \mathbf{P}[\text{Win}(G, s, \mathcal{F})] \right)^t$$

or equivalently $\text{gn}(G(t), s) \geq t\, \text{gn}(G, s)$.

Proof. The digraph $G(t)$ can be split into t vertex disjoint copies of G. We can construct a strategy for $(G(t), s)$ by playing the optimal strategy of (G, s) on each of the t copies of G in $G(t)$. The result follows immediately. □

Lemma 4. *Given a digraph G, and integers $s, t \geq 1$,*

$$\max_{\mathcal{F}} \mathbf{P}[\text{Win}(G(t), s, \mathcal{F})] = \max_{\mathcal{F}} \mathbf{P}[\text{Win}(G, s^t, \mathcal{F})]$$

or equivalently $\text{gn}(G(t), s) = t\,\text{gn}(G, s^t)$.

Proof. First we will show that the optimal probability of winning on (G, s^t) is at least that of $(G(t), s)$. This follows simply from the fact that the members of the alphabet of size s^t, can be represented as t digit numbers in base s. Hence given a strategy on $(G(t), s)$, a corresponding strategy can be played on (G, s^t) by each player pretending to be t players: More precisely, if player v gets assigned value $a \in A_{s^t}$, he writes it as $a_{t-1} \cdots a_1 a_0$ in base s and pretends to be t players, say $v_0, v_1, \ldots, v_{t-1}$, where player v_i, for $0 \leq i \leq t-1$, gets assigned value $a_i \in A_s$. Furthermore, if player v sees the outcome of player u, then he can construct the values assigned to the new players $u_0, u_1, \ldots, u_{t-1}$. So these new fictitious players can play the $(G(t), s)$ game using an optimal strategy. But if the fictitious players can win the $(G(t), s)$ game then the original players can win the (G, s^t) game as we can reconstruct the value of a from the values of $a_0, a_1, \ldots, a_{t-1}$.

A similar argument can be used to show that the optimal probability of winning on (G, s^t) is at most that of $(G(t), s)$. We will show that for every strategy on (G, s^t) there is a corresponding strategy on $(G(t), s)$. Every vertex class of t players can simulate playing as one fictitious player by its members agreeing to use the same strategy. The t values assigned to the players in the vertex class can be combined to give an overall value for the vertex class. The strategy on (G, s^t) can then be played allowing the members of the vertex class to make a guess for the overall value assigned to the vertex class. This guess will be the same for each member as they all agreed to use the same strategy and have access to precisely the same information. Once the guess for the vertex class is made its value can be decomposed into t values from A_s which can be used as the individual guesses for each of its members. □

Using these results about blowups of digraphs we can show that in some sense the guessing number is almost monotonically increasing with respect to the size of the alphabet.

Lemma 5. *Given any digraph G, positive integer s, and real number $\varepsilon > 0$, there exists $t_0(G, s, \varepsilon) > 0$ such that for all integers $t \geq t_0$*

$$\text{gn}(G, t) \geq \text{gn}(G, s) - \varepsilon.$$

Proof. We will prove the result by showing that

$$\operatorname{gn}(G,t) \geq \frac{\lfloor \log_s t \rfloor}{\log_s t} \operatorname{gn}(G,s) \tag{3.2}$$

holds for all $t \geq s$. This will be sufficient since as t increases the right hand side of (3.2) tends to $\operatorname{gn}(G,s)$.

We will prove (3.2) by constructing a strategy for (G,t). Let $k = \lfloor \log_s t \rfloor$ and note that s^k is at most t. By considering only strategies in which every player is restricted to guess a value in $\{0,1,\ldots,s^k-1\}$ we get

$$\max_{\mathcal{F}} \mathbf{P}[\operatorname{Win}(G,t,\mathcal{F})] \geq \mathbf{P}[a_v < s^k \text{ for all } v \in V(G)] \max_{\mathcal{F}} \mathbf{P}[\operatorname{Win}(G,s^k,\mathcal{F})].$$

Hence

$$\frac{t^{\operatorname{gn}(G,t)}}{t^{|V(G)|}} \geq \left(\frac{s^k}{t}\right)^{|V(G)|} \frac{s^{k\operatorname{gn}(G,s^k)}}{s^{k|V(G)|}}$$

which rearranges to

$$\operatorname{gn}(G,t) \geq \frac{k}{\log_s t} \operatorname{gn}(G,s^k). \tag{3.3}$$

From Lemmas 3 and 4 we can show $\operatorname{gn}(G,s^k) \geq \operatorname{gn}(G,s)$ which together with (3.3) completes the proof of (3.2). □

Theorem 8. *For any digraph G, $\lim_{s\to\infty} \operatorname{gn}(G,s)$ exists.*

We denote the value $\lim_{s\to\infty} \operatorname{gn}(G,s)$ by $\operatorname{gn}(G)$.

Proof. By definition $\operatorname{gn}(G,s) \leq |V(G)|$ for all s, and $\max_{s\leq n} \operatorname{gn}(G,s)$ is an increasing sequence with respect to n, therefore its limit exists which we will call ℓ. Since $\operatorname{gn}(G,s) \leq \ell$ for all s it will be enough to show that $\operatorname{gn}(G,s)$ converges to ℓ from below.

By the definition of ℓ, given $\varepsilon > 0$ there exists $s_0(\varepsilon)$ such that $\operatorname{gn}(G,s_0(\varepsilon)) \geq \ell - \varepsilon$. From Lemma 5 we know that there exists $t_0(\varepsilon)$ such that for all $t \geq t_0(\varepsilon)$, $\operatorname{gn}(G,t) \geq \operatorname{gn}(G,s_0(\varepsilon)) - \varepsilon$ which implies $\operatorname{gn}(G,t) \geq \ell - 2\varepsilon$ proving we have convergence. □

Before we move on to the next section it is worth mentioning that for any s the guessing number $\operatorname{gn}(G,s)$ is a lower bound for $\operatorname{gn}(G)$. This follows immediately from Lemma 5. Furthermore for any strategy $\mathcal{F}$ on (G,s) we have

$$\operatorname{gn}(G,s) \geq |V(G)| + \log_s \mathbf{P}[\operatorname{Win}(G,s,\mathcal{F})].$$

Consequently we can lower bound the asymptotic guessing number by considering any strategy on any alphabet size.

3.3 The code graph

In the previous section, we have seen that the asymptotic guessing number is a unique value for each digraph G, and it is defined as the convergence of the almost increasing series: $\mathrm{gn}(G) = \lim_{s\to\infty} \mathrm{gn}(G, s)$. In this section, we investigate the value of $\mathrm{gn}(G, s)$ for each finite s in terms of maximal independent set of code graph $X(G, s)$ which was introduced independently by Gadoleau, Riis in [45], and Christofides, Markström [28]. This section is our attempt to collect important properties of code graphs that we will need for this thesis. As such, none of the results presented here were discovered by the author of this thesis, and the proofs of statements are often omitted or lifted almost verbatim from its original sources. The author thanks Dr. Gadoleau and Dr. Riis for letting him reproduce parts of their work here. We strongly recommend our readers to refer to the original paper by Gadoleau and Riis [45] for a beautiful and comprehensive treatment of code graphs[1].

Let G be a digraph on n vertices, and s be a positive integer, $s \geq 2$. We play the guessing game (G, s). According to our definition of guessing game, if the players agreed to follow a pure strategy $\mathcal{F} = (f_{v_1}, \ldots, f_{v_n})$, a game is won if the assigned value $(x_1, \ldots x_n)$ equals to the guessing value $(\hat{x_1}, \ldots, \hat{x_n})$ where each $\hat{x_i} = f_{v_i}(x_{j_1}, \ldots x_{j_k})$ is the guessing value of player i using strategy f_{v_i} given the value of its neighbourhoods. We call the set of all possible assignments $(x_1, \ldots x_n) \in A_s^n$ the space of configuration. For each pure guessing strategy $\mathcal{F}$, if a configuration $\mathbf{x} \in A_s^n$ satisfies that $\mathbf{x} = \mathcal{F}(\mathbf{x})$, then we call such configuration a fixed configuration of $\mathcal{F}$. We denote $\mathcal{A}_{\mathcal{F}}$ the set of all fixed configurations of $\mathcal{F}$.

We construct our code graph $X(G, s)$ having the space of configurations as its vertex set, and for each guessing strategy $\mathcal{F}$, the set of fixed configurations of $\mathcal{F}$ corresponds to an independent set of $X(G, s)$. As a result, the guessing number of (G, s) is equivalent to the logarithm of the independence number of $X(G, s)$.

Definition 15. *[45] Let (G, s) be a guessing game. The code graph $X(G, s)$ has A_s^n as vertex set and two vertices $\mathbf{x} = (x_1, \ldots x_n)$ and $\mathbf{x}' = (x_1', \ldots x_n')$ are adjacent if and only if there is no guessing strategy $\mathcal{F}$ for (G, s) which fixes them both, i.e. $\mathbf{x} = \mathcal{F}(\mathbf{x})$ and $\mathbf{x}' = \mathcal{F}(\mathbf{x}')$.*

Alternatively, we can take the following proposition as our definition of the code graph as it provides a concrete and elementary description of the edge set of $X(G, s)$.

[1] The graph $X(G, s)$ is called guessing graph in [45] and code graph in [28]. We choose the latter term to differentiate between guessing game of a graph and the code graph of such guessing game

Proposition 4. *[45] The code graph $X(G,s)$ has the following properties:*

1 $|V(X(G,s))| = s^n$.

2 The edge set of $X(G,s)$ is $E(X(G,s)) = \cup_{i=1}^n E_i(s)$, where $E_i(s) = \{\mathbf{xy} : x_{\Gamma^-(v_i)} = y_{\Gamma^-(v_i)}$ and $x_i \neq y_i\}$.

3 $X(G,s)$ is vertex-transitive. In particular, if $\mathbf{x}$ and $\mathbf{x}'$ is adjacent, we have

- *For any $\mathbf{e} \in A_s^n$, $\mathbf{x}+\mathbf{e} \mod (\mathbf{s})$ and $\mathbf{x}'+\mathbf{e} \mod (\mathbf{s})$ are also adjacent;*
- *For any $\pi \in \mathrm{Aut}(D)$, $\pi(x)$ is adjacent to $\pi(y)$.*
- *If we endow our alphabet the structure of a finite field $\mathbb{F}_s$ where s is power of a prime, then $\lambda_1 x_1, \ldots, \lambda_n x_n$ is adjacent to $\lambda_1 x_1', \ldots, \lambda_n x_n'$ where each λ_i is a non-zero element of $\mathbb{F}_s$.*

4 $X(G,s)$ is regular with degree

$$d(X(G,s)) = \sum_{I \text{ indepedent set of} G} (-1)^{|I|-1}(s-1)^{|I|} s^{n-|\Gamma^-(I)|-|I|},$$

where $\Gamma^-(I) = \cup_{v_i \in I} \Gamma^-(v_i)$.

Proof. Property 1 follows from our definition.

For Property 2, we see that if $\mathbf{xx}' \in E_i(s)$ for some i, then for any local guessing function f_{v_i} that fixes $\mathbf{x}$, we have $f_{v_i}(x'_{\Gamma^-(x_i)}) = f_{v_i}(x_{\Gamma^-(x_i)}) = x_i \neq x_i'$. Hence any $\mathcal{F} = (\ldots, f_{v_i}, \ldots)$ does not fix $\mathbf{x}'$. Conversely, if $\mathbf{xx}' \notin E(X(G,s))$, then any guessing strategy satisfying $f_{v_i}(x'_{\Gamma^-(x_i)}) = x_i'$ and $f_{v_i}(x_{\Gamma^-(x_i)}) = x_i$ for all v_i fixes both $\mathbf{x}$ and $\mathbf{x}'$.

Property 3 follows from the observation that $\mathbf{xx}' \in E(X(G,s))$ if and only if $(\mathbf{x} - \mathbf{x}')_{\Gamma(v_i)} = 0$ and $(x_i - x_i') \neq 0$ for some v_i.

Since every vertex-transitive graph is automatically regular. The degree of $X(G,s)$ therefore can be determined by the degree of $\mathbf{0} = (0,\ldots,0) \in A_s^n$. By the inclusion-exclusion principle, we have

$$d(X(G,s)) = d(\mathbf{0}) = |\cup_{i=1}^n |E_i(s) \cap \{\mathbf{0}\}| = \sum_{R \subseteq V(G)} (-1)^{|R|-1} |E_R \cap \{\mathbf{0}\}|,$$

where $E_R = \cap_{v_i \in R} E_i$, and hence we only have to determine $|E_R \cap \{\mathbf{0}\}|$ for all $R \subseteq V(G)$. The configuration $\mathbf{x}$ adjacent to $\mathbf{0}$ satisfy $\omega(\mathbf{x}_R) = |R|$ and $\mathbf{x}_{\Gamma(R)} = 0$, while $\mathbf{x}_{V-\Gamma(R)-R}$ is arbitrary. If R is not independent, $R \cap \Gamma(R) \neq \emptyset$ the two conditions are contradictory; otherwise $R \cap \Gamma(R) = \emptyset$ and there are $(s-1)^{|R|}s^{n-|\Gamma^-(R)|-|R|}$ choices for $\mathbf{x}$. □

Using Proposition 4, we can completely characterize the code graph of certain digraphs.

Example 4. *[45]*

- *If G is an acyclic digraph, then $X(G,s)$ is the complete graph.*
- *If G is an undirected clique K_n, then $X(G,s)$ is the Hamming graph $H(s,n)$, where two configurations are adjacent if and only if they are at Hamming distance 1.*
- *If G is a directed cycle $\overrightarrow{C_n}$, then $X(G,s)$ is characterized by the condition that two configurations are adjacent if and only if their Hamming distance is at most $n-1$.*

Proof. Let G be an acyclic digraph. We sort the vertices of G in topological order, such that $\Gamma(v_i) \subseteq \{v_1, ..., v_{i-1}\}$. Let $\mathbf{x}$, $\mathbf{x}'$ in A_s^n be two distinct configurations. We write l for $\min\{i : x_i \neq x'_i\}$. It is easy to see that $x_{\Gamma(v_l)} = x'_{\Gamma(v_l)}$ and $\mathbf{xx}' \in E_l(s)$.

If G is a clique K_n. It is clear that $E_i(s) = \{\mathbf{xx}' : x_i \neq x'_i, x_{V(G)-\{i\}} = x_{V(G)-\{i\}}\}$. Therefore, $\mathbf{x}$ is adjacent to $\mathbf{x}'$ if and only if the coordinates of $\mathbf{x}$ and $\mathbf{x}'$ are identical except one coordinate.

Let G be a directed cycle $\overrightarrow{C_n}$. The edge set of G is $\{(v_i, v_{i+1} \mod n) : 0 \leq i \leq n-1\}$. Suppose $\mathbf{x}$ and $\mathbf{x}'$ are distinct configurations and $\mathbf{xx}'$ is not an edge of $X(G,s)$, we show the i-th coordinate of $\mathbf{x}$ and $\mathbf{x}'$ are distinct for all $0 \leq i \leq n-1$. By assumption, we can find a coordinate i such that $x_i \neq x'_i$. This implies that $\mathbf{xx}' \notin E_i(s)$, hence $x_{i-1} \neq x'_{i-1}$. Applying this recursively, we obtain that $x_i \neq x'_i$ for all $0 \leq i \leq n-1$. The converse direction is clear. □

3.4 Lower bounds using the fractional clique cover

In this section we will describe a strategy specifically for undirected graphs. As shown in the previous section this can be used to provide a lower bound for the asymptotic guessing number. Christofides and Markström [28] conjectured that this bound always equals the asymptotic guessing number.

In Section 3.1 we saw that when an undirected graph is complete an optimal strategy is for each player to play assuming the sum of all the values is congruent to 0 mod s (where s is the alphabet size). We call this the *complete graph strategy.*

We can generalise this strategy to undirected graphs which are not complete. We simply decompose the undirected graph into vertex disjoint cliques and then let the players play the complete graph strategy on each of the cliques. If we are playing on an alphabet of size s and we decompose the graph into t disjoint cliques, then on each clique the probability of winning is s^{-1} and so the probability of winning the guessing game, which is equal to the probability of winning in each of the cliques, is s^{-t}. Clearly the probability of winning is higher if we choose to decompose the graph into as few cliques as possible. The smallest number of cliques that we can decompose a graph into is called the *minimum clique cover number* of G and we will represent it by $\kappa(G)$. In this notation we have

$$\operatorname{gn}(G) \geq \operatorname{gn}(G, s) \geq |V(G)| - \kappa(G).$$

It is worth mentioning that finding the minimum clique cover number of a graph is equivalent to finding the chromatic number of the graph's complement. As such it is difficult to determine this number in the sense that the computation of the chromatic number of a graph is an NP-complete problem [58].

We can improve this bound further by considering blowups of G. From Lemma 4 we know that $\operatorname{gn}(G, s^t) = \operatorname{gn}(G(t), s)/t$, hence by the clique cover strategy on $G(t)$ we get a lower bound of $|V(G)| - \kappa(G(t))/t$. The question is now to determine $\min_t \kappa(G(t))/t$. We do this by looking at the fractional clique cover of G.

Let $K(G)$ be the set of all cliques in G, and let $K(G, v)$ be the set of all cliques containing vertex v. A *fractional clique cover* of G is a weighting $w : K(G) \rightarrow [0, 1]$ such that for all $v \in V(G)$

$$\sum_{k \in K(G,v)} w(k) \geq 1.$$

The minimum value of $\sum_{k \in K(G)} w(k)$ over all choices of fractional clique covers w is known as *the fractional clique cover number* which we will denote by $\kappa_f(G)$. (Although we do not define it here, we point out that the fractional clique cover number of a graph is equal to the fractional chromatic number of its complement.)

For the purposes of guessing game strategies it will be more convenient to instead consider a special type of fractional clique cover called the regular fractional clique cover. A *regular fractional clique cover* of G is a weighting $w : K(G) \to [0,1]$ such that for all $v \in V(G)$

$$\sum_{k \in K(G,v)} w(k) = 1.$$

The minimum value of $\sum_{k \in K(G)} w(k)$ over all choices of regular fractional clique covers w can be shown to be equal to the fractional clique cover number $\kappa_f(G)$. To see this, observe firstly that since all regular fractional clique covers are fractional clique covers the minimum value of $\sum_{k \in K(G)} w(k)$ over all choices of regular fractional clique covers w is at least $\kappa_f(G)$. Finally, to show it is at most $\kappa_f(G)$ we simply observe that the optimal fractional clique cover can be made into a regular fractional cover by moving weights from larger cliques to smaller cliques. In particular, given a vertex v for which $\sum_{k \in K(G,v)} w(k) > 1$ we pick a clique $k_1 \in K(G,v)$ with $w(k_1) > 0$ and proceed as follows: We change the weight of k_1 from $w(k_1)$ to

$$w'(k_1) = \max \left\{ 0, 1 - \sum_{\substack{k \in K(G,v) \\ k \neq k_1}} w(k) \right\} < w(k_1)$$

We also change the weight of the clique $k_1' = k_1 \setminus \{v\}$ from $w(k_1')$ to $w'(k_1') = w(k_1') + w(k_1) - w'(k_1)$. We leave the weight of all other vertices the same. In this way, the total sum of weights over all cliques remains the same, the total sum of weights over all cliques containing a given vertex $v' \neq v$ also remains the same, but the total sum of weights over all cliques containing v is reduced. This process has to terminate because whenever we change the weight of k_1 it will either become equal to 0 or the total sum of weight of all cliques containing v will become equal to 1.

Clearly $\kappa_f(G)$ and an optimal regular fractional clique cover w can be determined by linear programming. Since all the coefficients of the constraints and objective function are integers, $w(k)$ will be rational for all $k \in K(G)$ as will $\kappa_f(G)$. If we let d be the common denominator of all the weights, then $dw(k)$ for $k \in K(G)$ describes a clique cover of $G(d)$. In particular it decomposes $G(d)$ into $d\kappa_f(G)$ cliques, proving a lower bound of

$$\operatorname{gn}(G) \geq |V(G)| - \kappa_f(G). \tag{3.4}$$

We claim that

$$\min_t \frac{\kappa(G(t))}{t} \geq \kappa_f(G)$$

and therefore we cannot hope to use regular fractional clique cover strategies to improve (3.4). To prove our claim we begin by observing that for all t we have

$\kappa(G(t)) \geq \kappa_f(G(t))$. This is immediate as a minimal clique cover is a special type of fractional clique cover, namely one where all weights are 0 or 1. Hence it is enough to show that

$$\kappa_f(G(t)) = t\kappa_f(G).$$

This can be proved simply from observing that an optimal weighting of $K(G(t))$ can always be transformed into another optimal weighting which is symmetric with respect to vertices in the same vertex class. This can be done just by moving the weights between cliques. Therefore determining $\kappa_f(G(t))$ is equivalent to determining $\kappa_f(G)$ but with the constraints $\sum_{k \in K(G,v)} w(k) = t$ rather than 1. The result $\kappa_f(G(t)) = t\kappa_f(G)$ is a simple consequence of this.

A useful bound on $\kappa_f(G)$ which we will make use of later is given by the following lemma.

Lemma 6. *For any undirected graph G*

$$\kappa_f(G) \geq \frac{|V(G)|}{\omega(G)},$$

where $\omega(G)$ is the number of vertices in a maximum clique in G.

Proof. Let w be an optimal regular fractional clique cover. Since

$$\sum_{k \in K(G,v)} w(k) = 1$$

holds for all $v \in V(G)$, summing both sides over v gives us,

$$\sum_{k \in K(G)} w(k)|V(k)| = |V(G)|,$$

where $|V(k)|$ is the number of vertices in clique k. The result trivially follows from observing

$$\sum_{k \in K(G)} w(k)|V(k)| \leq \sum_{k \in K(G)} w(k)\omega(G) = \kappa_f(G)\omega(G). \qquad \square$$

Christofides and Markström [28] proved the following:

Theorem 9. *If G is an undirected graph then*

$$\mathrm{gn}(G) \geq |V(G)| - \kappa_f(G).$$

In [28] it was proved that the above lower bound is actually an equality for various families of undirected graphs including perfect graphs, odd cycles and complements of odd cycles. This led Christofides and Markström [28] to conjecture that we always have equality.

Conjecture 1 (Fractional clique cover conjecture (FCCC)). If G is an undirected graph then

$$\mathrm{gn}(G) = |V(G)| - \kappa_f(G).$$

To prove or disprove such a claim we require a way of upper bounding $\mathrm{gn}(G)$. This is the purpose of the next section.

3.5 Upper bounds using entropy

Recall that it is sufficient to only consider pure strategies on guessing games. Hence given a strategy $\mathcal{F}$ on a guessing game (G, s) we can explicitly determine $\mathcal{S}(\mathcal{F})$ the set of all assignment tuples $(a_v)_{v \in V(G)}$ that result in the players winning given they are playing strategy $\mathcal{F}$. In this context the players' objective is to choose a strategy that maximizes $|\mathcal{S}(\mathcal{F})|$. We have

$$\begin{aligned} \mathrm{gn}(G, s) &= |V(G)| + \log_s \left(\max_{\mathcal{F}} \mathbf{P}[\mathrm{Win}(G, s, \mathcal{F})] \right) \\ &= |V(G)| + \max_{\mathcal{F}} \log_s \frac{|\mathcal{S}(\mathcal{F})|}{s^{|V(G)|}} \\ &= \max_{\mathcal{F}} \log_s |\mathcal{S}(\mathcal{F})|. \end{aligned}$$

Consider the probability space on the set of all assignment tuples $A_s^{|V(G)|}$ with the members in $\mathcal{S}(\mathcal{F})$ occurring with uniform probability and all other assignments occurring with 0 probability. For each $v \in V(G)$ we define the discrete random variable X_v on this probability space to be the value assigned to vertex v. The *s-entropy* of a discrete random variable X with outcomes $x_1, x_2, \ldots, x_n$ is defined as

$$H_s(X) = -\sum_{i=1}^{n} \mathbf{P}[X = x_i] \log_s \mathbf{P}[X = x_i],$$

where we take $0 \log_s 0$ to be 0 for consistency. Note that traditionally entropy is defined using base 2 logarithms, however it will be more convenient for us to work with base s logarithms. We will usually write $H(X)$ instead of $H_s(x)$. We will mention here all basic results concerning entropy that we are going to use. For more information, we refer the reader to [30, 95].

Given a set of random variables $Y_1, \ldots, Y_n$ with sets of outcomes $\mathrm{Im}(Y_1), \ldots, \mathrm{Im}(Y_n)$ respectively, recall from Section2.1 the *joint entropy* $H(Y_1, \ldots, Y_n)$ is defined as

$$-\sum_{y_1 \in \mathrm{Im}(Y_1)} \cdots \sum_{y_n \in \mathrm{Im}(Y_n)} \mathbf{P}[Y_1 = y_1, \ldots, Y_n = y_n] \log_s \mathbf{P}[Y_1 = y_1, \ldots, Y_n = y_n].$$

Given a set of random variables $Y = \{Y_1, \ldots, Y_n\}$ we will also use the notation $H(Y)$ to represent the joint entropy $H(Y_1, \ldots, Y_n)$. Furthermore for sets of random variables Y and Z we will use the notation $H(Y, Z)$ as shorthand for $H(Y \cup Z)$. For completeness we also define $H(\emptyset) = 0$.

Observe that under these definitions, the joint entropy of the set of variables $X_G = \{X_v : v \in V(G)\}$ is

$$\begin{aligned} H(X_G) &= -\sum_{(a_v) \in \mathcal{S}(\mathcal{F})} \frac{1}{|\mathcal{S}(\mathcal{F})|} \log_s \left(\frac{1}{|\mathcal{S}(\mathcal{F})|} \right) \\ &= \log_s |\mathcal{S}(\mathcal{F})| \end{aligned}$$

Therefore by upper bounding $H(X_G)$ for all choices of $\mathcal{F}$ we can upper bound $\mathrm{gn}(G, s)$.

We begin by stating some inequalities that most hold regardless of $\mathcal{F}$.

Theorem 10. *Given $X, Y, Z \subset X_G$,*

1. $H(X) \geq 0$.
2. $H(X) \leq |X|$.
3. *Shannon's information inequality:*

$$H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z) \geq 0.$$

4. *Suppose $A, B \subset V(G)$ with $\Gamma^-(u) \subset B$ for all $u \in A$. Let $X = \{X_v : v \in A\}$ and $Y = \{X_v : v \in B\}$. Then*

$$H(X, Y) = H(Y).$$

Proof.

- Property 1 follows immediately from the definition of entropy.

- Property 2 follows from first observing that $H(X) = \mathbf{E}[\log_s(1/\mathbf{P}[X])]$. Since the function $x \mapsto \log_s(x)$ is concave, by Jensen's inequality we get that

$$H(X) \leq \log_s \mathbf{E}[1/\mathbf{P}[X]] = \log_s |\operatorname{Im}(X)|$$

  where $\operatorname{Im}(X)$ is the set of outcomes for X. Since $\operatorname{Im}(X) = A_s^{|X|}$ we have the desired inequality $H(X) \leq |X|$.

- Property 3 again follows from Jensen's inequality. First we observe that

$$H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) = \mathbf{E}_{X,Y,Z}\left[-\log_s\left(\frac{\mathbf{P}[X,Z]\mathbf{P}[Y,Z]}{\mathbf{P}[X,Y,Z]\mathbf{P}[Z]}\right)\right].$$

  By an application of Jensen's inequality this is at least

$$-\log_s\left(\mathbf{E}_{X,Y,Z}\left[\frac{\mathbf{P}[X,Z]\mathbf{P}[Y,Z]}{\mathbf{P}[X,Y,Z]\mathbf{P}[Z]}\right]\right) = -\log_s\left(\sum_{X,Y,Z}\frac{\mathbf{P}[X,Z]\mathbf{P}[Y,Z]}{\mathbf{P}[Z]}\right) = 0.$$

- Property 4 is a simple consequence of the fact that the values assigned to the vertices in A are completely determined by the values assigned to the vertices in B. Since $\mathbf{P}[X,Y]$ is either 0 or $\mathbf{P}[Y]$, the result is trivially attained by considering the definition of $H(X,Y)$ and summing over the variables in X. □

From Theorem 10 we can form a linear program to upper bound $H(X_G)$. In particular the linear program consists of $2^{|V(G)|}$ variables corresponding to the values of $H(X)$ for each $X \subset X_G$. The variables are constrained by the linear inequalities given in Theorem 10 and the objective is to maximize the value of the variable corresponding to $H(X_G)$. We call the result of the optimization *the Shannon bound* of G and denote it by $\operatorname{Sh}(G)$.

Note that $\operatorname{Sh}(G)$ can be calculated without making any explicit use of $\mathcal{F}$ or s. Hence it is not only an upper bound on $\operatorname{gn}(G,s)$ but also on $\operatorname{gn}(G)$.

More recently information entropy inequalities that cannot be derived from linear combinations of Shannon's inequality (Property 3 in Theorem 10) have been discovered. The first such inequality was found by Zhang and Yeung [98]. The *Zhang-Yeung inequality* states that

$$\begin{aligned}&-2H(A) - 2H(B) - H(C) + 3H(A,B) + 3H(A,C) + H(A,D) +\\&\qquad 3H(B,C) + H(B,D) - H(C,D) - 4H(A,B,C) - H(A,B,D) \geq 0\end{aligned}$$

for sets of random variables A, B, C, D. By setting $A = X \cup Z$, $B = Z$, $C = Y \cup Z$, $D = Z$, the Zhang-Yeung inequality reduces to Shannon's inequality. By replacing the Shannon inequality constraints with those given by the Zhang-Yeung inequality we can potentially get a better upper bound from the linear program. However, we pay for this potentially better bound by a significant increase in the running time of the linear program. We will call the bound on $\mathrm{gn}(G)$ obtained by use of the Zhang-Yeung inequality *the Zhang-Yeung bound* and denote it by $\mathrm{ZY}(G)$.

In fact there are known to be infinite families of non-Shannon inequalities even on 4 variables. We cannot hope to add infinite constraints to the linear program so instead we will consider the 214 inequalities given by Dougherty, Freiling, and Zeger [38, Section VIII]. We will refer to the resulting bound as *the Dougherty-Freiling-Zeger bound* and denote it by $\mathrm{DFZ}(G)$. It is perhaps worth mentioning for those interested that the 214 Dougherty-Freiling-Zeger inequalities imply the Zhang-Yeung inequality (simply sum inequalities 56 and 90) and therefore they also imply Shannon's inequality.

The final bound we will consider is *the Ingleton bound* which we will denote by $\mathrm{Ingl}(G)$. This is obtained when we replace the Shannon inequality constraints with the *Ingleton inequality*

$$-H(A) - H(B) + H(A,B) + H(A,C) + H(A,D) + H(B,C) + \\ H(B,D) - H(C,D) - H(A,B,C) - H(A,B,D) \geq 0.$$

The Ingleton inequality provides the outer-bound of the inner-cone of linearly representable entropy vectors [27]. By setting $A = Z, C = Y$ and $B = D = X \cup Z$, the Ingleton inequality reduces to Shannon's inequality.

If each player's strategy can be expressed as a linear combination of the values it sees, then the Ingleton inequality will hold. Therefore the inequality holds for a strategy on (G, s^t) that can be represented as a linear strategy on $(G(t), s)$ (as described in the proof of Lemma 4). As such, the Ingleton bound gives us an upper bound when we restrict ourselves to strategies which are linear on the digits of the values. An important such strategy is the fractional clique cover strategy [28] which leads to the proof of Theorem 9.

In searching for a counterexample to Conjecture 1 we carried out an exhaustive search on all undirected graphs with at most 9 vertices. We compared the lower bound given by the fractional clique cover with the upper bound given by the Shannon bound and in all cases the two bounds matched. The bounds were calculated using floating point arithmetic and so we do not claim this search to be rigorous, however it leads us to make the following conjecture.

Conjecture 2. If G is an undirected graph then $\mathrm{gn}(G) = \mathrm{Sh}(G)$.

Chapter 4

Refuting the Fractional Clique Cover Conjecture

4.1 The first counter example to FCCC

In this section we provide counterexamples to Conjecture 1 is false. Counterexamples were found by searching through all undirected graphs on 10 vertices or less. For speed purposes, the search was done using floating point arithmetic and as such there may be counterexamples that were missed due to rounding errors. (Although this is highly unlikely, we do not claim that it is impossible. According to our knowledge, Christofides and Markström had already computed the guessing number of some undirected graphs of small order. This strengthens our belief that no such counter example can be found for graphs of order less than 10.) Despite this, we feel that it is still remarkable that of the roughly 12 million graphs that were checked we only found 2 graphs whose lower and upper bounds (given by the fractional clique cover, and Shannon bound respectively) did not match: the graph R given in Figure 4.1, and the graph R^{-} which is identical to R but with the undirected edge between vertices 9 and 10 removed.

The graph R is particularly extraordinary as we will see that with a few simple modifications we can create graphs which answer a few other open problems.

We begin our analysis of R and R^{-} by determining their fractional clique cover number.

Lemma 7. *We have* $\kappa_f(R) = \kappa_f(R^{-}) = 10/3$.

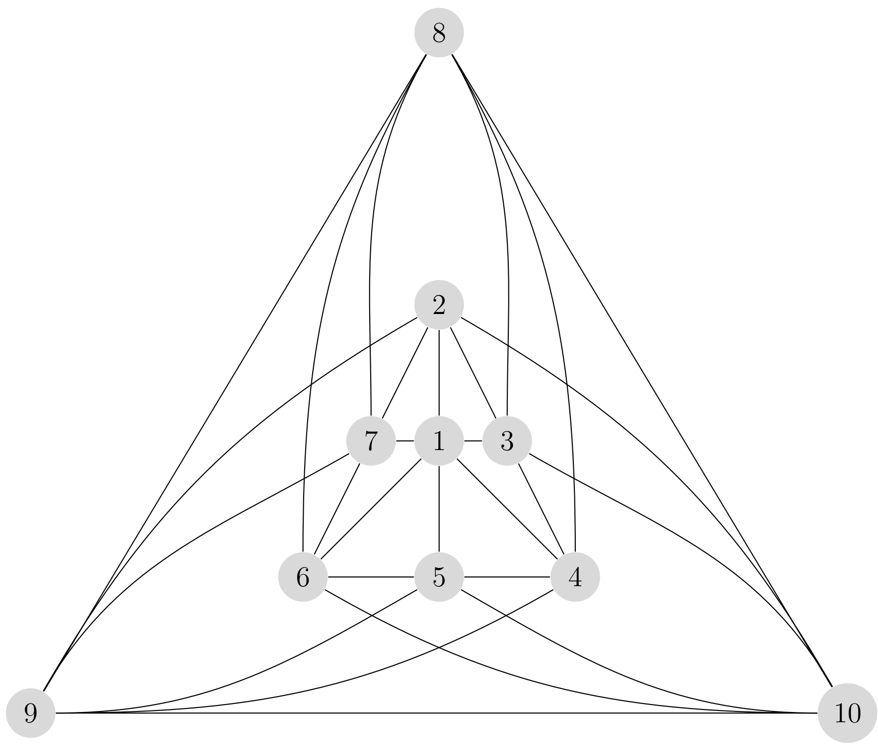


Figure 4.1: The undirected graph R.

Proof. By Lemma 6 we know that $\kappa_f(R)$ and $\kappa_f(R^-)$ are bounded below by 10/3. To show they can actually attain 10/3 we need to construct explicit regular fractional clique covers whose weights add up to 10/3.

For R^- we give a weight of 1/3 to the cliques $\{1,2,3\}$, $\{1,4,5\}$, $\{1,6,7\}$, $\{2,3,9\}$, $\{2,7,10\}$, $\{3,8,9\}$, $\{4,5,10\}$, $\{4,8,10\}$, $\{5,6,9\}$, $\{6,7,8\}$, and a weight of 0 to all other cliques. Note that this is also an optimal regular fractional clique cover for R. □

Theorem 11. *We have*

- $\text{Sh}(R^-) = 114/17 = 6.705\ldots$
- $\text{Ingl}(R^-) = 20/3 = 6.666\ldots$

From Lemma 7 and Theorem 11 we know that

$$20/3 \le \text{gn}(R^-) \le 114/17,$$

and although we could not determine the asymptotic guessing number exactly we will show that it does not equal the Shannon bound (see Section 5.2).

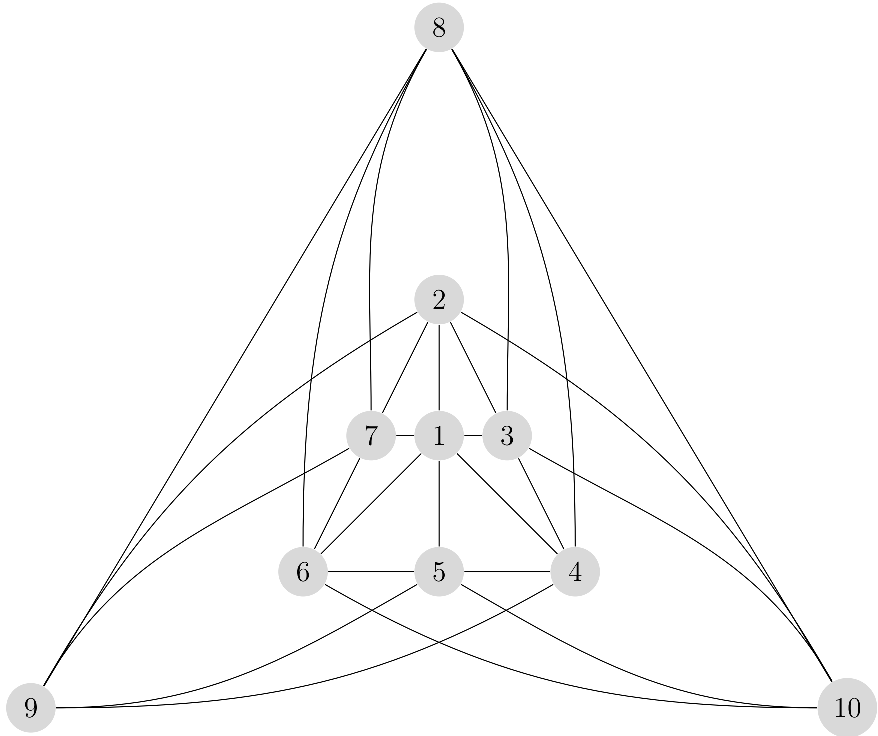


Figure 4.2: The undirected graph R^{-}.

Proof of Theorem 11. Calculating the upper bounds involves solving rather large linear programs. Hence the proofs are too long to reproduce here and it is infeasible for them to be checked by humans. Data files verifying our claims can be provided upon request. We stress that although the results were verified using a computer that no floating point data types were used during the verification. Consequently no rounding errors could occur in the calculations making the results completely rigorous. □

Although R is a counterexample to Conjecture 1 its optimal strategy is somewhat complicated. So instead we will disprove the conjecture by showing a related graph which we will call R_c is a counterexample. The undirected graph R_c is constructed from R by *cloning* 3 of its vertices. (Cloning 3 vertices is equivalent to creating a blowup of R with 2 vertices in 3 of the vertex classes and just 1 vertex in the other classes.) The vertices we clone are $8, 9, 10$, and we label the resulting new vertices $8', 9'$, and $10'$ respectively.

Theorem 12. *We have* $\text{gn}(R_c) = 9$ *while the fractional clique cover bound of* R_c *is* $26/3 < 9$*. In particular,* R_c *provides a counterexample to Conjecture 1.*

Proof. To prove that the fractional clique cover bound is $26/3$ it is enough to show that $\kappa_f(R_c) = 13 - 26/3 = 13/3$. Lemma 6 tells us $\kappa_f(R_c) \geq 13/3$. It is also easy to

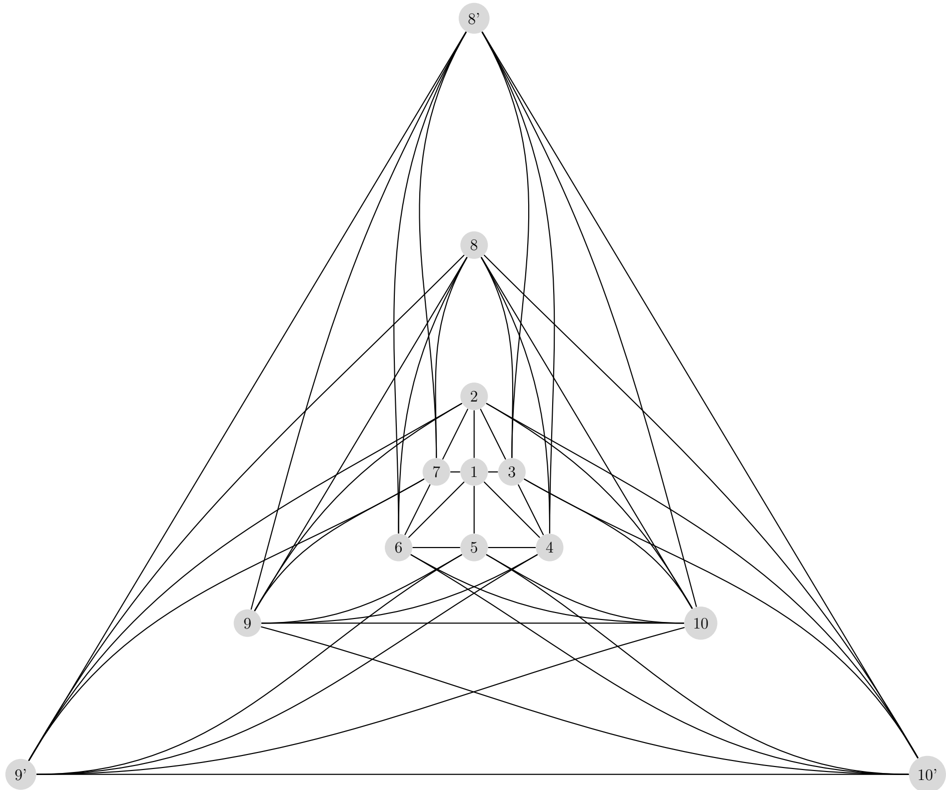


Figure 4.3: The undirected graph R_c.

show $\kappa_f(R_c) \leq 13/3$ as it trivially follows from extending the regular fractional clique cover given in the proof of Lemma 7 by giving a weight of 1 to the clique $\{8', 9', 10'\}$.

The Shannon bound of R_c is 9 proving $\text{gn}(R_c) \leq 9$. We do not provide the details of the Shannon bound proof as it is too long to present here, however data files containing the proof are available upon request.

All that remains is to prove $\text{gn}(R_c) \geq 9$. Even though this proof was discovered partly using a computer it can be easily verified by humans. In particular, the main conclusion of this theorem, that R_c is a counterexample to Conjecture 1, can be verified without the need of any computing power.

Recall that in Section 3.2 we showed that the asymptotic guessing number can be lower bounded by considering any strategy on any alphabet size. We will take our alphabet size s to be 3. Our strategy involves all players agreeing to play assuming the following four conditions hold on the assigned values

$$a_1 + a_2 + 2a_3 + a_4 + 2a_5 + a_6 + 2a_7 \equiv 0 \bmod 3, \tag{4.1}$$

$$a_2 + a_5 + a_8 + a_{8'} + a_9 + a_{10'} \equiv 0 \bmod 3, \tag{4.2}$$

$$a_3 + a_6 + a_8 + a_{9'} + a_{10} + a_{10'} \equiv 0 \bmod 3, \tag{4.3}$$

$$a_4 + a_7 + a_{8'} + a_9 + a_{9'} + a_{10} \equiv 0 \bmod 3. \tag{4.4}$$

Note that the terms in (4.1) consist of a_1, and values which player 1 can see. Hence (4.1) naturally gives us a strategy for player 1, i.e. that player 1 should guess $-a_2 - 2a_3 - a_4 - 2a_5 - a_6 - 2a_7 \mod 3$. Similarly strategies for players $8, 8', 9, 9', 10$, and $10'$ can be achieved by rearranging conditions (4.3), (4.4), (4.2), (4.3), (4.4) and (4.2) respectively. A strategy for player 2 can be obtained by taking a linear combination of the conditions. In particular if we sum (4.3), (4.4), twice (4.1), and twice (4.2) we get

$$2a_1 + a_2 + 2a_3 + 2a_7 + 2a_{9'} + 2a_{10} \equiv 0 \bmod 3,$$

which consists of a_2 and values which player 2 can see, allowing us to construct a strategy for player 2. We leave it to the reader to verify that by taking the following linear combinations we obtain strategies for players $3, 4, 5, 6$, and 7:

- For player 3, we sum (4.1),(4.2),(4.4) and twice (4.3).
- For player 4, we sum (4.2),(4.3), twice (4.1) and twice (4.4).
- For player 5, we sum (4.1),(4.3),(4.4) and twice (4.2).
- For player 6, we sum (4.2),(4.4), twice (4.1) and twice (4.3).
- For player 7, we sum (4.1),(4.2),(4.3) and twice (4.4).

The probability that all players guess correctly under this strategy is 3^{-4}, i.e. the probability that (4.1), (4.2), (4.3), (4.4) all hold. (It is not difficult to check that the conditions are linearly independent.) Consequently

$$\text{gn}(R_c) \geq |V(R_c)| + \log_3 \mathbf{P}[\text{Win}(R_c, 3, \mathcal{F})] = 9$$

as desired.

□

For completeness we give the asymptotic guessing number of R and note that it does not match the fractional clique cover bound of 20/3 as claimed.

Theorem 13. *We have* $\text{gn}(R) = 27/4$.

Proof. The Shannon bound of R is 27/4 (data files can be provided upon request).

To show $\text{gn}(R) \geq 27/4$ we will show $\text{gn}(R, 81) \geq 27/4$. By Lemma 4 this can be achieved if we can construct a strategy on the guessing game $(R(4), 3)$ which has a probability of winning 3^{-13} (which implies $\text{gn}(R(4), 3) \geq 27$). To describe such a strategy let us label the vertices of $R(4)$ such that the four vertices that are constructed from blowing up $v \in V(R)$ are labelled v_a, v_b, v_c, and v_d. Under this labelling our strategy for $R(4)$ is to have the cliques $\{1_a, 2_a, 3_a\}$, $\{1_b, 4_a, 5_a\}$, $\{1_c, 6_a, 7_a\}$, $\{2_b, 3_b, 9_a\}$, $\{2_c, 3_c, 9_b\}$, $\{4_b, 5_b, 10_a\}$, $\{4_c, 5_c, 10_b\}$, $\{6_b, 7_b, 8_a\}$ and $\{6_c, 7_c, 8_b\}$ play the complete graph strategy, and the remaining 13 vertices, which form a copy of R_c, to play the strategy for R_c as described in the proof of Theorem. 12. □

Now that we have shown that Conjecture 1 is not true, we turn our attention to other open questions. Due to the limited tools and methods currently available, there are many seemingly trivial problems on guessing games which still remain unsolved. One such problem is the following.

Problem 3. Does there exist an undirected graph whose asymptotic guessing number increases when a single directed edge is added?

Adding a directed edge gives one of the players more information, which cannot lower the probability that the players win. However, surprisingly it seems extremely difficult to make use of the extra directed edge to increase the asymptotic guessing number. An exhaustive (but not completely rigorous) search on undirected graphs with 9 vertices or less did not yield any examples.

As such, we significantly weaken the requirements in Problem 3 by introducing the concept of a Superman vertex. We define a Superman vertex to be one that all other vertices can see. I.e., given a digraph G, we call vertex $u \in V(G)$ a *Superman vertex* if $uv \in E(G)$ for all $v \in V(G) \setminus \{u\}$. We can similarly define a Luthor vertex as one which sees all other vertices. To be precise u is a *Luthor vertex* if $vu \in E(G)$ for all $v \in V(G) \setminus \{u\}$.

Problem 4. Does there exist an undirected graph whose asymptotic guessing number increases when directed edges are added to change one of the vertices into a Superman vertex (or a Luthor vertex)?

To change one of the vertices into a Superman or Luthor vertex will often involve adding multiple directed edges, meaning the players will have a lot more information at their disposal when making their guesses. We again searched all undirected graphs on 9 vertices or less and remarkably still could not find any examples.

With the discovery of the graph R and in particular the graph R_c we can show the answer is yes to Problem 3 and consequently Problem 4. We define the undirected graph R_c^- to be the same as the graph R_c but with the undirected edge between vertices 3 and 8 removed. We also define the directed graph R_c^+ to be the same as R_c^- but with the addition of a single directed edge going from vertex 3 to vertex 8.

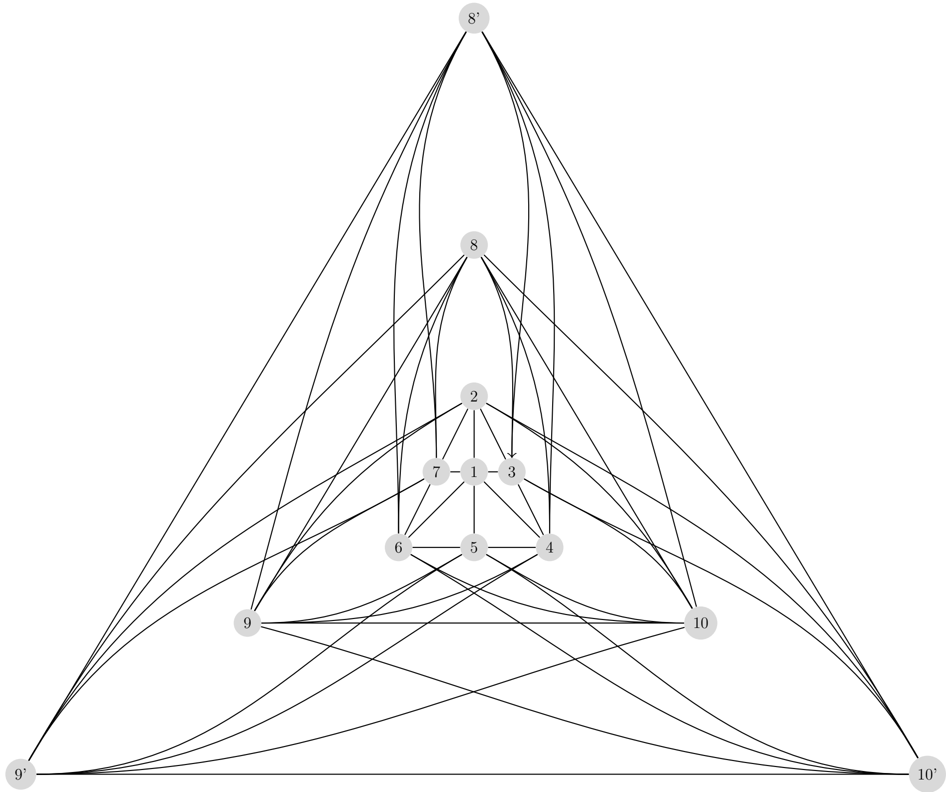


Figure 4.4: The graph R_c^+.

Theorem 14. *We have* $\mathrm{gn}(R_c^+) = 9$.

Proof. The Shannon bound for R_c^+ is 9 (data files can be provided upon request).

We will prove $\mathrm{gn}(R_c^+) \geq 9$ by observing that the strategy for $(R_c, 3)$ (see the proof of Theorem 12) is a valid strategy for $(R_c^+, 3)$. With the exception of player 3 all players in $(R_c^+, 3)$ have access to the same information they did in $(R_c, 3)$. Player 3 however, now no longer has access to a_8. By studying the strategy player 3 uses in $(R_c, 3)$ we will see that this is of no consequence. Summing conditions (4.1), (4.2), (4.4), and twice (4.3), gives

$$a_1 + 2a_2 + a_3 + 2a_4 + 2a_{8'} + 2a_9 \equiv 0 \bmod 3,$$

hence player 3 guesses $-a_1 - 2a_2 - 2a_4 - 2a_{8'} - 2a_9 \bmod 3$ in $(R_c, 3)$. Since player 3 makes no use of a_8 this validates our claims. $\square$

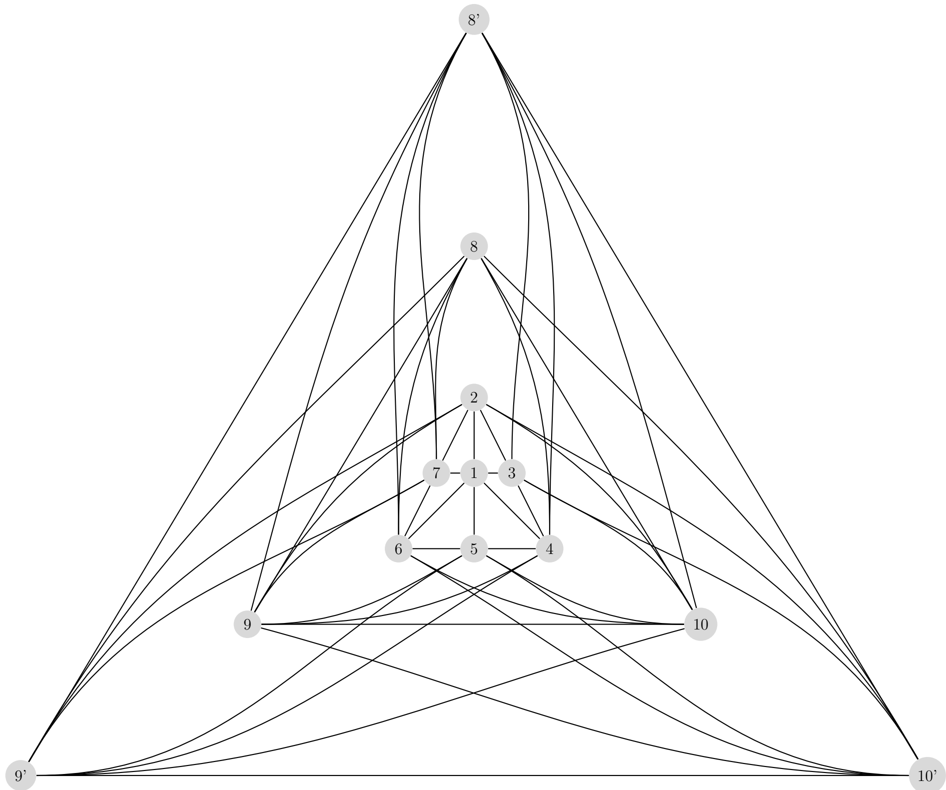


Figure 4.5: The digraph R_c^-.

Theorem 15. *We have* $\mathrm{gn}(R_c^-) = 53/6$.

Proof. The Shannon bounds for R_c^- is 53/6 (data files can be provided upon request).

We complete our proof by showing $\mathrm{gn}(R_c^-) \geq 53/6$. We know $\mathrm{gn}(R_c^-) \geq \mathrm{gn}(R_c^-, 3^6) = \mathrm{gn}(R_c^-(6), 3)/6$ so it is enough to show $\mathrm{gn}(R_c^-(6), 3) \geq 53$. Since $R_c^-(6)$ had 78 vertices we can do this by finding a strategy on $(R_c^-(6), 3)$ that wins with a probability of 3^{-25}. To this end, let us label the vertices of $R_c^-(6)$ such that the six vertices that are constructed from blowing up $v \in V(R_c^-)$ are labelled v_a, v_b, v_c, v_d, v_e, and v_f. Under this labelling, our strategy for $R_c^-(6)$ is to play the complete graph strategy on the cliques

$$\{1_a, 2_a, 3_a\}, \quad \{1_b, 2_b, 7_a\}, \quad \{1_c, 3_b, 4_a\}, \quad \{2_c, 3_c, 9'_a\}, \quad \{4_b, 5_a, 10'_a\},$$
$$\{4_c, 5_b, 10'_b\}, \quad \{5_c, 6_a, 9'_b\}, \quad \{6_b, 7_b, 8_a\}, \quad \{6_c, 7_c, 8_b\}, \quad \{8_c, 9'_c, 10'_c\},$$
$$\{8_d, 9'_d, 10'_d\}, \quad \{8_e, 9'_e, 10'_e\}, \quad \{8_f, 9'_f, 10'_f\},$$

and to play the R_c strategy on the vertices

$$\{1_d, 2_d, 3_d, 4_d, 5_d, 6_d, 7_d, 8'_a, 8'_b, 9_a, 9_b, 10_a, 10_b\},$$
$$\{1_e, 2_e, 3_e, 4_e, 5_e, 6_e, 7_e, 8'_c, 8'_d, 9_c, 9_d, 10_c, 10_d\},$$
$$\{1_f, 2_f, 3_f, 4_f, 5_f, 6_f, 7_f, 8'_e, 8'_f, 9_e, 9_f, 10_e, 10_f\}.$$

The probability of winning in each of these 13 cliques is 3^{-1} while the probability of winning in each of the three copies of R_c is 3^{-4}. So the overall probability of winning is indeed 3^{-25}, therefore completing the proof. □

4.2 Speeding up the computer search

In this section we mention a few of the simple tricks we used in order to speed up the computer search which allowed us to search through all the 10 vertex graphs and find the graph R. We hope that this may be of use to others continuing this research.

The majority of time spent during the searches was spent determining the Shannon bound by solving a large linear program. By reducing the number of constraints that we add to the linear program we can speed up the optimisation. Given a graph on n vertices a naive formation of the linear program would result in considering all 2^{3n} Shannon inequalities of the form

$$H(A, C) + H(B, C) - H(A, B, C) - H(C) \geq 0 \text{ for } A, B, C \subset X_G.$$

However most of these do not need to be added to the linear program. In fact it is sufficient to just include the inequalities given by the following lemma.

Lemma 8. *Given a set of discrete random variables X_G, the set of Shannon inequalities*

$$H(A, C) + H(B, C) - H(A, B, C) - H(C) \geq 0 \textit{ for } A, B, C \subset X_G,$$

is equivalent to the set of inequalities given by

1. $H(Y) \leq H(X_G)$ *for* $Y \subset X_G$ *with* $|Y| = |X_G| - 1$.
2. $H(Y) + H(Z) - H(Y \cup Z) - H(Y \cap Z) \geq 0$ *for* $Y, Z \subset X_G$ *with* $|Y| = |Z| = |Y \cap Z| + 1$.

Observe that for a graph on n vertices there are n inequalities of type (1) and $n(n-1)2^{n-3}$ inequalities of type (2). (Counting the inequalities of type (2) is equivalent to counting the number of squares in the hypercube poset formed from looking at the subsets of X_G.) Overall, this is about the cube root of the initial number of inequalities.

Proof of Lemma 8. Setting $A = X_G$, $B = X_G$, and $C = Y$, shows that the Shannon inequalities imply the set of inequalities described by (1). Setting $A = Y$, $B = Z$, and $C = Y \cap Z$, shows that the Shannon inequalities imply (2).

To show (1) and (2) imply the Shannon inequalities we will first generalise (1) and (2).

We will begin by showing that (2) implies

$$H(Y) + H(Z) - H(Y \cup Z) - H(Y \cap Z) \geq 0$$

for any $Y, Z \subset X_G$. Let $Y \setminus (Y \cap Z) = \{Y_1, Y_2, \ldots, Y_n\}$ and $Z \setminus (Y \cap Z) = \{Z_1, Z_2, \ldots, Z_m\}$, where $Y_1, \ldots, Y_n$ and $Z_1, \ldots, Z_m$ are single discrete random variables. Define Y'_i to be $\{Y_1, \ldots, Y_i\}$ for $1 \leq i \leq n$ and $Y'_0 = \emptyset$. We define Z'_i similarly. Finally let $X_{i,j} = (Y \cap Z) \cup Y'_i \cup Z'_j$, and note that $X_{0,0} = Y \cap Z$, $X_{n,0} = Y$, $X_{0,m} = Z$, and $X_{n,m} = Y \cup Z$. By (2) we have

$$0 \leq \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} [H(X_{i+1,j}) + H(X_{i,j+1}) - H(X_{i+1,j+1}) - H(X_{i,j})].$$

Here, the right hand side is telescopic and simplifies to the desired expression

$$H(Y) + H(Z) - H(Y \cup Z) - H(Y \cap Z).$$

Next we will generalise (1) to show that for any $Y \subset Z \subset X_G$ with $|Y| = |Z| - 1$ we have $H(Y) \leq H(Z)$. Let us define $\overline{Z}$ to be $X_G \setminus Z$. Then, by the generalised version of (2) we know that

$$H(Z) + H(Y \cup \overline{Z}) - H(Z \cup (Y \cup \overline{Z})) - H(Z \cap (Y \cup \overline{Z})) \geq 0$$

which simplifies to

$$H(Z) + H(Y \cup \overline{Z}) - H(X_G) - H(Y) \geq 0. \tag{4.5}$$

Observe that $|Y \cup \overline{Z}| = |X_G| - 1$, so (1) tells us that $H(X_G) - H(Y \cup \overline{Z}) \geq 0$ which when added to (4.5) gives the inequality $H(Z) - H(Y) \geq 0$ as required.

We can now further generalise (1) to show that for any $Y \subset Z \subset X_G$ we have that $H(Y) \leq H(Z)$. To do this, let $Z \setminus Y = \{Z_1, Z_2, \ldots, Z_n\}$, where $Z_1, \ldots, Z_n$ are single discrete random variables. Then, by repeated applications of our generalisation of (1) we have

$$H(Y) \leq H(Y, Z_1) \leq H(Y, Z_1, Z_2) \leq \cdots \leq H(Y, Z_1, \ldots, Z_n) = H(Z).$$

It is now a trivial matter to show that (1) and (2) imply Shannon's inequality. Simply set $Y = A \cup C$ and $Z = B \cup C$ in the generalised version of (2) to get

$$H(A, C) + H(B, C) - H(A, B, C) - H(A \cap B, C) \geq 0$$

and since $H(A \cap B, C) \geq H(C)$ by the improved version of (1), the result follows. □

It is also worth mentioning that $H(\emptyset) = 0$ together with the Shannon inequalities imply $H(Y, Z) \leq H(Y) + H(Z)$ for disjoint Y, Z. Hence, the constraints $H(X) \leq |X|$ for all X in the Shannon bound linear program are not all necessary and can be reduced to $H(X) \leq |X|$ for $|X| = 0$, or 1.

When determining each graph's asymptotic guessing number, the natural approach is to calculate the lower bound using the fractional clique cover number, then calculate the Shannon bound and check if they match. However the linear program that gives us the fractional clique cover number also gives us a regular fractional clique cover from which an explicit strategy can be constructed. It is easy to convert this strategy into a feasible point of the Shannon bound linear program. Hence we can save a significant amount of time by simply checking if this feasible point is optimal, rather than by calculating the Shannon bound from scratch. Note that we check for optimality by solving the same Shannon bound linear program with the modification that we remove those constraints for which equality is not achieved by the feasible point.

The modified Shannon bound linear program is still the most time consuming process in the search, so ideally we would like to avoid it when possible. Christofides and Markström [28] show that for an undirected graph G

$$\text{gn}(G) \leq |V(G)| - \alpha(G),$$

where $\alpha(G)$ is the number of vertices in the maximum independent set. This can be interpreted as a simple consequence of the fact that removing players increases the probability the remaining players will win. (If the probability of winning decreased, the players could just create fictitious replacement players before the game started.) As such we present a simple generalization of this result.

Lemma 9. *Given a digraph G and an induced subgraph G',*

$$\max_{\mathcal{F}} \mathbf{P}[\mathrm{Win}(G, s, \mathcal{F})] \leq \max_{\mathcal{F}} \mathbf{P}[\mathrm{Win}(G', s, \mathcal{F})]$$

or equivalently $\mathrm{gn}(G, s) \leq |V(G)| - |V(G')| + \mathrm{gn}(G', s)$. *Hence*

$$\mathrm{gn}(G) \leq |V(G)| - |V(G')| + \mathrm{gn}(G').$$

This lemma is based on a simple fact that we can convert all guessing strategies played on G into strategies played on induced subgraph G' by assuming the assigned values for vertices of G living outside of G' are 0. The conclusion follows. Note that the result $\mathrm{gn}(G) \leq |V(G)| - \alpha(G)$ is a simple corollary of this result as an independent set has a guessing number of 0.

Given a graph G, if we can find a subgraph such that the upper bound given in Lemma 9 matches the fractional clique cover bound, then we have determined the asymptotic guessing number, and can avoid an expensive Shannon bound calculation. This approach is particularly fast when doing an exhaustive search as all the smaller graphs will have had their asymptotic guessing numbers already determined.

One issue with this method is that if we are looking for a counterexample to the Shannon bound being sharp, there is a possibility that we may miss them because we avoided calculating the Shannon bound for every graph. Consequently to alleviate our fear we need the following result.

Lemma 10. *Given a digraph G and an induced subgraph G', we have*

$$\mathrm{Sh}(G) \leq |V(G)| - |V(G')| + \mathrm{Sh}(G').$$

Proof. It is sufficient to prove the result only for induced subgraphs G' for which $|V(G)| - |V(G')| = 1$, as the result then follows by induction on $|V(G)| - |V(G')|$. Let $u \in V(G)$ be the vertex that is removed from G to produce G'.

The Shannon bound for G' comes from solving a linear program, and as such the solution to the dual program naturally gives us a proof that $H(X_{G'}) \leq \mathrm{Sh}(G')$. In particular, this proof consists of summing appropriate linear combinations of the constraints. Suppose that in each such contraint we replace $H(X)$ with $H(X, X_u) - H(X_u)$ for every $X \subset X_{G'}$. This effectively would replace constraints from the linear program for G' with inequalities which are implied from the linear program for G. For example, $H(X) \geq 0$ for G', would become $H(X, X_u) - H(X_u) \geq 0$ for G (which is true by Shannon's inequality). As another example, $H(X) \leq |X|$ becomes $H(X, X_u) -$

$H(X_u) \leq |X|$ (which is true as $H(X, X_u) \leq H(X) + H(X_u) \leq |X| + H(X_u)$). This shows that all constraints in Theorem 10 of types (1) and (2) can be replaces as claimed. The same happens for constraints of types (3) and (4). Consequently, under this transformation, the proof that $H(X_{G'}) \leq \mathrm{Sh}(G')$ becomes a proof that $H(X_G) - H(X_u) \leq \mathrm{Sh}(G')$. Since $H(X_u) \leq 1$ the result immediately follows. □

We have seen that by removing vertices from a graph G we make the game easier allowing us to upper bound $\mathrm{gn}(G)$. Another way we can make the game easier is by adding extra edges to G. Consequently we can avoid the Shannon bound calculation by also using the asymptotic guessing number of supergraphs of G which have the same number of vertices as G.

We end this section by considering the problem of how to calculate the non-Shannon bounds, i.e. the Zhang-Yeung bound, the Dougherty-Freiling-Zeger bound, and the Ingleton bound. They all involve inequalities on 4 variables and consequently a naive approach is to add at least 2^{4n} inequalities to the linear program, where n is the order of the graph. Unfortunately such a linear program is far too large to be computationally feasible. Our approach is given by the following algorithm:

- Let $\mathcal{C}$ be the set of Shannon bound constraints.
- Solve the linear program which consists only of constraints $\mathcal{C}$.
- Check if the solution satisfies all required 4 variable information inequalities (e.g. the Zhang-Yeung inequalities if we are calculating the Zhang-Yeung bound).
  - If all the inequalities are satisfied then terminate, returning the objective value.
  - If one of the inequalities is not satisfied add this constraint to $\mathcal{C}$ and go back to 4.2.

We note that due to the large number of inequalities, Step 4.2 can take a while. So it is advisable to add some extra constraints to the linear program to limit the search to a solution which is symmetric under the automorphisms of the graph (there always exists such a solution due to the linearity of the problem). This extra symmetry can be used to avoid checking a significant proportion of the inequalities in Step 4.2.

4.3 Triangle-free graphs with large guessing number

In the previous section, we have seen that the fractional clique cover strategy conjecture does not hold for our graph R and its variation. However, we have noticed that the graph R contains at least one triangle, i.e. clique of size 3. A natural question is that if we forbid the appearance of triangle in undirected graph, then is the fractional clique cover the best guessing strategy for our undirected graphs? In other words,

Conjecture 5. If G is an undirected triangle-free graph then

$$\mathrm{gn}(G) = |V(G)| - \kappa_f(G).$$

Recall Lemma 6 that: for any undirected graph G

$$\kappa_f(G) \geq \frac{|V(G)|}{\omega(G)},$$

where $\omega(G)$ is the number of vertices in a maximum clique in G.

Therefore, for any triangle-free graph G, we have a simple bound for $\kappa_f(G)$:

Corollary 3. *For triangle-free graph G, the $\kappa_f(G) \geq |V(G)|/2$.*

We will show in this section that there are triangle-free graphs for which the asymptotic guessing $\mathrm{gn}(G)$ of G is strictly greater than $|V(G)|/2$. Combining this with Corollary 3, we will prove that the answer to the Conjecture 5 is negative. Before illustrating our results, we need to introduce the following definition:

Definition 16. *Given graph $G = (V, E)$ of order n, we say a square matrix M of order n with entries selected from a finite field $\mathbb{F}_q$ of q elements with rows and columns indexed by vertices $i \in V(G)$ represents G over $\mathbb{F}_q$ if the diagonal entries of M are non-zero and the non-diagonal entries m_{ij} are 0 whenever $ij \notin E(G)$.*

Let M be a representing matrix of G over $\mathbb{F}_q$. We can form a guessing strategy for (G, q) by asking each player i to adapt an assumption that the assigned values of itself and every player in its neighbourhood are taken from $\mathbb{F}_q$ and satisfy a linear equation

$$m_{ii}x_i + \sum_{j \in \Gamma(i)} m_{ij}x_j = 0$$

where x_i, x_js are assigned values of players i, js, and coefficients m_{ii} and m_{ij} are the (i, i)-th and (i, j)-th entries of M. Then the value of x_i produced by this strategy is

$$x_i = m_{ii}^{-1} \sum_{j \in \Gamma(i)} m_{ij}x_j,$$

which is well-defined since $m_i \neq 0$ by assumption.

The guessing game (G, q) is won by adopting strategy M if the assigned values $X = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix}^\top$ give a solution of a system of linear equations $MX = 0$ defined over $\mathbb{F}_q$.

$$\text{gn}(G, q, M) = \log_q |\{X \in \mathbb{F}_q^n \,|MX = 0\}| = n - \text{rk}_{\mathbb{F}_q}(M) \tag{4.6}$$

We note that $MX = 0$ always has a trivial solution $X = \mathbf{0}$ hence $\log_q |\{X \in \mathbb{F}_q^n \,|MX = 0\}|$ is well defined.

The value $n - \text{rk}_{\mathbb{F}_q}(M)$ is a valid lower-bound of $\text{gn}(G)$, i.e.

$$\text{gn}(G) \geq \text{gn}(G, q, M) = n - \text{rk}_{\mathbb{F}_q}(M).$$

It is clear that we can disprove Conjecture 5 by constructing a triangle-free graph that has a representation matrix M with $\text{rk}_{\mathbb{F}_q}(M) < |V(G)|/2$ over some finite field $\mathbb{F}_q$.

Definition 17. *A* Steiner system $S(t, k, n)$ *is a family of k-element subsets of* $\{1, 2, \ldots, n\} =: [n]$ *with the property that each t-element subset of $[n]$ is contained in exactly one element of $S(t, k, n)$. Elements of $S(t, k, n)$ are called blocks, and elements of $[n]$ are referred to as points.*

For more information about Steiner systems, and the particular system used here, we refer to [24, Chapter 1].

The following proposition plays a crucial role in our construction:

Proposition 5. *The Steiner system $S(3, 6, 22)$ has the following properties:*

(a) $S(3, 6, 22)$ contains 77 blocks.

(b) Any two blocks in $S(3, 6, 22)$ intersect in zero or two points.

(c) No three blocks in $S(3, 6, 22)$ are disjoint.

(d) Each point is contained in exactly 21 blocks.

Proof. (a) We simply count the number of blocks containing a fixed set of points. Given two points i, j in $[n]$, there are 20 choices of the third point $k \in [n] \backslash \{i, j\}$ to form a group of 3 points. By definition, any 3 points of $[n]$ belongs to exactly one block, hence there are exactly 20 blocks containing both two fixed points i and j.

Let B and C be two blocks containing both i and j. We have $B \cap C = \{i, j\}$ and there are 4 points in B other than i and j, so there are $20/4 = 5$ blocks that contain both i and j.

Now we fix one point i in $[n]$. There are 21 pairs of $[n]$ containing i and if x is a block that contains i then it also contains 5 pairs of $[n]$ containing i, hence each point i of $[n]$ belongs to $21 \cdot 5/5 = 21$ blocks.

We repeat our argument for zero point of $[n]$ and we derive that there are $22{\cdot}21/6 = 77$ blocks of $S(3, 6, 22)$.

(b) We see in the first part that each point in $S(3, 6, 22)$ belongs to 21 different blocks. If we fix a point p, then there are 21 points $q \neq p$, and each of these points belongs to 5 blocks that containing p. The system of 21 blocks on 21 points satisfies the following properties:

(i) For every two distinct points q, $l \neq p$, there is exactly one block that contains both points (by definition of $S(3, 6, 22)$).

(ii) Let B be a block in the set of 21 blocks containing p. For each point $q \neq p$ in B, there are exactly 5 blocks contains q including B (from (i)). Moreover, it is clear that for any two distinct points q, l which are different from p, the set of blocks containing q and the set of blocks containing l share B as their unique common element. Since B is arbitrarily, this shows that for any two blocks B and C in the set of 21 blocks containing a fixed point p, B and C intersect at exactly one point beside p.

(iii) Let B be a block that contains the fixed point p. We choose other 3 points q, k, l in B and a point h that does not belong to B. The set of four points $\{q, k, l, h\}$ obviously cannot be contained in one single block of the 21 blocks having p as their element.

Hence these 21 blocks on 21 points form a projective plane where each block is a line in this plane. A corollary is that any two blocks that contain a fixed point p must also contain another point q. This proves that any two blocks in $S(3, 6, 22)$ either intersect in zero or two points.

(c) We fix a block $B \in S(3, 6, 22)$. There are 16 points that are not in B. Moreover, for every two distinct pairs of points of B, the set of blocks containing one pair and the set of blocks containing the other pair share B as their unique common element. Therefore, there are exactly 60 blocks having non-empty intersection with B. This leaves 16 blocks that have empty intersection with B. From (b) we know that any two blocks must intersect in zero or two points, this makes the 16 points and 16 blocks a symmetric balanced incomplete block design (BIBD) $(16, 6, 2)$. It follows from the property of symmetric BIBD that any two blocks intersect in 2 points.

(d) This is already proved in part (a). □

Theorem 16. *There exists an undirected triangle-free graph G on 100 vertices with $\mathrm{gn}(G) \geq 77$.*

Proof. We define the vertex set of the graph G to be 22 points plus 77 blocks of the Steiner system $S(3, 6, 22)$ plus an extra point $\{\infty\}$. There is an edge between two vertices u and v if one of the following conditions is satisfied:

- u is $\{\infty\}$, and v is a point.
- u is a point and v is a block which contains u as an element.
- u and v are blocks of $S(3, 6, 22)$ and $u \cap v = \emptyset$.

According to the previous proposition, the graph obtained form our construction is triangle-free. It remains to show that there is a matrix representing G with rank less than 50 over some finite field $\mathbb{F}_q$.

The chosen matrix is $A+I$ where A is the adjacency matrix of G and I is the identity matrix of order 100. The rank of the matrix $A + I$ is 23 over the finite field $\mathbb{F}_3$ (see the next Proposition).

In this graph, the size of the maximal independent set is 22 (and the independent sets of size 22 are the vertex neighbourhoods), hence the guessing number of this graph is at most 78. □

The constructed graph is in fact the Higman–Sims graph [52], which is a strongly regular triangle-free graph with parameters $(100, 22, 6)$. The Higman-Sims graph was first introduced by Dale Mesner in his 1956 PhD thesis [77]; see [61] for a historical account.

Proposition 6. *If A is the adjacency matrix of the Higman–Sims graph, then the rank of $A + I$ over $\mathbb{F}_3$ is 23.*

Proof. Let r_v be the row of $B = A+I$ corresponding to vertex v. We write the vertex set as $\{\infty\} \cup X \cup Y$, where X and Y are the neighbours and non-neighbours of ∞. Consider the 22 vectors r_x for $x \in X$. Since the graph is triangle-free, the restriction of r_x to the coordinates in X has a one in position x and zeros elsewhere; so these 22 vectors are linearly independent. Take the 23rd vector to be the all-1 vector j. Note that j is not in the span of the first 22. For if it were, it would have to be their sum (looking at the restriction to X. But the sum of the r_x has coordinate $22 \equiv 1 \mod 3$ at ∞, 1 at each point of X, and $6 \equiv 0 \mod 3$ at each point of Y; that is, it is r_∞. So our 23 vectors are linearly independent. Also, they are all contained in the row space of B. (This is clear for the r_x; also the sum of all the vectors r_v is $2j$, since all column sums of B are $23 \equiv 2 \mod 3$, so j is also in the row space.

We claim that they span the row space. It is clear that their span contains all r_x for $x \in X$, and we just showed that it contains r_∞. Take a vertex $y \in Y$. Consider the sum of the vectors r_x for the 16 vertices $x \in X$ which are not joined to y. This has coordinate $16 \equiv 1 \mod 3$ at ∞, 0 at points of X joined to y, and 1 at points of X not joined to y. The coordinate at y is zero. If y' is joined to y, then the six neighbours of y' in X are a subset of the 16 points not joined to y, so the coefficient at y' is $6 \equiv 0 \mod 3$. If y' is not joined to y, then y' is joined to two neighbours of y in X and to four non-neighbours, so the coefficient at y' is $4 \equiv 1 \mod 3$. Thus the sum of our sixteen vectors is $j - r_y$, showing that r_y lies in the span of our 23 chosen vectors.

Notice incidentally that $B^2 = 2B$, so that the minimum polynomial of B is the product of distinct linear factors, so B is diagonalisable. □

We also found other strongly regular triangle-free graphs which have guessing number larger than the lower bound given by fractional clique cover. See [24, Chapter 8] for further details about these graphs.

Proposition 7. *The following triangle-free graphs on n vertices have their guessing number larger than $n/2$:*

(a) The Clebsch graph on 16 *vertices has* $10 \leq \mathrm{gn}(G) \leq 11$.

(b) The Hoffman–Singleton graph on 50 *vertices has* $29 \leq \mathrm{gn}(G) \leq 35$.

(c) The Gewirtz graph on 56 *vertices has* $36 \leq \mathrm{gn}(G) \leq 40$.

(d) The M_{22} *graph on* 77 *vertices has* $55 \leq \mathrm{gn}(G) \leq 56$.

(e) The Higman-Sims graph on 100 *vertices has* $77 \leq \mathrm{gn}(G) \leq 78$.

The triangle-free graphs presented in Proposition 7 are strongly regular graph with parameters (n, k, λ, μ) where n is the number of vertices, k is the degree, λ is the number of common neighbours of every two adjacent vertices, and μ is the number of common neighbours of every two non-adjacent vertices.

Proof. The upper bound for the guessing number derived for these graphs is $\text{gn}(G) \leq |V(G)| - \alpha(G)$ where $\alpha(G)$ is the independence number [82]. The independence numbers of the Clebsch graph, the Hoffman–Singleton graph, and the Gewirtz graph are 5, 15, and 16, respectively [18]. The independence number of the M_{22} graph is 21, as shown below.

(a) The Clebsch graph is a triangle-free strongly regular graph with parameters $(16, 5, 0, 2)$ which is constructed as follows: We start with a finite set $S = \{1, 2, 3, 4, 5\}$. The set V contains all subsets of size 1, and 2 of S, and V also contains an extra single point set $\{*\}$. We form the $(16, 5, 0, 2)$ graph with vertex set V and an edge between two vertices u and v if one of the following conditions is satisfied:

- u is $\{\infty\}$, and v is a subset of S with cardinal 1.
- u is a subset of S with cardinal 1, v is a subset of S with cardinal 2, and u is a subset of v.
- u and v are subsets of S with cardinal 2, and u intersect v is empty.

We have the rank of the matrix $A + I$ is 6 over finite field $\mathbb{F}_2$. A basis of $A + I$ is $\{j\} \cap \{r_v | v \in \Gamma(\infty)\}$, where r_v is the row of $A + I$ corresponding to vertex v and j is the all–1 vector.

(b) The Hoffman-Singleton graph which is triangle-free strongly regular with parameters $(50, 7, 0, 1)$ has one way of construction as follows: We take five 5-cycles C_h and their complements C_i^c, and we join vertex j of C_h to vertex $hi + j \mod 5$ of C_i^c. This construction is due to Conway. The rank of the matrix $A + 3I$ over finite field $\mathbb{F}_5$ is 21[1]. A basis for this matrix over $\mathbb{F}_5$ is recorded in the file `Basis.txt` which can be downloaded from `https://www.eecs.qmul.ac.uk/~smriis/`. This file also includes a description for coordinates of each row in $A + 3I$ over $\mathbb{F}_5$ with respect to the given basis.

(c) The Gewirtz graph with parameters $(56, 10, 0, 2)$ can be constructed from the $S(3, 6, 22)$ by fixing an element and let the vertices be the 56 blocks not containing that element. Two vertices are adjacent if the intersection of their corresponding blocks is empty. The rank of the matrix $A + I$ over finite field $\mathbb{F}_3$ is 20. A basis for this matrix over $\mathbb{F}_3$ is recorded in the file `Basis.txt`.

[1]Brouwer and Van Eijl derived the same result for $A + 3I$ over $\mathbb{F}_5$ [19, pages 340, 341] using eigenvalue method.

(d) The triangle-free strongly regular graph M_{22} with parameters $(77, 16, 0, 4)$ which can be constructed by let the 77 blocks of $S(3, 6, 22)$ be the vertices of the graph, and an edge uv between two vertices u and v if u and v are disjoint as blocks. Note that this is the induced subgraph of the Higman–Sims graph on the set of non-neighbours of ∞.

To see that its independence number is 21, note that the vertices other than ∞ non-adjacent to a vertex in X in the Higman–Sims graph form an independent set of size 21; and there is no larger independent set, since all independent sets of size 22 in the Higman–Sims graph are vertex neighbourhoods.

The rank of $A+I$ over finite field $\mathbb{F}_3$ is 22. A basis for this matrix over $\mathbb{F}_3$ is recorded in the file `Basis.txt`.

(e) Theorem 16. □

Remark 6. For a more extensive list of computations of ranks of matrices $A+kI$ over $\mathbb{F}_q$ for $q = 2, 3, 5, 7$ see `EBasis.zip` at `https://www.eecs.qmul.ac.uk/~smriis/`.

Chapter 5

Shannon and Non-Shannon Information Bounds

5.1 Graphs with guessing numbers matching Shannon bounds

In this section, we derive the exact guessing number of some new families of undirected graphs.

5.1.1 Primarily

Firstly, let us introduce a definition concerning sum of two undirected graphs which will be used throughout this section. Our definition mimics the definition the wedge sum of two spaces established in the context of topology.

Definition 18. *Let G, H be undirected graphs with vertex sets $V(G) = \{v_1, v_2, \ldots, v_n\}$ and $V(H) = \{u_1, u_2, \ldots, u_m\}$ respectively. We define a wedge sum*

$$K := G \bigvee_{v_{i_1} \equiv u_{j_1}, \ldots, v_{i_k} \equiv u_{j_k}} V$$

to be an undirected graph with the following data:

- *$V(K)$ is a quotient set of the disjoint union of $V(G)$ and $V(H)$ by the identification $v_{i_1} \sim u_{j_1}, \ldots, v_{i_k} \sim u_{j_k}$.*
- *$(u, v) \in E(K)$ if and only if (u, v) is either an edge of G or H.*

All undirected graphs presented in this section have their guessing numbers obtained using the fractional clique cover strategy. A special case of this strategy is the clique cover strategy which is defined as follows:

Let G be an undirected graph with vertex set V. We can partition $V(G)$ into vertex disjoint cliques $V_1, \ldots, V_k$. When we play a guessing game (G, s), a strategy, which each group of players corresponding to a clique V_i follows the clique strategy described in 3.1, is a proper guessing strategy.

Definition 19. *[28] The clique cover number $\kappa(G)$ of an undirected graph G is the minimum cardinality of a clique cover of G.*

Note that the complement of a clique is an independent set, hence a clique cover of G induce a proper vertex colouring of G^c-the complement of G. Therefore, $\kappa(G) = \chi(G^c)$.

The following fact can be deduced from our definition of clique cover number:

Proposition 8. *[28]*

1. *For every graph G and every positive integer s, $\mathrm{gn}(G, s) > |V(G)| - \kappa(G)$.*
2. *For every graph G and every positive integer s, $\mathrm{gn}(G, s) \leq n - \alpha(G)$ where $\alpha(G)$ denotes the independence number of G.*

If G satisfies that $\alpha(G) = \kappa(G)$, then, following the previous proposition, we have $\mathrm{gn}(G) = n - \alpha(G)$. The class of perfect graphs introduced by Berge [10] is a natural class which satisfies this property. A graph is said to be perfect if $\chi(H) = \omega(H)$ for all induced subgraphs H of G.

Corollary 4. *[28] If G is perfect then*

$$\mathrm{gn}(G) = n - \alpha(G).$$

Proof. By Lovász's perfect graph theorem [70], we know that a graph is perfect if and only if its complement is perfect.

We have $\alpha(G) = \omega(G^c) = \chi(G^c) = \kappa(G)$. □

Remark 7. The strong perfect graph theorem [29] tells us that a graph is perfect if and only if it does not contain an odd hole (odd-length induced cycle of length greater than 4) or an odd antihole (complement of an odd hole of length greater than 4).

When G is not perfect, we will try to compute an upper-bound of $\mathrm{gn}(G)$ using information inequalities and graph constraints. Recall that for each $v \in V(G)$, we define the discrete random variable X_v on the probability space on the set of all assignment tuples $A_s^{|V(G)|}$ to be the value assigned to vertex v. The associated entropy value of X_v is denoted $H(X_v)$.

Follow the discussion in Section 3.5, we have the following proposition characterizing the information upper-bound of guessing number:

Proposition 9. *Let $G = (V, E)$ be an undirected graph with $V = \{1, 2, \ldots, n\}$. Given $X, Y, Z \subset X_G = \{X_1, \ldots, X_n\}$,*

positivity $H(X) \geq 0$.

normality $H(X) \leq |X|$.

submodularity *Shannon's information inequality:*

$$H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z) \geq 0.$$

graph constraints *Suppose $A, B \subset V(G)$ with $\Gamma^-(u) \subset B$ for all $u \in A$. Let $X = \{X_v : v \in A\}$ and $Y = \{X_v : v \in B\}$. Then*

$$H(X, Y) = H(Y).$$

Proposition 9 describes a linear program to upper bound $H(X_G)$. In particular the linear program consists of $2^{|V(G)|}$ variables corresponding to the values of $H(X)$ for each $X \subset X_G$. The variables are constrained by the linear inequalities given in Proposition 9 and the objective is to maximize the value of the variable corresponding to $H(X_G)$. We call the result of the optimization *the Shannon bound* of G and denote it by $\mathrm{Sh}(G)$.

All calculations in this section will explicitly use the linear program described above. To ease the notation, we write $H(I)$ for $H(X_I)$, and $H(G)$ for $H(X_G)$ where I is a subset of vertices of $V(G)$.

Now we are ready to compute the guessing numbers of some families of undirected graphs.

Remark 8. Unless otherwise stated, we always assume that our graphs are triangle-free.

5.1.2 The guessing number of a Theta graph

Definition 20. *Let C_m be the m-cycle undirected graph with vertex set $\{1, 2, \ldots, k, k+1, \ldots, m\}$, and P_h be the path graph of order h with vertex set $\{1', 2', \ldots, h'\}$. A theta graph (l, m, h) is a planar triangle-free undirected graph which is isomorphic to the graph $K := C_m \bigvee_{1\equiv 1', k\equiv h'} P_h$.*

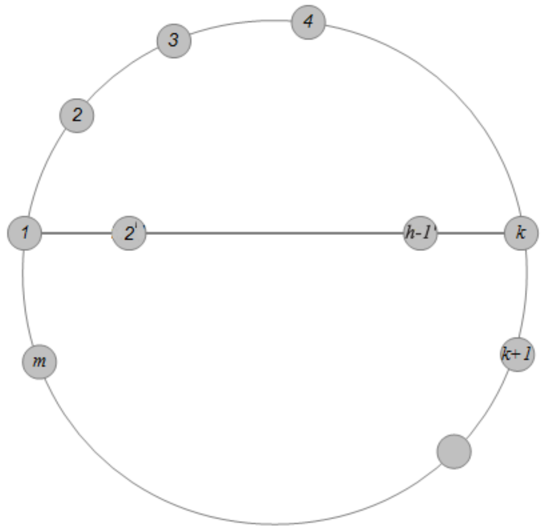


Figure 5.1: The undirected Theta graph.

Theorem 17. *Let G be a theta graph (l, m, h) of order $n := m + h - 2$. The guessing number of G is*

$$\mathrm{gn}(G) = \frac{n}{2}.$$

Proof. Since G is triangle-free, the fractional clique cover strategy provides a lower bound $\frac{n}{2} \leq \mathrm{gn}(G)$ by Lemma 6. Therefore, all we need to show is $\mathrm{gn}(G) \leq \mathrm{Sh}(G) = \frac{n}{2}$.

To compute $\mathrm{Sh}(G)$, we will break the computation into four cases:

1. n is even and h is even.
2. n is even and h is odd.
3. n is odd and h is even.
4. n is odd and h is odd.

The first case is straightforward. If both n and h are even numbers, then G is a perfect graph, and the result follows from our Corollary 4. The second case is established in Lemma 11. The third and fourth cases are proved in Lemma 12 illustrated below. □

Lemma 11. *Let G be a theta graph (l, m, h) of even order $n := m+h-2$. Moreover, we assume that h is an odd integer. The guessing number of G is*

$$\mathrm{gn}(G) = \frac{n}{2}.$$

Proof. The graph constraints give us the following equalities:

- $H(1|2, m, 2') = 0$,
- $H(i|i-1, i+1 \mod m) = 0$ for $i \in \{2, 3, \ldots, k-1, k+1, \ldots, m\}$,
- $H(k|k-1, k+1, (h-1)') = 0$,
- $H((i)'|(i-1)', (i+1)') = 0$ for $2 \leq i \leq h-1$. (Note that $1' \sim 1$ and $k \sim h'$.)

In particular, if k is odd, we have:

$$\begin{aligned}
H(G) &= H(G|1, 3, 5, \ldots, k, k+2, \ldots, m, 3', 5', \ldots, (h-2)') \\
&\quad + H(1, 3, 5, \ldots, m, 3', 5', \ldots, (h-2)') && \text{(conditional entropy)} \\
&= H(1, 3, 5, \ldots, m, 3', 5', \ldots, (h-2)') && \text{(graph constraints)} \\
&\leq |\{1, 3, 5, \ldots, m, 3', 5', \ldots, (h-2)'\}| && \text{(normality)} \\
&= n/2 = \mathrm{Sh}(G).
\end{aligned}$$

If k is even, we have:

$$\begin{aligned}
H(G) &= H(G|1, 3, 5, \ldots, m, 2', 3', \ldots (h-1)') && && (5.1) \\
&\quad + H(1, 3, 5, \ldots, m, 2', 3', \ldots (h-1)') && \text{(c.e.)} && (5.2) \\
&= H(1, 3, 5, \ldots, m, 2', 3', \ldots (h-1)') && \text{(g.c.)} && (5.3) \\
&\leq H(1, m, 2', 3', \ldots, (h-1)') && && (5.4) \\
&\quad + H(3, 5, \ldots, m-2) && \text{(submodularity)} && (5.5) \\
&\leq H(1, m, 2', 3', \ldots, (h-1)') && && (5.6) \\
&\quad + |\{3, 5, \ldots, m-2\}| && \text{(n.)} && (5.7) \\
&= H(1, m, 2', 3', \ldots, (h-1)') + \frac{m-3}{2}. && && (5.8)
\end{aligned}$$

Similarly,

$$\begin{aligned}
H(G) &= H(G|1,2,4,\ldots,m-1,m,2',3',\ldots,(h-1)') && (5.9)\\
&+ H(1,2,4,\ldots,m-1,m,2',3',\ldots,(h-1)') \quad \text{(c.e.)} && (5.10)\\
&= H(1,2,4,\ldots,m-1,m,2',3',\ldots,(h-1)') \quad \text{(g.c.)} && (5.11)\\
&\le H(1,2,k,m-1,m,2',3',\ldots,(h-1)') && (5.12)\\
&+ H(4,6,\ldots,k-2,k+2,k+4,\ldots,m-3) \quad \text{(s.)} && (5.13)\\
&\le H(1,2,k,m-1,m,2',3',\ldots,(h-1)') && (5.14)\\
&+ |\{4,6,\ldots,k-2,k+2,k+4,\ldots,m-3\}| \quad \text{(n.)} && (5.15)\\
&= H(1,2,k,m-1,m,2',3',\ldots,(h-1)') + \frac{m-7}{2}. && (5.16)
\end{aligned}$$

Adding 5.8 and 5.16 we get that:

$$\begin{aligned}
2H(G) &\le H(1,m,2',3',\ldots,(h-1)')\\
&+ H(1,2,k,m-1,m,2',3',\ldots,(h-1)')\\
&+ \frac{m-3}{2} + \frac{m-7}{2}\\
&\le H(1,2,m,2',3',\ldots,(h-1)')\\
&+ H(1,m-1,m,k,2',3',\ldots,(h-1)')\\
&+ \frac{m-3}{2} + \frac{m-7}{2} && \text{(s.)}\\
&= H(1,2,m,2',3',\ldots,(h-1)'|2,15,2',4',6',\ldots,(h-1)')\\
&+ H(2,m,2',4',6',\ldots,(h-1)')\\
&+ H(1,m-1,m,k,2',3',\ldots,(h-1)'|1,m-1,k,3',5',\ldots,(h-2)')\\
&+ H(1,m-1,k,3',5',\ldots,(h-2)')\\
&+ m-5. && \text{(c.e.)}\\
&= H(2,m,2',4',6',\ldots,(h-1)')\\
&+ H(1,m-1,k,3',5',\ldots,(h-2)')\\
&+ m-5. && \text{(g.c.)}\\
&\le |\{2,m,2',4',6',\ldots,(h-1)'\}|\\
&+ |\{1,m-1,k,3',5',\ldots,(h-2)'\}|\\
&+ m-5. && \text{(n.)}\\
&= n.
\end{aligned}$$

Therefore, $\mathrm{Sh}(G) = n/2$. $\square$

Lemma 12. *Let G be a theta graph (l, m, h) of odd order $n := m + h - 2$. The guessing number of G is*

$$\mathrm{gn}(G) = \frac{n}{2}.$$

Proof. Since G has an odd order, we know that within the three cycles

- $\{1, 2, \ldots, k-1, k, k+1, \ldots, m\}$,
- $\{1, 2, \ldots, k-1, k, (h-1)', (h-2)', \ldots, (2)'\}$,
- and $\{1, (2)', \ldots, (h-1)', k, k+1, \ldots, m\}$

at least one of them has odd length. Applying graph isomorphism, we can always assume that m is odd.

The graph constraints give us the following equalities:

- $H(1|2, m, 2') = 0$,
- $H(i|i-1, i+1 \mod m) = 0$ for $i \in \{2, 3, \ldots, k-1, k+1, \ldots, m\}$,
- $H(k|k-1, k+1, (h-1)') = 0$,
- $H((i)'|(i-1)', (i+1)') = 0$ for $2 \leq i \leq h-1$. (Note that $1' \sim 1$ and $k \sim h'$.)

If k is odd, we have:

$$\begin{aligned}
H(G) &= H(G|1, 3, 5, \ldots, m-2, m, 2', 4', \ldots, (h-2)') && (5.17)\\
&+ H(1, 3, 5, \ldots, m-2, m, 2', 4', \ldots, (h-1)') \quad \text{(c.e.)} && (5.18)\\
&= H(1, 3, 5, \ldots, m-2, m, 2', 4', \ldots, (h-1)') \quad \text{(g.c.)} && (5.19)\\
&\leq H(1, m, 2') && (5.20)\\
&+ H(3, 5, \ldots, m-2, 4', 6', \ldots, (h-1)') \quad \text{(s.)} && (5.21)\\
&\leq H(1, m, 2') && (5.22)\\
&+ |\{3, 5, \ldots, m-2, 4', 6', \ldots, (h-1)'\}| \quad \text{(n.)} && (5.23)\\
&= H(1, m, 2') + \frac{n+1}{2} - 3. && (5.24)
\end{aligned}$$

Likewise, we have

$$\begin{aligned}
H(G) &= H(G|1,2,4,6,\ldots,m-3,m-1,m,2',3',5',\ldots,(h-1)') && && (5.25)\\
&+ H(1,2,4,6,\ldots,m-3,m-1,m,2',3',5',\ldots,(h-1)') && \text{(c.e.)} && (5.26)\\
&= H(1,2,4,6,\ldots,m-3,m-1,m,2',3',5',\ldots,(h-1)') && \text{(g.c.)} && (5.27)\\
&\le H(1,2,m-1,m,2',3') && && (5.28)\\
&+ H(4,6,\ldots,m-3,5',7',\ldots,(h-1)') && \text{(s.)} && (5.29)\\
&\le H(1,2,m-1,m,2',3') && && (5.30)\\
&+ |\{4,6,\ldots,m-3,5',7',\ldots,(h-1)'\}| && \text{(n.)} && (5.31)\\
&= H(1,2,m-1,m,2',3') + \frac{n+1}{2} - 4 && && (5.32)
\end{aligned}$$

Adding 5.24 and 5.32 we get that

$$\begin{aligned}
2H(G) &\le H(1,m,2') + H(1,2,m-1,m,2',3') + n - 6 && \\
&\le H(m,2',1,2) + H(m-1,m,2',3',1) + n - 6 && \text{(s.)}\\
&= H(m,2',1,2|m,2',2) + H(m,2',2) && \\
&+ H(m-1,m,2',3',1|m-1,1,3') + H(m-1,1,3') && \\
&+ n - 6 && \text{(c.e.)}\\
&= H(m,2',2) + H(m-1,3',1) + n - 6 && \text{(g.c.)}\\
&\le |\{m,2',2\}| + |\{m-1,3',1\}| + n - 6 && \text{(n.)}\\
&= n.
\end{aligned}$$

Therefore, $\mathrm{Sh}(G) = n/2$.

If k is even, we have that

$$\begin{aligned}
H(G) &= H(G|1,3,5,\ldots,m-2,m,3',5',\ldots,(h-1)') && && (5.33)\\
&+ H(1,3,5,\ldots,m-2,m,3',5',\ldots,(h-1)') && \text{(c.e.)} && (5.34)\\
&= H(1,3,5,\ldots,m-2,m,3',5',\ldots,(h-1)') && \text{(g.c.)} && (5.35)\\
&\le H(1,m) && && (5.36)\\
&+ H(3,5,\ldots,m-2,3',5',\ldots,(h-1)') && \text{(s.)} && (5.37)\\
&\le H(1,m) && && (5.38)\\
&+ |\{3,5,\ldots,m-2,3',5',\ldots,(h-1)'\}| && \text{(n.)} && (5.39)\\
&= H(1,m) + \frac{n+1}{2} - 2. && && (5.40)
\end{aligned}$$

Likewise, we have

$$
\begin{aligned}
H(G) &= H(G|1,2,4,6,\ldots,m-3,m-1,m,2',4',6',\ldots,(h-2)') && && (5.41)\\
&+ H(1,2,4,6,\ldots,m-3,m-1,m,2',4',6',\ldots,(h-2)') && \text{(c.e.)} && (5.42)\\
&= H(1,2,4,6,\ldots,m-3,m-1,m,2',4',6',\ldots,(h-2)') && \text{(g.c.)} && (5.43)\\
&\le H(1,2,m-1,m,2') && && (5.44)\\
&+ H(4,6,\ldots,m-3,4',6',\ldots,(h-2)') && \text{(s.)} && (5.45)\\
&\le H(1,2,m-1,m,2') && && (5.46)\\
&+ |\{4,6,\ldots,m-3,4',6',\ldots,(h-2)'\}| && \text{(n.)} && (5.47)\\
&= H(1,2,m-1,m,2') + \frac{n+1}{2} - 4 && && (5.48)
\end{aligned}
$$

Adding 5.40 and 5.48 we get that

$$
\begin{aligned}
2H(G) &\le H(1,m) + H(1,2,m-1,m,2') + n - 5 \\
&\le H(m,2',1,2) + H(m-1,m,1) + n - 5 && \text{(s.)}\\
&= H(m,2',1,2|m,2',2) + H(m,2',2)\\
&+ H(m-1,m,1|m-1,1) + H(m-1,1)\\
&+ n - 5 && \text{(c.e.)}\\
&= H(m,2',2) + H(m-1,1) + n - 5 && \text{(g.c.)}\\
&\le |\{m,2',2\}| + |\{m-1,1\}| + n - 5 && \text{(n.)}\\
&= n.
\end{aligned}
$$

Therefore, $\mathrm{Sh}(G) = n/2$. □

Remark 9. The guessing number of Theta graph was independently computed by Christofides and Markström (private communication).

5.1.3 The guessing number of a lollipop graph

Definition 21. *Let C_m be the m-cycle undirected graph with vertex set $\{1,2,\ldots,k,k+1,\ldots,m\}$, and P_h be the path graph of order h with vertex set $\{1',2',\ldots,h'\}$. A lollipop graph (m,h) is a planar triangle-free undirected graph which is isomorphic to the graph $K := C_m \bigvee_{1\equiv 1'} P_h$.*

Theorem 18. *Let G be a lollipop graph (m,h) of order $n := m+h-1$. The guessing number of G is*

$$\mathrm{gn}(G) = \frac{n}{2}.$$

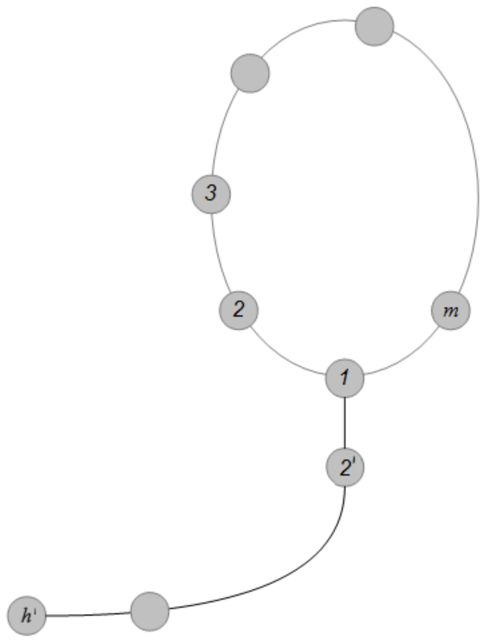


Figure 5.2: The Lollipop graph.

Proof. Since G is triangle-free, the fractional clique cover strategy provides a lower bound $\frac{n}{2} \leq \text{gn}(G)$ by Lemma 6. Therefore, all we need to show is $\text{gn}(G) \leq \text{Sh}(G) = \frac{n}{2}$.

We see that one can obtain a lollipop graph from a theta graph of same order by removing one edge of a vertex of degree 3 in the theta graph. Therefore, it is clear that the guessing number of a lollipop graph is at most the guessing number of a theta graph of same order. Our claim is now a Corollary of Theorem 17. □

5.1.4 The guessing number of a spiral graph

Definition 22. *Let* $\mathcal{C} := \{C_{n_1}, C_{n_2}, \ldots, C_{n_k}\}$ *be a collection of* $\{n_i\}_{i=1}^{k}$*-cycle graphs. We denote the vertex set of* C_{n_i} *as* $\{1_i, 2_i, \ldots, (n_i)_i\}$*. A spiral graph* $\bigvee \mathcal{C}$ *is a planar triangle-free undirected graph which is isomorphic to the graph* $K := \bigvee_{1_1 \equiv 1_2 \equiv \ldots \equiv 1_k} (C | C \in \mathcal{C}$*.*

Theorem 19. *Let* G *be a spiral graph* $\bigvee \mathcal{C}$ *of order* $n := \sum_{i=1}^{k} n_i - k + 1$*. The guessing number of* G *is*

$$\text{gn}(G) = \frac{n}{2}.$$

Proof. The Spiral graph can be thought of as attaching k different cycles $\{C_{n_i}\}_{i=1,\ldots,k}$ with each other at vertex $1_1 \equiv 1_2 \equiv \ldots \equiv 1_k$. Therefore, the process for computing

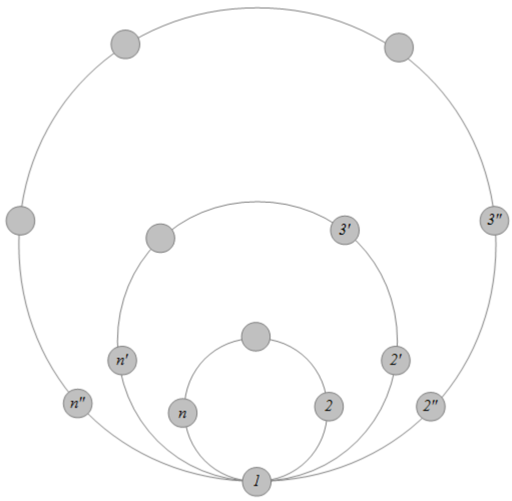


Figure 5.3: An example of a spiral graph.

the upper-bound of this graph is essentially a generalization of the computation used to derive Shannon's bound for the cycle graph C_n. For the sake of completeness, we will prove here for the case when G is obtained by attaching two different cycles C_m and C_n. The order of G is $n + m - 1$. For the general case, the deriving process can be modified accordingly without any obstruction.

Since G is triangle-free, the fractional clique cover strategy provides a lower bound $\frac{n+m-1}{2} \leq \mathrm{gn}(G)$ by Lemma 6. Therefore, all we need to show is $\mathrm{gn}(G) \leq \mathrm{Sh}(G) = \frac{n+m-1}{2}$.

To compute $\mathrm{Sh}(G)$, we will break the computation into three cases:

1. m and n are even.

2. m and n are odd.

3. m is even and n is odd.

For the first case, G is a perfect graph, so the result follows and the result follows from our Corollary 4.

For a clear demonstration, we will denote the vertices in G as 1, $2_m, \ldots m_m$, and $2_n, \ldots, n_n$.

If m and n are odd, we have:

$$
\begin{aligned}
H(G) &= H(G|1, 3_m, 3_n, \ldots, m_m, n_n) && && (5.49)\\
&\quad + H(1, 3_m, 3_n, \ldots, m_m, n_n) && \text{(c.e.)} && (5.50)\\
&= H(1, 3_m, 3_n, \ldots, m_m, n_n) && \text{(g.c.)} && (5.51)\\
&\leq H(1, m_m, n_n) && && (5.52)\\
&\quad + H(3_m, 3_n, \ldots, (m-2)_m, (n-2)_n) && \text{(s.)} && (5.53)\\
&\leq H(1, m_m, n_n) && && (5.54)\\
&\quad + |\{3_m, 3_n, \ldots, (m-2)_m, (n-2)_n\}| && \text{(n.)} && (5.55)\\
&= H(1, m_m, n_n) + \frac{n+m}{2} - 3. && && (5.56)
\end{aligned}
$$

Likewise, we have

$$
\begin{aligned}
H(G) &= H(G|1, 2_m, 2_n, 4_m, 4_n, \ldots, (m-1)_m, (n-1)_n, m_m, n_n) && && (5.57)\\
&\quad + H(1, 2_m, 2_n, 4_m, 4_n, \ldots, (m-1)_m, (n-1)_n, m_m, n_n) && \text{(c.e.)} && (5.58)\\
&= H(1, 2_m, 2_n, 4_m, 4_n, \ldots, (m-1)_m, (n-1)_n, m_m, n_n) && \text{(g.c.)} && (5.59)\\
&\leq H(1, 1, 2_m, 2_n, (m-1)_m, (n-1)_n, m_m, n_n) && && (5.60)\\
&\quad + H(4_m, 4_n, \ldots, (m-3)_m, (n-3)_n) && \text{(s.)} && (5.61)\\
&\leq H(1, 1, 2_m, 2_n, (m-1)_m, (n-1)_n, m_m, n_n) && && (5.62)\\
&\quad + |\{4_m, 4_n, \ldots, (m-3)_m, (n-3)_n\}| && \text{(n.)} && (5.63)\\
&= H(1, 1, 2_m, 2_n, (m-1)_m, (n-1)_n, m_m, n_n) + \frac{n+m}{2} - 5. && && (5.64)
\end{aligned}
$$

Adding 5.56 and 5.64 we get:

$$
\begin{aligned}
2H(G) &\leq H(1, m_m, n_n) \\
&+ H(1, 1, 2_m, 2_n, (m-1)_m, (n-1)_n, m_m, n_n) \\
&+ n + m - 8 \\
&\leq H(1, (m-1)_m, (n-1)_n, m_m, n_n) \\
&+ H(1, 2_m, 2_n, m_m, n_n) \\
&+ m + n - 8 && \text{(s.)} \\
&= H(1, (m-1)_m, (n-1)_n, m_m, n_n | 1, (m-1)_m, (n-1)_n) \\
&+ H(1, (m-1)_m, (n-1)_n) \\
&+ H(1, 2_m, 2_n, m_m, n_n | 2_m, 2_n, m_m, n_n) \\
&+ H(2_m, 2_n, m_m, n_n) \\
&+ m + n - 8 && \text{(c.e.)} \\
&= H(1, (m-1)_m, (n-1)_n) + H(2_m, 2_n, m_m, n_n) + m + n - 8 && \text{(g.c.)} \\
&\leq |\{1, (m-1)_m, (n-1)_n\}| + |\{2_m, 2_n, m_m, n_n\}| + m + n - 8 && \text{(n.)} \\
&= m + n - 1.
\end{aligned}
$$

The result follows.

If m is odd and n is even, we have:

$$H(G) = H(G | 3_m, 4_m, 6_m, \ldots (m-1)_m, 1, 3_n, \ldots, (n-1)_n) \tag{5.65}$$

$$+ H(3_m, 4_m, 6_m, \ldots (m-1)_m, 1, 3_n, \ldots, (n-1)_n) \quad \text{(c.e.)} \tag{5.66}$$

$$= H(3_m, 4_m, 6_m, \ldots (m-1)_m, 1, 3_n, \ldots, (n-1)_n) \quad \text{(g.c.)} \tag{5.67}$$

$$\leq H(3_m, 4_m) + H(6_m, \ldots, (m-1)_m, 1, 3_n, \ldots, (n-1)_n) \quad \text{(s.)} \tag{5.68}$$

$$\leq H(3_m, 4_m) + |\{6_m, \ldots, (m-1)_m, 1, 3_n, \ldots, (n-1)_n\}| \quad \text{(n.)} \tag{5.69}$$

$$= H(3_m, 4_m) + \frac{n + m + 1}{2} - 3. \tag{5.70}$$

Likewise, we have

$$
\begin{aligned}
H(G) &= H(G|2_m, 3_m, 4_m, 5_m, 7_m, \ldots, m_m, 2_n, 4_n, \ldots, n_n) && (5.71)\\
&+ H(2_m, 3_m, 4_m, 5_m, 7_m, \ldots, m_m, 2_n, 4_n, \ldots, n_n) \quad \text{(c.e.)} && (5.72)\\
&= H(2_m, 3_m, 4_m, 5_m, 7_m, \ldots, m_m, 2_n, 4_n, \ldots, n_n) \quad \text{(g.c.)} && (5.73)\\
&\leq H(2_m, 3_m, 4_m, 5_m) + H(7_m, 9_m, \ldots, m_m, 2_n, 4_n, \ldots, n_n) \quad \text{(s.)} && (5.74)\\
&\leq H(2_m, 3_m, 4_m, 5_m) + |\{7_m, 9_m, \ldots, m_m, 2_n, 4_n, \ldots, n_n\}| \quad \text{(n.)} && (5.75)\\
&= H(2_m, 3_m, 4_m, 5_m) + \frac{n+m+1}{2} - 3. && (5.76)
\end{aligned}
$$

Adding 5.70 and 5.76 we get:

$$
\begin{aligned}
2H(G) &\leq H(3_m, 4_m) + H(2_m, 3_m, 4_m, 5_m) + n + m - 5\\
&\leq H(2_m, 3_m, 4_m) + H(3_m, 4_m, 5_m) + m + n - 5 && \text{(s.)}\\
&= H(2_m, 3_m, 4_m|2_m, 4_m) + H(2_m, 4_m)\\
&+ H(3_m, 4_m, 5_m|3_m, 5_m) + H(3_m, 5_m) + m + n - 5 && \text{(c.e.)}\\
&= H(2_m, 4_m) + H(3_m, 5_m) + m + n - 5 && \text{(g.c.)}\\
&\leq |\{2_m, 4_m\}| + |\{3_m, 5_m\}| + m + n - 5 && \text{(n.)}\\
&= m + n - 1.
\end{aligned}
$$

The result follows. □

Remark 10. The proof presented above is just a special case of the Theorem 17. However, we decided to illustrate a different proof as the computation process in this case is much simpler and can be generalized for the Spiral graphs.

Remark 11. One can extend the proof above to the case where $G \cong C_m \bigvee_{1 \equiv v} T$ where C_m is an m-cycle with vertex set $\{1, 2, \ldots, m\}$ and T is a tree and $v \in V(T)$.

Following the previous remarks, we obtain:

Theorem 20. *If G is a planar triangle-free graph having at most one vertex of degree greater than 2, then the guessing number of G is*

$$\text{gn}(G) = \frac{|V(G)|}{2}.$$

5.1.5 The guessing number of a $C_n \Box P_k$ graph

Definition 23. *Let G and H be undirected graphs. The Cartesian product $G \Box H$ is a graph such that:*

- $V(G \Box H) = V(G) \times V(H)$,
- *two vertices* (u, u') *and* (v, v') *are adjacent in* $G \Box H$ *if and only if wither* $u = v$ *and* $(u', v') \in E(H)$, *or* $u' = v'$ *and* $(u, v) \in V(G)$.

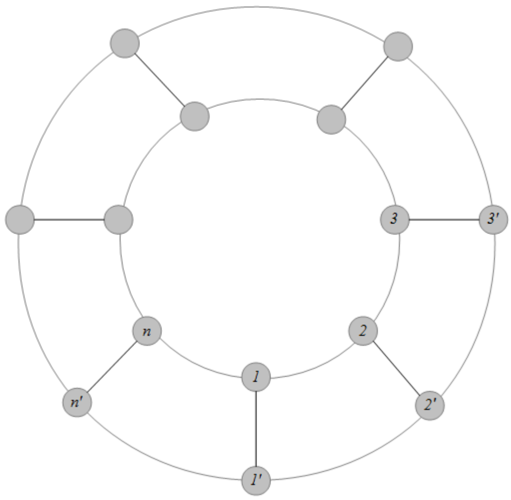


Figure 5.4: The graph $C_n \Box P_2$.

Theorem 21. *If* G *is a planar triangle-free graph which is isomorphic to* $C_n \Box P_k$ *for some positive integers* $n \geq 4$ *and* $k \geq 1$, *then the guessing number of* G *is*

$$\mathrm{gn}(G) = \frac{nk}{2}.$$

Proof. We will denote vertices of C_n as $1, \ldots, n$ with $\Gamma_i := \{i-1, i+1\}$ where addition and subtraction are done modulo n. We also denote the vertices of P_k as $1, \ldots, k$.

Since G is obtained as the Cartesian product of C_n and P_k, its vertices are indexed by a pair (i, j) where $1 \leq i \leq n$, and $1 \leq j \leq k$.

Similar to the previous theorem, we will only illustrate the computation for the product $C_n \Box P_2$ as for the general case $C_n \Box P_k$ the computation can be modified accordingly without any obstruction.

Since G is triangle-free, the fractional clique cover strategy provides a lower bound $n \leq \mathrm{gn}(G)$ by Lemma 6. Therefore, all we need to show is $\mathrm{gn}(G) \leq \mathrm{Sh}(G) = n$.

To compute $\mathrm{Sh}(G)$, we will break the computation into two cases:

1. n is even,
2. n is odd.

Since $k = 2$, we will ease the notation by writing i for $(i,1)$ and i' for $(i,2)$.

If C_n is an even cycle, then we have

$$\begin{aligned} H(G) &= H(G|1,2',3,\ldots,n') + H(1,2',3,\ldots,n') && \text{(c.e.)} \\ &= H(1,2',3,\ldots,n') && \text{(g.e.)} \\ &\leq |\{1,2',3,\ldots,n'\}| && \text{(n.)} \\ &= n. \end{aligned}$$

This proves the first case.

If C_n is an odd cycle, then we have

$$\begin{aligned} H(G) &= H(G|1,2,2',3',4,5',6,7',\ldots,n-1,n') && (5.77) \\ &+ H(1,2,2',3',4,5',6,7',\ldots,n-1,n') \quad \text{(c.e.)} && (5.78) \\ &= H(1,2,2',3',4,5',6,7',\ldots,n-1,n') \quad \text{(g.c.)} && (5.79) \\ &\leq H(1,2,2',3') + H(4,5',6,7',\ldots,n-1,n') \quad \text{(s.)} && (5.80) \\ &\leq H(1,2,2',3') + |\{4,5',6,7',\ldots,n-1,n'\}| \quad \text{(n.)} && (5.81) \\ &= H(1,2,2',3') + n - 3. && (5.82) \end{aligned}$$

Likewise, we have

$$\begin{aligned} H(G) &= H(G|n,1,1',2,2',3,3',4',5,6',7,\ldots,n-2,(n-1)') && (5.83) \\ &+ H(n,1,1',2,2',3,3',4',5,6',7,\ldots,n-2,(n-1)') \quad \text{(c.e.)} && (5.84) \\ &= H(n,1,1',2,2',3,3',4',5,6',7,\ldots,n-2,(n-1)') \quad \text{(g.c.)} && (5.85) \\ &\leq H(n,1,1',2,2',3,3',4) + H(5,6',7,\ldots,n-2,(n-1)') \quad \text{(s.)} && (5.86) \\ &\leq H(n,1,1',2,2',3,3',4) + |\{5,6',7,\ldots,n-2,(n-1)'\}| \quad \text{(n.)} && (5.87) \\ &= H(n,1,1',2,2',3,3',4) + n - 5. && (5.88) \end{aligned}$$

Adding 5.82 and 5.88 we get:

$$\begin{aligned} 2H(G) &\leq H(1,2,2',3') + H(n,1,1',2,2',3,3',4') + 2n - 8 \\ &\leq H(n,1,1',2,2',3') + H(1,2,2',3,3',4') + 2n - 8 && \text{(s.)} \\ &= H(n,1,1',2,2',3'|n,1',2,3') + H(n,1',2,3') \\ &+ H(1,2,2',3,3',4'|1,2',3,4') + H(1,2',3,4') + 2n - 8 && \text{(c.e.)} \\ &= H(n,1',2,3') + H(1,2',3,4') + 2n - 8 && \text{(g.c.)} \\ &\leq |\{n,1',2,3'\}| + |\{1,2',3,4'\}| + 2n - 8 && \text{(n.)} \\ &= 2n. \end{aligned}$$

The proof is completed. ∎

5.1.6 The guessing number of a dumbbell graph

Definition 24. *Let C_m be the m-cycle undirected graph with vertex set $\{1, 2, \ldots, m\}$, C_n be the n-cycle undirected graph with vertex set $\{1', 2', \ldots, n'\}$, and P_k be the path graph of order k with vertex set $\{p_1, p_2, \ldots, p_k\}$. A dumbbell graph $Db(m, n, k)$ is a planar triangle-free undirected graph which is isomorphic to the graph*

$$K := (C_m \bigvee_{1 \equiv p_1} P_k) \bigvee_{p_k \equiv 1'} C_n.$$

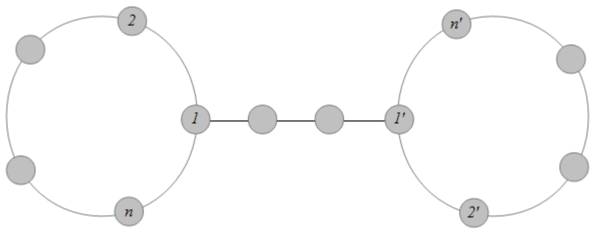


Figure 5.5: The Dumbbell graph.

Theorem 22. *Let G be a dumbbell graph $Db(m, n, k)$ of order $m + n + k - 2$. The guessing number of G is*

$$\mathrm{gn}(G) = \frac{m + n + k - 2}{2}.$$

Proof. The graph constraints give us the following equalities:

- $H(1|2, m, p_2) = 0$,
- $H(i|i-1, i+1 \mod m) = 0$ for $2 \leq i \leq m$,
- $H(1'|2', n', p_{k-1}) = 0$,
- $H((i)'|(i-1)', (i+1 \mod n)') = 0$ for $2 \leq i \leq n$,
- $H(p_i|p_{i-1}, p_{i+1}) = 0$ for $2 \leq i \leq k-1$ (Note that $1 \sim p_1$ and $p_k \sim 1'$.)

Since G is triangle-free, the fractional clique cover strategy provides a lower bound $\frac{m+n+k-2}{2} \leq \mathrm{gn}(G)$ by Lemma 6. Therefore, all we need to show is $\mathrm{gn}(G) \leq \mathrm{Sh}(G) = \frac{m+n+k-2}{2}$.

To compute $\mathrm{Sh}(G)$, we will break the computation into four cases: There are three cases:

1. m and n are even,
2. m is odd and n is even,
3. m and n are odd.

If m and n are even numbers, then G is a perfect graph and the result follows from our Corollary 4. The second case is established in Lemma 13. The third case is proved in Lemma 14. □

Lemma 13. *Let G be a dumbbell graph with parameters (m, n, k) where m is odd and n even. The guessing number of G is*

$$\text{gn}(G) = \frac{m+n+k-2}{2}.$$

Proof. There are two cases to prove:

1. k is even,
2. k is odd.

If k is even, we have:

$$\begin{aligned}
H(G) &= H(G|1,2,4,6,\dots,(m-1),p_3,p_5,\dots,p_{k-1},2',4',\dots,n') && (5.89)\\
&\quad + H(1,2,4,6,\dots,(m-1),p_3,p_5,\dots,p_{k-1},2',4',\dots,n') \quad \text{(c.e.)} && (5.90)\\
&= H(1,2,4,6,\dots,(m-1),p_3,p_5,\dots,p_{k-1},2',4',\dots,n') \quad \text{(g.c.)} && (5.91)\\
&\le H(1,2) + H(4,6,\dots,(m-1),p_3,p_5,\dots,p_{k-1},2',4',\dots,n') \quad \text{(s.)} && (5.92)\\
&\le H(1,2) + |\{4,6,\dots,(m-1),p_3,p_5,\dots,p_{k-1},2',4',\dots,n'\}| \quad \text{(n.)} && (5.93)\\
&= H(1,2) + \frac{n+m+k-5}{2}. && (5.94)
\end{aligned}$$

Similarly,

$$\begin{aligned}
H(G) &= H(G|1,2,3,5,\dots,m,p_2,p_4,\dots,p_{k-2},1',3',\dots,(n-1)') && (5.95)\\
&\quad + H(1,2,3,5,\dots,m,p_2,p_4,\dots,p_{k-2},1',3',\dots,(n-1)') \quad \text{(c.e.)} && (5.96)\\
&= H(1,2,3,5,\dots,m,p_2,p_4,\dots,p_{k-2},1',3',\dots,(n-1)') \quad \text{(g.c.)} && (5.97)\\
&\le H(1,2,3,m,p_2) && (5.98)\\
&\quad + H(5,7,\dots,(m-2),p_2,p_4,\dots,p_{k-2},1',3',\dots,(n-1)') \quad \text{(s.)} && (5.99)\\
&\le H(1,2,3,m,p_2) && (5.100)\\
&\quad + |\{5,7,\dots,(m-2),p_2,p_4,\dots,p_{k-2},1',3',\dots,(n-1)'\}| \quad \text{(n.)} && (5.101)\\
&= H(1,2,3,m,p_2) + \frac{m+n+k-9}{2}. && (5.102)
\end{aligned}$$

Adding 5.94 and 5.102 we get that:

$$\begin{aligned}
2H(G) &\leq H(1,2) + H(1,2,3,m,p_2) + m + n + k - 7 \\
&\leq H(1,2,3) + H(m,1,2,p_2) + m + n + k - 7 && \text{(s.)} \\
&= H(1,2,3|1,3) + H(1,3) \\
&+ H(m,1,2,p_2|m,2,p_2) + H(m,2,p_2) + m + n + k - 7 && \text{(c.e.)} \\
&= H(1,3) + H(m,2,p_2) + m + n + k - 7 && \text{(g.c.)} \\
&\leq |\{1,3\}| + |\{m,2,p_2\}| + m + n + k - 7 && \text{(n.)} \\
&= m + n + k - 2.
\end{aligned}$$

If k is odd, we have:

$$\begin{aligned}
H(G) &= H(G|1,2,4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',3',\ldots,(n-1)') && (5.103) \\
&+ H(1,2,4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',3',\ldots,(n-1)') && \text{(c.e.)} \quad (5.104) \\
&= H(1,2,4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',3',\ldots,(n-1)') && \text{(g.c.)} \quad (5.105) \\
&\leq H(1,2) + H(4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',3',\ldots,(n-1)') && \text{(s.)} \quad (5.106) \\
&\leq H(1,2) + |\{4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',3',\ldots,(n-1)'\}| && \text{(n.)} \quad (5.107) \\
&= H(1,2) + \frac{n+m+k-5}{2}. && (5.108)
\end{aligned}$$

Similarly,

$$\begin{aligned}
H(G) &= H(G|1,2,3,5,\ldots,m,p_2,p_4,\ldots,p_{k-1},2',4',\ldots,n') && (5.109) \\
&+ H(1,2,3,5,\ldots,m,p_2,p_4,\ldots,p_{k-1},2',4',\ldots,n') && \text{(c.e.)} \quad (5.110) \\
&= H(1,2,3,5,\ldots,m,p_2,p_4,\ldots,p_{k-1},2',4',\ldots,n') && \text{(g.c.)} \quad (5.111) \\
&\leq H(1,2,3,m,p_2) && (5.112) \\
&+ H(5,7,\ldots,(m-2),p_2,p_4,\ldots,p_{k-1},2',4',\ldots,n') && \text{(s.)} \quad (5.113) \\
&\leq H(1,2,3,m,p_2) && (5.114) \\
&+ |\{5,7,\ldots,(m-2),p_2,p_4,\ldots,p_{k-1},2',4',\ldots,n'\}| && \text{(n.)} \quad (5.115) \\
&= H(1,2,3,m,p_2) + \frac{m+n+k-9}{2}. && (5.116)
\end{aligned}$$

Adding 5.108 and 5.116 we get that:

$$\begin{aligned}
2H(G) &\leq H(1,2) + H(1,2,3,m,p_2) + m + n + k - 7 \\
&\leq H(1,2,3) + H(m,1,2,p_2) + m + n + k - 7 && \text{(s.)} \\
&= H(1,2,3|1,3) + H(1,3) + H(m,1,2,p_2|m,2,p_2) + H(m,2,p_2) \\
&+ m + n + k - 7 && \text{(c.e.)} \\
&= H(1,3) + H(m,2,p_2) + m + n + k - 7 && \text{(g.c.)} \\
&\leq |\{1,3\}| + |\{m,2,p_2\}| + m + n + k - 7 && \text{(n.)} \\
&= m + n + k - 2.
\end{aligned}$$

□

Lemma 14. *Let G be a dumbbell graph with parameters (m,n,k) where m and n are odd. The guessing number of G is*

$$\text{gn}(G) = \frac{m+n+k-2}{2}.$$

Proof. There are two cases to prove:

1. k is even,
2. k is odd.

If k is even, we have:

$$\begin{aligned}
H(G) &= H(G|1,2,4,6,\ldots,(m-1),p_2,p_3,\ldots,p_{k-1},1',2',4',6',\ldots,(n-1)') && (5.117) \\
&+ H(1,2,4,6,\ldots,(m-1),p_2,p_3,\ldots,p_{k-1},1',2',4',6',\ldots,(n-1)') && \text{(c.e.)} \; (5.118) \\
&= H(1,2,4,6,\ldots,(m-1),p_2,p_3,\ldots,p_{k-1},1',2',4',6',\ldots,(n-1)') && \text{(g.c.)} \; (5.119) \\
&\leq H(1,2,p_2,p_3,\ldots,p_{k-1},1',2') && (5.120) \\
&+ H(4,6,\ldots,(m-1),4',6',\ldots,(n-1)') && \text{(s.)} \; (5.121) \\
&\leq H(1,2,p_2,p_3,\ldots,p_{k-1},1',2') && (5.122) \\
&+ |\{4,6,\ldots,(m-1),4',6',\ldots,(n-1)'\}| && \text{(n.)} \; (5.123) \\
&= H(1,2) + \frac{n+m-6}{2}. && (5.124)
\end{aligned}$$

Similarly,

$$\begin{aligned}
H(G) &= H(G|1,2,3,5,\ldots,m,p_2,p_3,\ldots,p_{k-1},1',2',3',5',\ldots,n') && (5.125)\\
&+ H(1,2,3,5,\ldots,m,p_2,p_3,\ldots,p_{k-1},1',2',3',5',\ldots,n') \quad \text{(c.e.)} && (5.126)\\
&= H(1,2,3,5,\ldots,m,p_2,p_3,\ldots,p_{k-1},1',2',3',5',\ldots,n') \quad \text{(g.c.)} && (5.127)\\
&\leq H(1,2,3,m,p_2,p_3,\ldots,p_{k-1},1',2',3',n') && (5.128)\\
&+ H(5,7,\ldots,(m-2),5',7',\ldots,(n-2)') \quad \text{(s.)} && (5.129)\\
&\leq H(1,2,3,m,p_2,p_3,\ldots,p_{k-1},1',2',3',n') && (5.130)\\
&+ |\{5,7,\ldots,(m-2),5',7',\ldots,(n-2)'\}| \quad \text{(n.)} && (5.131)\\
&= H(1,2,3,m,p_2,p_3,\ldots,p_{k-1},1',2',3',n') + \frac{m+n-10}{2}. && (5.132)
\end{aligned}$$

Adding 5.124 and 5.132 we get that:

$$\begin{aligned}
2H(G) &\leq H(1,2,p_2,p_3,\ldots,p_{k-1},1',2')\\
&+ H(1,2,3,m,p_2,p_3,\ldots,p_{k-1},1',2',3',n')\\
&+ m+n-8\\
&\leq H(1,2,3,p_2,p_3,\ldots,p_{k-1},1',2',n')\\
&+ H(1,2,m,p_2,p_3,\ldots,p_{k-1},1',2',3')\\
&+ m+n-8 \quad \text{(s.)}\\
&= H(1,2,3,p_2,p_3,\ldots,p_{k-1},1',2',n'|1,3,p_3,p_5,\ldots,p_{k-1},2',n')\\
&+ H(1,3,p_3,p_5,\ldots,p_{k-1},2',n')\\
&+ H(1,2,m,p_2,p_3,\ldots,p_{k-1},1',2',3'|1,2,m,p_2,p_4,\ldots,p_{k-2},1',3')\\
&+ H(1,2,m,p_2,p_4,\ldots,p_{k-2},1',3')\\
&+ m+n-8 \quad \text{(c.e.)}\\
&= H(1,3,p_3,p_5,\ldots,p_{k-1},2',n')\\
&+ H(1,2,m,p_2,p_4,\ldots,p_{k-2},1',3') + m+n-8 \quad \text{(g.c.)}\\
&\leq |\{1,3,p_3,p_5,\ldots,p_{k-1},2',n'\}|\\
&+ |\{1,2,m,p_2,p_4,\ldots,p_{k-2},1',3'\}| + m+n-8 \quad \text{(n.)}\\
&= m+n+k-2.
\end{aligned}$$

If k is odd, we have:

$$H(G) = H(G|1,2,4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',2',4',6',\ldots,(n-1)') \tag{5.133}$$
$$+ H(1,2,4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',2',4',6',\ldots,(n-1)') \quad \text{(c.e.)} \tag{5.134}$$
$$= H(1,2,4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},1',2',4',6',\ldots,(n-1)') \quad \text{(g.c.)} \tag{5.135}$$
$$\leq H(1,2,1',2') \tag{5.136}$$
$$+ H(4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},4',6',\ldots,(n-1)') \quad \text{(s.)} \tag{5.137}$$
$$\leq H(1,2,1',2') \tag{5.138}$$
$$+ |\{4,6,\ldots,(m-1),p_3,p_5,\ldots,p_{k-2},4',6',\ldots,(n-1)'\}| \quad \text{(n.)} \tag{5.139}$$
$$= H(1,2,1',2') + \frac{n+m+k-9}{2}. \tag{5.140}$$

Similarly,

$$H(G) = H(G|1,2,3,5,\ldots,m,p_2,p_4,\ldots,p_{k-1},1',2',3',5',\ldots,n') \tag{5.141}$$
$$+ H(1,2,3,5,\ldots,m,p_2,p_4,\ldots,p_{k-1},1',2',3',5',\ldots,n') \quad \text{(c.e.)} \tag{5.142}$$
$$= H(1,2,3,5,\ldots,m,p_2,p_4,\ldots,p_{k-1},1',2',3',5',\ldots,n') \quad \text{(g.c.)} \tag{5.143}$$
$$\leq H(1,2,3,m,1',2',3',n',p_2,p_{k-1}) \tag{5.144}$$
$$+ H(5,7,\ldots,(m-2),p_4,p_6,\ldots,p_{k-3},5',7',\ldots,(n-2)') \quad \text{(s.)} \tag{5.145}$$
$$\leq H(1,2,3,m,1',2',3',n',p_2,p_{k-1}) \tag{5.146}$$
$$+ |\{5,7,\ldots,(m-2),p_4,p_6,\ldots,p_{k-3},5',7',\ldots,(n-2)'\}| \quad \text{(n.)} \tag{5.147}$$
$$= H(1,2,3,m,1',2',3',n',p_2,p_{k-1}) + \frac{m+n+k-15}{2}. \tag{5.148}$$

Adding 5.140 and 5.148 we get that:

$$\begin{aligned}
2H(G) &\leq H(1,2,1',2') + H(1,2,3,m,1',2',3',n',p_2,p_{k-1}) \\
&\quad + m + n + k - 12 \\
&\leq H(1,2,3,1',2',3') + H(m,1,2,p_2,1',2',n',p_{k-1}) \\
&\quad + m + n + k - 12 && \text{(s.)} \\
&= H(1,2,3,1',2',3'|1,3,1',3') + H(1,3,1',3') \\
&\quad + H(m,1,2,p_2,1',2',n',p_{k-1}|m,2,p_2,2',n',p_{k-1}) + H(m,2,p_2,2',n',p_{k-1}) \\
&\quad + m + n + k - 12 && \text{(c.e.)} \\
&= H(1,3,1',3') + H(m,2,p_2,2',n',p_{k-1}) + m + n + k - 12 && \text{(g.c.)} \\
&\leq |\{1,3,1',3'\}| + |\{m,2,p_2,2',n',p_{k-1}\}| + m + n + k - 12 && \text{(n.)} \\
&= m + n + k - 2.
\end{aligned}$$

□

5.1.7 The guessing number of a flower graph

Definition 25. *Let T be a tree with k-leaves $\{1_T, 2_T, \ldots, k_T\}$. Let $\mathcal{C} = \{C_{n_1}, \ldots, C_{n_k}\}$ be a collection of cycles of length n_i's. We denote the vertices of C_{n_i} as $\{1^i, 2^i, \ldots, (n_i)^i\}$. A flower graph $T \bigvee \mathcal{C}$ is a planar triangle-free undirected graph which is isomorphic to the graph*

$$K := \left(\ldots \left((T \bigvee_{1_T \equiv 1^1} C_{n_1}) \bigvee_{2_T \equiv 1^2} C_{n_2} \right) \ldots \right) \bigvee_{k_T \equiv 1^k} C_{n_k}$$

.

Theorem 23. *Let G be a planar triangle-free graph which is isomorphic to a flower graph $T \bigvee \mathcal{C}$, then the guessing number of G is*

$$\mathrm{gn}(G) = \frac{|V(G)|}{2}.$$

A key observation in proving Theorem 23 is that we can combine multiple copies of the computation processes demonstrated in Lemma 13 and Lemma 14. These calculations can be merged thanks to the following Lemma 15.

Let T be a tree with k-leaves $\{1, 2, \ldots k\}$. Let us fix an order of all branches in T corresponding to the order of its leaves.

We have the following facts about distances between leaves in T:

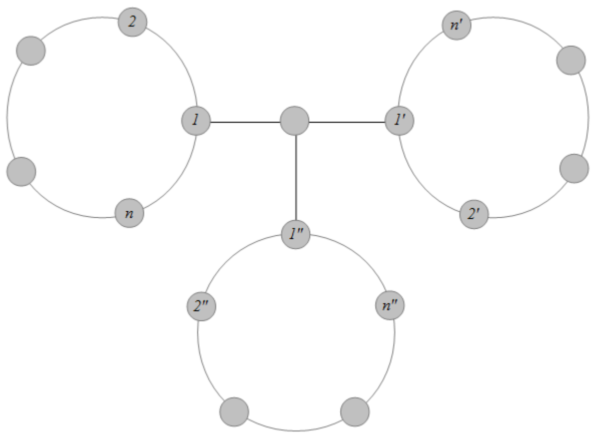


Figure 5.6: An instance of a Flower graph.

Observation 12. Let $1 \le i < j \le k$ be leaves of T with lowest common ancestor $l(i,j)$. Let $T_{l(i,j)}$ be the subtree of T with $l(i,j)$ as its root, then the shortest path between i and j in T is is the union of the shortest path from i to $l(i,j)$ and the shortest path between $l(i,j)$ and j in $T_{l(i,j)}$.

Observation 13. Let T be as above with leaves $\{1,2,\dots k\}$. Let $1 \le i < j < l \le k$ be leaves of T with pairwise least common ancestors are $l(i,j)$, $l(j,k)$, $l(i,k)$. We have:

- $l(j,k) \equiv l(i,k)$,
- The shortest path between i and k must contain $l(i,j)$.

Lemma 15. *Let T be a tree having k-leaves $\{1,2,\dots k\}$. We fix an order of all branches in T corresponding to the order of its leaves. Let $1 \le i < j < l \le k$ be leaves of T with pairwise distances are $d(i,j)$, $d(j,k)$, and $d(i,k)$. We have the following relation:*

$$d(i,k) \equiv d(i,j) + d(j,k) - 1 \mod 2.$$

Moreover, the set of all vertices $u \in V(T)$ such that $d(1,u)$ is odd is an vertex cover of T.

Proof. From the previous observations, we have that both $T_{l(i,j)}$ and $T_{l(j,k)}$ are subtrees of $T_{l(i,k)}$. Moreover,

$$
\begin{aligned}
d(i,j) &= d(i,l(i,j)) + d(l(i,j),j),\\
d(j,k) &= d(j,l(j,k)) + d(l(j,k),k) = d(j,l(i,j)) + d(l(i,j),l(j,k)) + d(l(j,k),k),\\
d(i,k) &= d(i,l(i,k)) + d(l(i,k),k) = d(i,j) + d(j,k) - 2d(l(i,j),j) + 1.
\end{aligned}
$$

□

Proof of Theorem 23. By definition, G is the wedge sum of a tree T having k-leaves with k different cycles $\{C_{n_i}\}_{i=1}^k$, each cycle is attached to a leaf of T. In order to clarify the computation, we will make use of the following notation: for each leaf $i_T \in V(T)$, we write p_T^i for the parent node of i_T.

By the graph constrain, we have the following equalities:

- $H(v^i|(v-1)^i, (v+1 \mod n_i)^i) = 0$ for any $i \in \{1, \dots, k\}$ and $v \in \{2, \dots, n_i\}$.
- $H(1^i|2^i, n_i^i, p_T^i) = 0$ for any $i \in \{1, \dots, k\}$.
- If v is a non-leaf of T, then $H(v|w, u_1, \dots, u_m) = 0$, where w is the parent node of v in T, and $\{u_1, \dots, u_m\}$ are children of v in T.

We will only need to prove the statement when G has at least one odd cycle. This is because if all the cycles C_{n_i} are even, then G is perfect.

As we can always reorder the leaves of T, without loss of generality, we will assume that C_{n_1} is an odd cycle.

We choose the set S as follows:

- If C_{n_j} is an odd cycle then we select $\{1^j, 2^j, 4^j, 6^j, \dots, (n_j-1)^j\}$.
- If C_{n_j} is an even cycle then we select $\{1^j, 2^j, \dots, n_j^j\}$.
- We select all v belonging to the T.

Since S is an vertex cover of G, we have:

$$\begin{aligned}
H(G) &= H(G|S) + H(S) && \text{(c.e.)} \quad (5.149)\\
&= H(S) && \text{(g.c.)} \quad (5.150)\\
&\leq H(S \backslash P) + H(P) && \text{(s.)} \quad (5.151)\\
&\leq H(S \backslash P) + |P| && \text{(n.)} \quad (5.152)
\end{aligned}$$

where $P = \{(4^j, 6^j, \dots, (n_j-1)^j)_j : \text{ if } n_j \text{ is odd}\}$.

We select the set L as follows:

- If C_{n_j} is an odd cycle, then we select $\{1^j, 2^j, 3^j, 5^j, \dots, n_j^j\}$.
- If C_{n_j} is an even cycle then we select $\{1^j, 2^j, \dots, n_j^j\}$.
- We select v if v belongs to the T.

Since L is an vertex cover of G, we have:

$$\begin{aligned} H(G) &= H(G|L) + H(L) && \text{(c.e.)} \quad (5.153) \\ &= H(L) && \text{(g.c.)} \quad (5.154) \\ &\leq H(L\backslash Q) + H(Q) && \text{(s.)} \quad (5.155) \\ &\leq H(L\backslash Q) + |Q| && \text{(n.)} \quad (5.156) \end{aligned}$$

where $Q = \{(5^j, 7^j, \ldots, (n_j - 2)^j)_j : \text{ if } n_j \text{ is odd}\}$.

Note that $(S\backslash P) \subset (L\backslash Q)$.

Adding 5.152 and 5.156 we have:

$$\begin{aligned} 2H(G) &\leq H(L\backslash Q) + H(S\backslash P) + |Q| + |P| && (5.157) \\ &\leq H(X) + H(Y) + |P| + |Q| \quad \text{(s.)} && (5.158) \end{aligned}$$

where X contains the following elements:

- $\{1^1, 2^1, (n_1)^1\}$,
- $\{1^j, 2^j, (n_j)^j\}$ if n_j is odd and $d(1^1, 1^j)$ is even,
- $\{1^j, 2^j, 3^j\}$ if n_j is odd and $d(1^1, 1^j)$ is odd,
- $\{1^j, 2^j, \ldots, n_j^j\}$ if n_j is an even,
- v if v belongs to the T.

and Y consists of the following vertices:

- $\{1^1, 2^1, 3^1\}$,
- $\{1^j, 2^j, 3^j\}$ if n_j is odd and $d(1^1, 1^j)$ is odd,
- $\{1^j, 2^j, (n_j)^j\}$ if n_j is odd and $d(1^1, 1^j)$ is even,
- $\{1^j, 2^j, \ldots, n_j^j\}$ if n_j is an even,
- v if v belongs to the T.

It is clear that:

$$X \cup Y = L\backslash Q,$$
$$X \cap Y = S\backslash P.$$

All we need is to bound the value of $H(X)$ and $H(Y)$. For X, we have:

- $H(1^1|2^1,(n_1)^1,p_T^i) = 0$.
- If C_{n_j} is an odd cycle and $d(1^1,1^j)$ is even, we have $H(1^j|2^j,(n_1)^j,p_T^j) = 0$.
- If C_{n_j} is an odd cycle and $d(1^1,1^j)$ is odd, we have

$$H(2^j|1^j,3^j) = 0.$$

- If v_T is a node in T and $d(1^1,v_T)$ is even, we have

$$H(v_T|w_T,u_{1,T},\ldots,u_{m,T}) = 0$$

  where w_T is parent node of v_T, and $u_{1,T},\ldots,u_{m,T}$ are children of v_T all are of odd distance from 1^1 (follow Lemma 15).
- If C_{n_j} is an even cycle and $d(1^1,1^j)$ is odd, we have

$$H(2^j,4^j,\ldots n_j^j|1^j,3^j,\ldots(n_j-1)^j) = 0.$$

- If C_{n_j} is an even cycle and $d(1^1,1^j)$ is even, we have

$$H(1^j,3^j,\ldots(n_j-1)^j|p_T^j,2^j,4^j,\ldots n_j^j) = 0.$$

Therefore,

$$H(X) = H(X|Z) + H(Z) = H(Z) \leq |Z|,$$

where Z is the set consisting of:

- 2^1, $(n_1)^1$,
- 2^j, $(n_1)^j$ if C_{n_j} is an odd cycle and $d(1^1,1^j)$ is even,
- 1^j, 3^j if C_{n_j} is an odd cycle and $d(1^1,1^j)$ is odd,
- v_T if $d(1^1,v_T)$ is odd,
- 1^j, 3^j, …, $(n_j-1)^j$ if C_{n_j} is an even cycle and $d(1^1,1^j)$ is odd,
- 2^j, 4^j, …, n_j^j if C_{n_j} is an even cycle and $d(1^1,1^j)$ is even.

For Y, we have:

- $H(2^1|1^1,3^1) = 0$,
- If C_{n_j} is an odd cycle and $d(1^1,1^j)$ is even, we have

$$H(2^j|1^j,3^j) = 0.$$

- If C_{n_j} is an odd cycle and $d(1^1, 1^j)$ is odd, we have

$$H(1^j | 2^j, (n_1)^j, p_T^j) = 0.$$

- If v_T is a node in T and $d(1^1, v_T)$ is odd, we have

$$H(v_T | w_T, u_{1,T}, \ldots, u_{m,T}) = 0$$

where w_T is parent node of v_T, and $u_{1,T}, \ldots, u_{m,T}$ are children of v_T all are of even distance from 1^1 (follow Lemma 15).

- If C_{n_j} is an even cycle and $d(1^1, 1^j)$ is odd, we have

$$H(1^j, 3^j, \ldots (n_j - 1)^j | p_T^j, 2^j, 4^j, \ldots n_j^j).$$

- If C_{n_j} is an even cycle and $d(1^1, 1^j)$ is even, we have

$$H(2^j, 4^j, \ldots n_j^j | 1^j, 3^j, \ldots (n_j - 1)^j) = 0.$$

Therefore,

$$H(Y) = H(Y|W) + H(W) = H(W) \leq |W|,$$

where W is the set consisting of:

- 1^1, 3^1,
- 1^j, 3^j if C_{n_j} is an odd cycle and $d(1^1, 1^j)$ is even,
- 2^j, $(n_1)^j$ if C_{n_j} is an odd cycle and $d(1^1, 1^j)$ is odd,
- v_T if $d(1^1, v_T)$ is even,
- 2^j, 4^j, …, n_j^j if C_{n_j} is an even cycle and $d(1^1, 1^j)$ is odd,
- 1^j, 3^j, …, $(n_j - 1)^j$ if C_{n_j} is an even cycle and $d(1^1, 1^j)$ is even.

Combine all previous information, we have

$$\begin{aligned} 2H(G) &\leq H(X) + H(Y) + |P| + |Q| \\ &= H(Z) + H(W) + |P| + |Q| \\ &\leq |Z| + |W| + |P| + |Q|. \end{aligned}$$

But since $Z \cup W \cup P \cup Q = V(G)$ and Z, W, P, Q are pairwise disjoint, we have:

$$2H(G) \leq |V(G)|.$$

This completes our proof. □

5.1.8 The guessing number of certain regular graphs

The following theorems are due to Blasiak et. al. [12]

Theorem 24. *[12] For any $n \geq 4$, a cyclic Cayley graph of $\mathbb{Z}/n\mathbb{Z}$ with generators $S \subseteq \{1, 2, \ldots, \lfloor n/2 \rfloor\}$ such that it is also a 3-regular graph has broadcast rate $\beta(G) = n/2$.*

Theorem 25. *[12] For any integers $n \geq 4$ and $k \leq \frac{n-1}{2}$, the Cayley graph of $\mathbb{Z}/n\mathbb{Z}$ with generators $\{\pm 1, \ldots, \pm k\}$ has broadcast rate $\beta = n/(k+1)$.*

We know that $\mathrm{gn}(G) = |V(G)| - \beta(G)$, hence we have:

Corollary 5. *Let G be a graph described in Theorem 24. G has guessing number* $\mathrm{gn}(G) = n/2$.

Corollary 6. *Let G be a graph described in Theorem 25. G has guessing number* $\mathrm{gn}(G) = \frac{nk}{k+1}$.

5.2 Existence of non-Shannon bounds for guessing numbers of undirected graphs

Prior to this work, a common method to come up with network information flow/index coding instances with gaps between bounds given by information inequalities was to adopt the matroidal construction introduced by Dougherty et al. [36]. The construction generally goes as follows:

Let $\mathcal{M} = (E, r)$ be the Vámos matroid[1]. This is an eight-element rank-four matroid defined on a ground set $E = \{a, b, c, d, w, x, y, z\}$ with dependent sets[2] are all the subsets which contain at least five elements in addition to the four-element sets $\{b, c, x, y\}$, $\{a, c, w, y\}$, $\{a, b, w, x\}$, $\{c, d, y, z\}$, and $\{b, d, x, z\}$. This matroid is known to be non-representable [78]. We can construct an index coding instance $G_{\mathcal{M}}$ as follows:

The message set of $G_{\mathcal{M}}$ has 8 messages. Each message corresponds to one of the element of the ground set $E(M)$. If a is an element of $E(M)$, we will be freely speaking about the message a instead of the message corresponding to the element a. A receiver

[1] A matroid is a pair (E, r) where E is a finite set called ground set and $r : 2^E \to \mathbb{N}$ is called a rank function. A rank function must satisfy a list of conditions, e.g. normality, monotonicity, submodularity (see [78]).

[2] A subset $S \subseteq E$ is a dependent set if $r(S) < |S|$

$j_{C,e}$ for each message $e \in C$ and circuit $C \subseteq E$[3]. Receiver $j_{C,e}$ wants to obtain e and it has side information $C \backslash e$. Following some intricate calculations of entropy, it can be shown that $b(G_{\mathcal{M}}) = 4$ and $\beta(G_{\mathcal{M}}) \geq 45/11$ where $b(G_{\mathcal{M}})$ is the bound derived from the Shannon inequalities while $\beta(G_{\mathcal{M}})$ is the optimal broadcasting rate. This construction, by far, is the simplest example that we can find in the literature yet the obtained graph $G_{\mathcal{M}}$ is somewhat complicated.

It is noted that the index coding instance obtained here is a hypergraph, but not an undirected graph. In fact, it is generally unknown whether we can construct an index coding instance over an undirected graph having non-Shannon bounds sharper than Shannon one using the matroidal construction.

Our new result demonstrates that a graph exists where non-Shannon inequalities provide a better approximation of the guessing number compared to using the Shannon information inequalities alone. In fact, we show that there are gaps between the bounds provided by the Zhang-Yeung inequalities and its alternative provided by the Dougherty-Freiling-Zeger inequalities.

One important point to note in our result is that the example is an undirected graph of small order. Therefore, it is possible to compute and check the calculation by computer. In fact, the process of deriving these bounds is achieved by computer, and data files can be obtained upon request.

We have

Theorem 26.

- $\mathrm{Sh}(R^-) = 114/17 = 6.705\ldots$
- $\mathrm{ZY}(R^-) = 1212/181 = 6.696\ldots$
- $\mathrm{DFZ}(R^-) = 59767/8929 = 6.693\ldots$
- $\mathrm{Ingl}(R^-) = 20/3 = 6.666\ldots$

From Lemma 7 and Theorem 26 we know that

$$20/3 \leq \mathrm{gn}(R^-) \leq 59767/8929,$$

and although we could not determine the asymptotic guessing number exactly it does show that it does not equal the Shannon bound, disproving Conjecture 2. Given that the Shannon bound is not sharp we might be tempted to conjecture that the asymptotic guessing number is the same as the Zhang-Yeung bound, but Theorem

[3]A subset $C \subseteq E$ is called a circuit if it is a minimal dependent set.

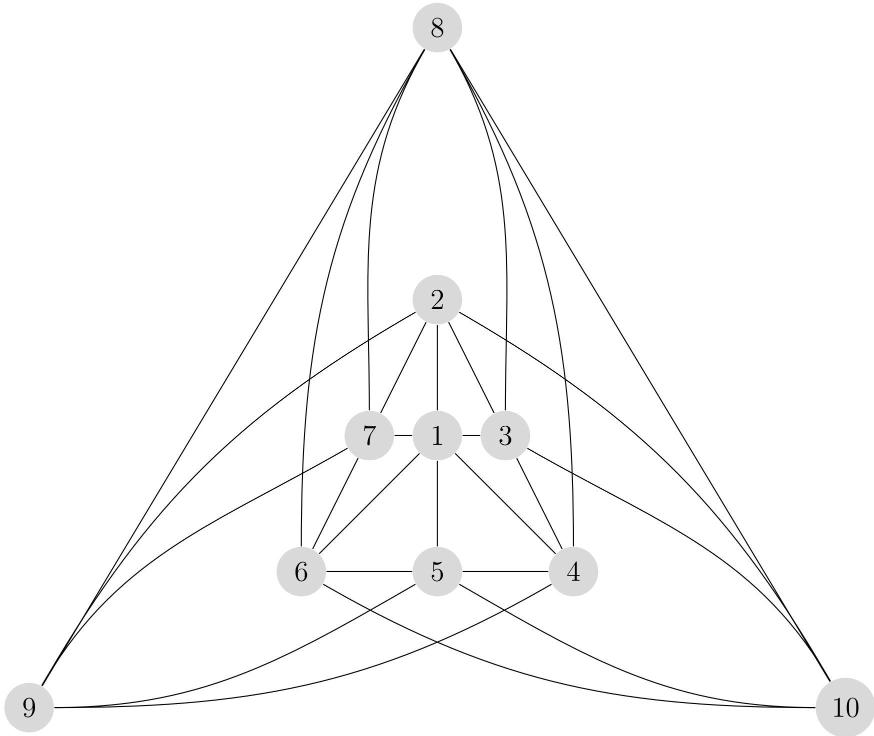


Figure 5.7: The undirected graph R^-.

11 also shows this to be false. Interestingly the Ingleton bound does match the lower bound, showing that if we restrict ourselves to only considering linear strategies on blowups we can do no better than the fractional clique cover strategy.

It remains an open question as to whether a non-linear strategy on R^- can do better than 20/3 or whether by considering the right set of entropy inequalities we can push the upper bound down to 20/3.

Proof of Theorem 11. Calculating the upper bounds involves solving rather large linear programs. Consequently, the proofs are too long to reproduce here and it is unfeasible for them to be checked by humans. Data files verifying our claims can be provided upon request. We stress that although the results were verified using a computer, no floating point data types were used during the verification. Consequently no rounding errors could occur in the calculations, making the results completely rigorous. □

We finish this section by considering a problem motivated by the reversibility of networks in network coding. Given a digraph G, let Reverse(G) be the digraph formed from G by reversing all the edges, i.e. $uv \in E(G)$ if and only if $vu \in E(\text{Reverse}(G))$.

Problem 14. Does a digraph G exist, such that $\mathrm{gn}(G) \neq \mathrm{gn}(\mathrm{Reverse}(G))$?

We were not able to solve this problem. We did, however, find a graph R^S for which the Shannon bound of R^S and the Shannon bound of Reverse(R^S) did not match. R^S is simply the digraph formed by making vertex 1 in R a Superman vertex. In other words, we add three directed edges to R: the edge going from 1 to 8, from 1 to 9, and from 1 to 10. Consequently Reverse(R^S) is the graph formed by making vertex 1 in R a Luthor vertex. As such, we will refer to it as R^L.

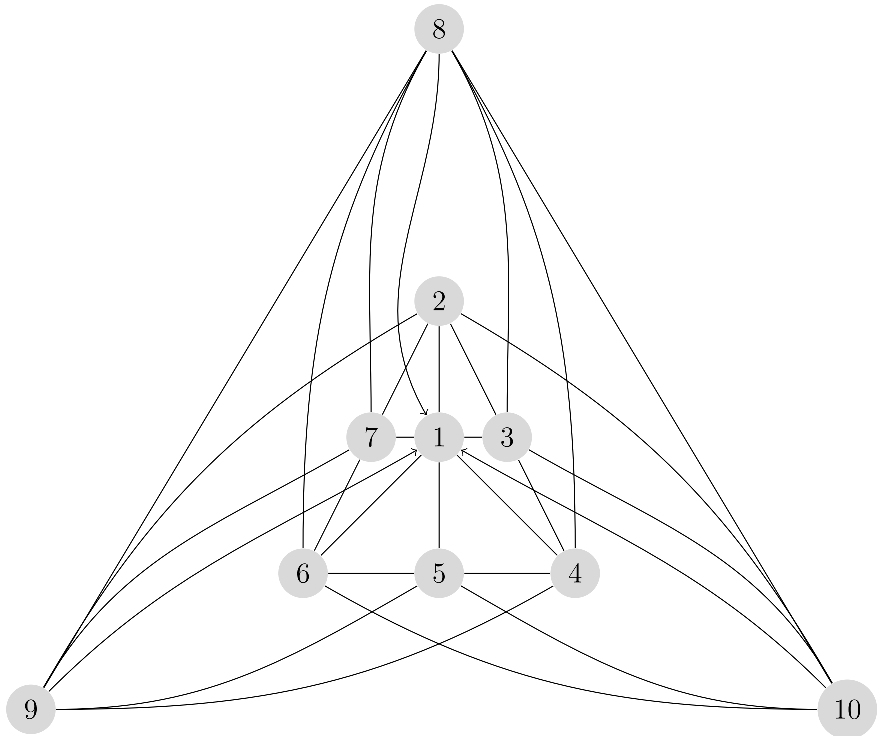


Figure 5.8: The digraph R^S.

Theorem 27. *We have* $\mathrm{Sh}(R^S) = 27/4 = 6.75$.

Proof. The proofs are given in data files, which can be made available upon request. □

Theorem 28. *For* R^L *we have the following bounds:*

- $\mathrm{Sh}(R^L) = 34/5 = 6.8$.
- $\mathrm{ZY}(R^L) = 61/9 = 6.777\ldots$
- $\mathrm{DFZ}(R^L) = 359/53 = 6.773\ldots$
- $\mathrm{Ingl}(R^L) = 27/4 = 6.75$.

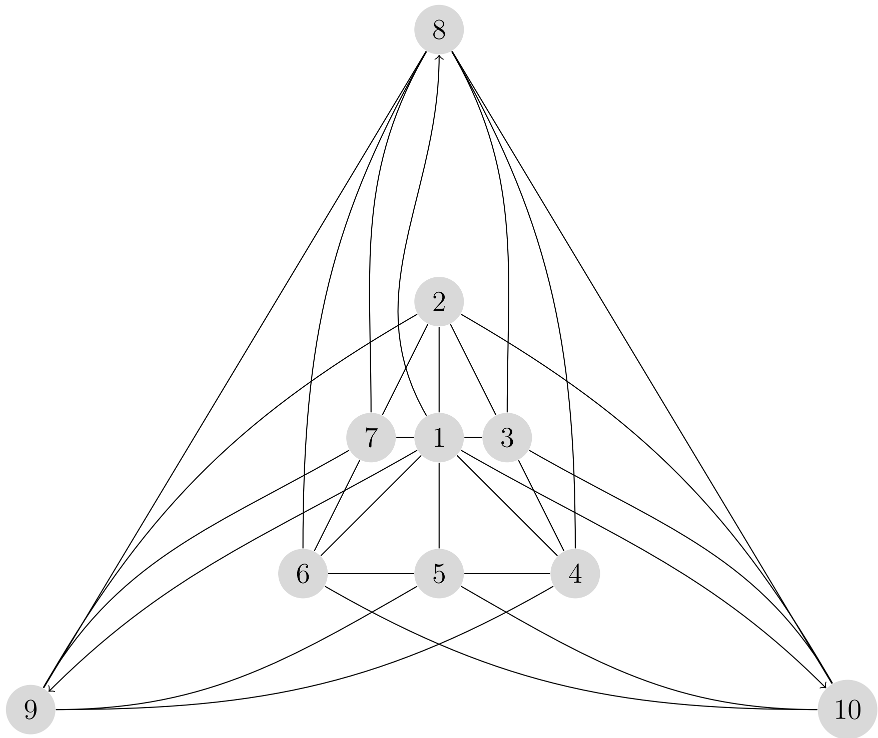


Figure 5.9: The digraph R^L.

Proof. The proofs are given in data files, which can be made available upon request. □

From the strategy on R we know that $\text{gn}(R^S) \geq 27/4$ and $\text{gn}(R^L) \geq 27/4$. Hence we have $\text{gn}(R^S) = 27/4$. We do not, however, know the precise value of $\text{gn}(R^L)$ so it is possible that the asymptotic guessing numbers of R^S and R^L do not match.

Chapter 6

Guessing Games with Noises

6.1 Motivation and discussion of related work

The beauty of network coding comes from two aspects: modifiable data packages and network topology. By choosing appropriate coding functions with respect to network topology, we may gain a huge throughput in comparison to using the traditional routing method. However, when a message is distorted, we face the problem of error propagation, which may cause all terminals to decipher wrong data.

We can battle against the problem of error-pruned communication channels by applying the classical theory of error correcting codes, which adds redundancy to the transmission in the time domain. With this approach, we can guarantee eliminating some noise effects on communication channels. However, this mechanism leaves out the information about network topology, which essentially is the source of all properties of network coding. In order to gain the full benefits of network coding, we wish to encode the topological properties of a given network within our coding functions as a method to improve the noisy coding capacity.

The first successful combination of linear network code and error correcting code was accomplished by Cai and Yeung [21, 96, 22] for multicast networks. We know that linear code is sufficient for multicast problems. Hence, in the case when there is a corruption during the transmission, the reception at the sinks will only be affected by a linear transformation of an error vector. Therefore, we can define the network Hamming weight as a measure that depends on linear transformations of the error vector. In particular, we would expect that if an error occurred but it was eliminated at the terminals due to coding functions, then the weight of that vector would be

0. Moreover, if the weight of the difference of two error vectors is 0, then these two error vectors must have the same weight. Once these expected properties of network Hamming weight are identified, one can define the weights of an error vector, a received vector, and a message vector. The coding distance and coding bounds are naturally derived once the concept of coding weight is settled. Cai and Yeung extended some useful bounds, e.g. Hamming bound, Singleton bound, and Gilbert-Varshamov bound to network coding in their influential papers on network error correction [21, 96, 22]. Based on this framework, various algorithmic constructions of network error correcting codes have been proposed [54, 92, 6, 74, 47, 53, 62, 17, 43, 46, 86, 100, 63, 101].

In the case of general network information flow problems, in contrast, a convinced framework for network error correction remains elusive. The obstacle exists due to the enigmatic interaction between coding function and noises in the network. In this chapter, we initialize studies of interaction between coding functions and noises restricted to our setting of guessing games. We proposed a definition of noisy guessing number which is a generalized version of the noiseless guessing number introduced by Riis [82]. First few properties of this quantity together with show cases on undirected graphs of small order are also established.

6.2 Definitions and some basic bounds

Definition 26. *Let G be a directed graph and $0 \leq \epsilon \leq 1/2$ be a real number. A* guessing game with noises $(G, 2, \epsilon)$ *is a game played on a digraph G and the alphabet $\{0, 1\}$. There are $|V(G)|$ players working as a team. Each player corresponds to one of the vertices of the digraph. Each player v is assigned an integer x_v from $\{0, 1\}$ uniformly and independently at random. Each player will be given a list of the players in its in-neighbourhood with their corresponding values. However, for each u in v's in-neighbourhood, there is a probability ϵ that instead of giving x_u to v, the organizer will show v the flipped value $(1 - x_u) \mod 2$. This process is done independently for each player. Using just the information about G and ϵ, each player must guess his own value. If all players guess correctly, they will all win, but if just one player guesses incorrectly they will all lose.*

Note that we recover the original guessing game $(G, 2)$ by letting $\epsilon = 0$. Just like the definition of a guessing strategy for a normal guessing game, we have definition for guessing strategy in this noisy guessing game as follows:

Definition 27. *Given a guessing game with noises $(G, 2, \epsilon)$, for $v \in V(G)$ a* strategy *for player v is formally a function $f_v : \{0,1\}^{|\Gamma^-(v)|} \to \{0,1\}$ which maps the values of the in-neighbours of v to an elements of $\{0,1\}$, which will be the guess of v. A* strategy *$\mathcal{F}$ for a guessing game is a sequence of such functions $(f_v)_{v\in V(G)}$ where f_v is a strategy for player v.*

We denote by $\mathrm{Win}(G, 2, \epsilon, \mathcal{F})$ *the event that all the players guess correctly when playing $(G, 2, \epsilon)$ with strategy $\mathcal{F}$.*

When $\epsilon = 0$, we can visualize the behaviour of a guessing strategy in terms of mappings from the space of all configurations $\{0,1\}^{|V(G)|}$ into itself, i.e. $\mathcal{F} : \{0,1\}^{|V(G)|} \to \{0,1\}^{|V(G)|}$. A correct guess on a given outcome of random numbers $\mathbf{x} = (x_v)_{v\in V(G)} \in \{0,1\}^{|V(G)|}$ by using strategy $\mathcal{F}$ is equivalent to a solution of the equation

$$\mathcal{F}(\mathbf{x}) = \mathbf{x}. \tag{6.1}$$

Thus the problem of maximizing the probability of winning can be interpreted as finding a map $\mathcal{F} = (f_v)_{v\in V(G)} : \{0,1\}^{|V(G)|} \to \{0,1\}^{|V(G)|}$ with as many fixed points as possible, where f_v is strategy for player v.

However, when $\epsilon > 0$, the picture is much more complicated. Let us firstly introduce some new notations to distinguish between an assigned value, an observed value, and a guess value of each player v. For a player $v \in V(G)$, we denote its assigned value as x_v^a which is either 0 or 1, and we write $\mathbf{x}^a$ for $(x_v^a)_{v\in V(G)}$. Note that $\mathbf{x}^a \in \{0,1\}^{|V(G)|}$. If u is an in-neighbourhood of v, we denote the value assigned to u that v observes as ${}_v x_u^o$. We write $\mathbf{x}_{\Gamma v}^o = ({}_v x_u^o)_{u\in\Gamma_v^-}$ for the observed value of player v. Note that if u and w belong to the out-neighbourhoods of v, ${}_u x_v^o$ might not equal ${}_w x_v^o$. We write $\mathbf{x}^o = (\mathbf{x}_{\Gamma v}^o)_{v\in V(G)}$ for the observed value by all players in one instance of the guessing game. The tuple $\mathbf{x}^o$ takes value in $\bigoplus_{v\in V(G)}(\{0,1\}^{\Gamma_v^-})$. We denote the guess value made by player v as x_v^g, and we write $\mathbf{x}^g$ for $(x_v^g)_{v\in V(G)}$. Note that $\mathbf{x}^g$ again is a vector in $\{0,1\}^{|V(G)|}$. Now we can rewrite the fixed points equation for the guessing game with noise as follows:

$$\mathbf{x}^g := \mathcal{F}(\mathbf{x}^o) = \mathbf{x}^a. \tag{6.2}$$

For the special case when every player observes correctly the assigned value, i.e. ${}_u x_v^o = x_v^a$ for all $u, v \in V(G)$, we write $\mathbf{x}^{a,o}$ instead of $\mathbf{x}^o$, and we call $\mathbf{x}^a$ a fixed point for strategy $\mathcal{F}$ if

$$\mathcal{F}(\mathbf{x}^{a,o}) = \mathbf{x}^a.$$

Equation 6.2 shows us two new phenomena in guessing a correct answer. The first aspect is the ability of auto-correcting errors of a guessing strategy $\mathcal{F}$. In particular, the situation where $\mathbf{x}^g = \mathbf{x}^a$ given that $\mathbf{x}^a$ is a fixed point of $\mathcal{F}$ is now a random event appearing if either no error occurs, i.e. $\mathbf{x}^o = \mathbf{x}^{a,o}$, or $\mathcal{F}(\mathbf{x}^o) = \mathbf{x}^a$ for some $\mathbf{x}^o \neq \mathbf{x}^{a,o}$, in other words, $\mathcal{F}$ can auto-correct some errors caused by noises.

The second aspect is the noisy phenomenon appeared when $\mathbf{x}^a$ is not a fixed point of $\mathcal{F}$, i.e. $\mathcal{F}(\mathbf{x}^{a,o}) \neq \mathbf{x}^a$. Unlike in the classical situation where the probability that everyone guesses correctly when $\mathbf{x}$ is not a fixed point of $\mathcal{F}$ is 0, it might be the case that even when $\mathcal{F}$ has no fixed point, the game is won with a non-zero probability. The first phenomena is a characteristic of a guessing strategy, while the second is the work of noises. Our aim is to study the combination of these two characteristics on guessing games; therefore, we make the following definition:

Definition 28. *The winning probability of a guessing game $(G, 2, \epsilon)$ with strategy $\mathcal{F}$ is defined to be:*

$$\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] = \frac{\sum_{\mathbf{x}^a} \sum_{\mathbf{x}^o} \mathbf{P}[\mathcal{F}(\mathbf{x}^o) = \mathbf{x}^a]}{2^{|V(G)|}}. \tag{6.3}$$

Our aim is finding a strategy $\mathcal{F}$ that maximizes $\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})]$ given G and ϵ. There are two types of guessing strategies for this guessing game: *pure strategy* and *mixed strategy*. The guessing strategies of the first type are strategies in which there is no randomness involved in the guess each player makes given the values it sees. The alternative is a *mixed strategy* in which the players randomly choose a strategy to play from a set of pure strategies. The winning probability of the mixed strategy is the average of the winning probabilities of the pure strategies weighted according to the probabilities that they are chosen. This, however, is at most the maximum of the probabilities $\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})]$ of the pure strategies as proved by the following lemma.

Lemma 16. *Every randomised strategy for the guessing game $(G, 2, \epsilon)$ has winning probability at most $\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F}_{\text{opt}})]$, where $\mathcal{F}_{\text{opt}}$ is an optimal pure guessing strategy.*

Proof. Following our previous paragraph, a randomised strategy $\mathcal{G}$ can be described by assigning a probability $\mathbf{P}[\mathcal{G} = \mathcal{F}]$ to each deterministic strategy $\mathcal{F}$. The winning probability $\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{G})]$ of such a strategy is

$$\begin{aligned}\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{G})] &= \sum_{\mathcal{F}} \mathbf{P}[\mathcal{G} = \mathcal{F}]\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] \\ &\leq \max_{\mathcal{F}} \mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] = \mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F}_{\text{opt}})]\end{aligned}$$

□

Therefore, we gain no advantage by playing a mixed strategy. As such, throughout this chapter we will only ever consider pure strategies.

We now define the noisy guessing number of a guessing game $(G, 2, \epsilon)$, which will be our measure of the winning probability obtained by an optimal strategy for $(G, 2, \epsilon)$.

Definition 29. *The noisy guessing number* $\text{gn}(G, 2, \epsilon)$ *of a guessing game* $(G, 2, \epsilon)$ *is the largest* β *such that there exists a strategy* $\mathcal{F}$ *for* $(G, 2, \epsilon)$ *satisfies that*

$$\text{gn}(G, 2, \epsilon) = |V(G)| + \log_2 \left(\max_{\mathcal{F}} \mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] \right).$$

As it is suggested from the name of $\text{gn}(G, 2, \epsilon)$, the noisy guessing number shares many common properties with the guessing number we have been investigated.

Proposition 10.

$$\text{gn}(G, 2, \epsilon) \leq \text{gn}(G, 2) \leq \text{gn}(G). \tag{6.4}$$

Proof. We denote $\mathcal{F}$ for the optimal strategy giving us $\text{gn}(G, 2, \epsilon)$ in our definition. We have

$$\begin{aligned}\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] &= \frac{\sum_{\mathbf{x}^a} \sum_{\mathbf{x}^o} \mathbf{P}[\mathcal{F}(\mathbf{x}^o) = \mathbf{x}^a]}{2^{|V(G)|}} \\ &\leq \frac{\sum_{\mathbf{x}^a | \mathcal{F}(\mathbf{x}^{a,o}) = \mathbf{x}^a} 1}{2^{|V(G)|}} = \mathbf{P}[\text{Win}(G, 2, \mathcal{F})] \\ &\leq \max_{\mathcal{H}} \mathbf{P}[\text{Win}(G, s, \mathcal{H})]\end{aligned}$$

□

We have two direct consequences of Proposition 10:

Corollary 7. *Let* G *be an undirected graph,*

$$\text{gn}(G, 2, \epsilon) \leq |V(G)| - \alpha(G),$$

where $\alpha(G)$ *is the size of the maximal independent set of* G*.*

and

Corollary 8. *If G is acyclic, then*

$$\mathrm{gn}(G, 2, \epsilon) = 0.$$

Proof. Since $\mathrm{gn}(G) = 0$, the result follows from Proposition 10. □

Based on our definition of noisy guessing games, it is clear that the winning probability $\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$ of such a strategy $\mathcal{F}$ is a continuous function with respect to ϵ. In particular, we have the following Lemma.

Lemma 17. *For a guessing strategy $\mathcal{F}$ played on the noisy guessing game $(G, 2, \epsilon)$, $\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$ is a polynomial with variable ϵ of order $|V(G)|$. Particularly, $\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$ is continuous with respect to ϵ.*

Moreover,

$$\lim_{\epsilon \to 0} \mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})] = \mathbf{P}[\mathrm{Win}(G, 2, 0, \mathcal{F})] = \mathbf{P}[\mathrm{Win}(G, 2, \mathcal{F})]$$

Proof. We write the winning probability of $\mathcal{F}$ in terms of ϵ, and obtain a polynomial of the form

$$\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})] = 2^{|-V(G)|} \left(a_0(1-\epsilon)^m + a_1(1-\epsilon)^{m-1}\epsilon + \ldots + a_m \epsilon^m \right),$$

where a_i's are the number of observed values for which their assignments can be guessed correctly using strategy $\mathcal{F}$ under the condition that there are exactly i errors occurred.

This is a continuous map with respect to ϵ and $\mathbf{P}[\mathrm{Win}(G, 2, 0, \mathcal{F})] = \frac{a_0}{2^{|-V(G)|}}$ where a_0 is the number of fixed points of strategy $\mathcal{F}$. □

The following theorem suggests that when the noise level is small enough, we are be able to treat the noisy guessing game as if there is no noise involved.

Theorem 29. *For every noisy guessing game $(G, 2, \epsilon)$, there exists a positive ϵ_0 such that for all $\epsilon < \epsilon_0$, we can find an optimal guessing strategy of $(G, 2, \epsilon)$ which is also an optimal guessing strategy for $(G, 2, 0)$.*

Proof. As the number of guessing strategies is finite, we can always choose a positive ϵ_0 which is small enough such that there is a guessing strategy $\mathcal{F}$ which is an optimal guessing strategy for all positive noise level ϵ smaller than ϵ_0. In particular, we do not switch guessing strategy when ϵ goes to 0.

By Lemma 17, the noisy guessing number of strategy $\mathcal{F}$ is a continuous function of ϵ, and the value $2^{|V(G)|}\mathbf{P}[\text{Win}(G,2,\epsilon=0,\mathcal{F})]$ is equal a_0 which is the number of fixed points of $\mathcal{F}$. Since $2^{|V(G)|}\lim_{\epsilon\to 0}\mathbf{P}[\text{Win}(G,2,\epsilon,\mathcal{F})]$ converges to a_0 and by our assumption, we do not have to switch to another strategy whenever $0<\epsilon\leq\epsilon_0$, this forces the number of fixed points of $\mathcal{F}$ to be equal the number of fixed points of an optimal strategy when $\epsilon=0$. Hence, $\mathcal{F}$ is an optimal strategy in a noiseless guessing game. □

6.3 Case studies

We study some examples of noisy guessing games. As we would need to investigate all possible combinations of guessing strategies that can be played in our game, the following fact comes in handy:

Proposition 11. *The number of different guessing strategies for a single player v in $(G,2,\epsilon)$ is $2^{2^{|\Gamma_v^-|}}$.*

Proof. The number of distinct inputs that one player can observe is $2^{|\Gamma_v^-|}$. Therefore, it is sufficient to prove the statement by showing a bijection between $\mathcal{M}_{f_v}$ – the set of distinct guessing strategies of player v and $\mathcal{P}_{\Gamma_v^-}$ the power set of distinct inputs observed by that player. This map can be constructed explicitly as follows:

$$\begin{array}{rcl}\mathcal{M}_{f_v} & \to & \mathcal{P}_{\Gamma_v^-}\\ f_v & \mapsto & A:=\{x\in\{0,1\}^{|\Gamma_v^-|}: f_v(x)=1\}\end{array}$$

□

6.3.1 Noisy guessing games played on K_2

Let us investigate the behaviour of guessing strategies with respect to noise level when G is a clique of order two. In this situation, we have two players v_1 and v_2, each is assigned a number x_1^a and x_2^a from $\{0,1\}$ uniformly and independently at random. Player v_1 observes a value x_2^o[1] such that

$$\begin{cases}\mathbf{P}(x_2^o=x_2^a)=1-\epsilon\\ \mathbf{P}(x_2^o=(1-x_2^a) \mod 2)=\epsilon.\end{cases}$$

[1]In previous section we use the notation ${}_1x_2^o$

Table 6.1: Guessing strategies of v_i played in $(K_2, 2, \epsilon)$

Guessing strategies of v_i	observes 0	observes 1
$f^1_{v_i}$	0	0
$f^2_{v_i}$	1	1
$f^3_{v_i}$	0	1
$f^4_{v_i}$	1	0

Table 6.2: Representative of classes of guessing strategies for $(K_2, 2, \epsilon)$

$(f^1_{v_1}, f^1_{v_2})$	$(f^1_{v_1}, f^2_{v_2})$	$(f^2_{v_1}, f^2_{v_2})$	
$(f^1_{v_1}, f^3_{v_2})$	$(f^1_{v_1}, f^4_{v_2})$	$(f^2_{v_1}, f^3_{v_2})$	$(f^2_{v_1}, f^4_{v_2})$
$(f^3_{v_1}, f^3_{v_2})$	$(f^4_{v_1}, f^4_{v_2})$		
$(f^3_{v_1}, f^4_{v_2})$			

Similarly for player x_2 who observes a value x_1^{o2} such that

$$\begin{cases} \mathbf{P}(x_1^o = x_1^a) = 1 - \epsilon \\ \mathbf{P}(x_1^o = (1 - x_1^a) \mod 2) = \epsilon. \end{cases}$$

Table 6.1 illustrates all possible guessing strategies we can form for player v_i $(i = 1, 2)$.

We have 16 possible combinations of guessing strategies $(f^i_{v_1}, f^j_{v_2})$ $(i, j \in \{1, 2, 3, 4\})$. However, a lot of the pairs $(f^i_{v_1}, f^j_{v_2})$ are equivalent to each others due to the symmetry of K_2. For example, the pair $(f^1_{v_1}, f^2_{v_2})$ is equivalent to pair $(f^2_{v_1}, f^1_{v_2})$. Therefore, we can divide all possible guessing strategies into $10 = 4 + 6$ equivalent classes with representatives, which are described in table 6.2

Table 6.3 shows the guessing outputs for each guessing strategies corresponding to assigned values, and the winning probability for each guessing strategy is shown in table 6.4.

We will explain in a few words how we computed the winning probability for each guessing strategy appeared in table 6.4.

For guessing strategies $\mathcal{F} = (f^1_{v_1}, f^1_{v_2})$, $(f^1_{v_1}, f^2_{v_2})$, $(f^2_{v_1}, f^2_{v_2})$, the guessing value of each player is independent of the noises, therefore the winning probability for each of these strategies equals the winning probability of our noiseless guessing game playing with each of these strategies, which in turns equals the proportion of number of fixed points of $\mathcal{F}$ over the order of the configuration space.

[2] In previous section we use the notation ${}_2x_1^o$

For the case where $\mathcal{F}$ is one of those $\{(f^1_{v_1}, f^3_{v_2}), (f^1_{v_1}, f^4_{v_2}), (f^2_{v_1}, f^3_{v_2}), (f^2_{v_1}, f^4_{v_2})\}$, we see that one of the player uses a guessing strategy which is independent to the its assigned value, hence given a fixed point of the guessing strategy, the winning probability of their game depends on whether the second player can guess correctly its assigned value. The probability of this event is equal to the probability that no error appears when the second player observes the assigned value of the first player. Therefore,

$$\begin{aligned}\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})] &= \sum_{(x^a_i, x^a_j)\in\{0,1\}^2} \mathbf{P}[{}_i x^o_j = x^a_j]\mathbf{P}[\mathcal{F}(x^{a,o}_i, x^{a,o}_j) = (x^a_i, x^a_j)] \\ &= (1-\epsilon)\frac{1}{4} = \frac{1-\epsilon}{4}.\end{aligned}$$

We mentioned in the previous section the phenomenon where the interaction between noises and guessing strategies comes into play. This aspect is illustrated in the case where $\mathcal{F}$ is one of those $\{(f^3_{v_1}, f^3_{v_2}), (f^4_{v_1}, f^4_{v_2})\}$. We demonstrate the calculation for $\mathcal{F} = (f^3_{v_1}, f^3_{v_2})$. In this situation, we have

$$\begin{aligned}\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})] &= \sum_{(\mathbf{x}^a)\in\{0,1\}^2} \sum_{\mathbf{x}^o\in\{0,1\}^2} \mathbf{P}[\mathcal{F}(\mathbf{x}^o) = \mathbf{x}^a] \\ &= \mathbf{P}[\mathbf{x}^a = 00]\mathbf{P}[\mathbf{x}^o = 00] + \mathbf{P}[\mathbf{x}^a = 11]\mathbf{P}[\mathbf{x}^o = 11] \\ &+ \mathbf{P}[\mathbf{x}^a = 01]\mathbf{P}[\mathbf{x}^o = 10] + \mathbf{P}[\mathbf{x}^a = 10]\mathbf{P}[\mathbf{x}^o = 10] \\ &= \frac{1}{4}(1-\epsilon)^2 + \frac{1}{4}(1-\epsilon)^2 \\ &+ \frac{1}{4}\epsilon^2 + \frac{1}{4}\epsilon^2 \\ &= \frac{(1-2\epsilon+2\epsilon^2)}{2}.\end{aligned}$$

The ‘collaborative’ interaction between noises and a guessing strategy contributes a portion of $\frac{\epsilon^2}{2}$ toward the winning probability.

The presence of this noisy phenomenon plays a significant role in the last type of guessing strategy where $\mathcal{F} = (f^3_{v_1}, f^4_{v_2})$. For this case

$$\begin{aligned}\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})] &= \mathbf{P}[\mathbf{x}^a = 00]\mathbf{P}[\mathbf{x}^o = 10] + \mathbf{P}[\mathbf{x}^a = 01]\mathbf{P}[\mathbf{x}^o = 00] \\ &+ \mathbf{P}[\mathbf{x}^a = 10]\mathbf{P}[\mathbf{x}^o = 11] + \mathbf{P}[\mathbf{x}^a = 11]\mathbf{P}[\mathbf{x}^o = 01] \\ &= \epsilon - \epsilon^2.\end{aligned}$$

Following our calculation, we have

Table 6.3: Inputs–outputs of 10 representative guessing strategies for $(K_2, 2, \epsilon)$

	$(f^1_{v_1}, f^1_{v_2})$	$(f^1_{v_1}, f^2_{v_2})$	$(f^2_{v_1}, f^2_{v_2})$	
00	00	01	11	
01	00	01	11	
10	00	01	11	
11	00	01	11	
	$(f^1_{v_1}, f^3_{v_2})$	$(f^1_{v_1}, f^4_{v_2})$	$(f^2_{v_1}, f^3_{v_2})$	$(f^2_{v_1}, f^4_{v_2})$
00	00	01	00	01
01	00	01	00	01
10	01	00	01	00
11	01	00	01	00
	$(f^3_{v_1}, f^3_{v_2})$	$(f^4_{v_1}, f^4_{v_2})$		
00	00	11		
01	10	01		
10	01	10		
11	11	00		
	$(f^3_{v_1}, f^4_{v_2})$			
00	01			
01	11			
10	00			
11	10			

Table 6.4: Winning probability of guessing strategies for $(K_2, 2, \epsilon)$

Guessing strategy $\mathcal{F}$	$(f^1_{v_1}, f^1_{v_2})$	$(f^1_{v_1}, f^2_{v_2})$	$(f^2_{v_1}, f^2_{v_2})$	$(f^4_{v_1}, f^4_{v_2})$
$\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
Guessing strategy $\mathcal{F}$	$(f^1_{v_1}, f^3_{v_2})$	$(f^1_{v_1}, f^4_{v_2})$	$(f^2_{v_1}, f^3_{v_2})$	$(f^2_{v_1}, f^4_{v_2})$
$\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$	$\frac{(1-\epsilon)}{4}$	$\frac{(1-\epsilon)}{4}$	$\frac{(1-\epsilon)}{4}$	$\frac{(1-\epsilon)}{4}$
Guessing strategy $\mathcal{F}$	$(f^3_{v_1}, f^3_{v_2})$	$(f^4_{v_1}, f^4_{v_2})$		
$\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$	$\frac{(1-2\epsilon+2\epsilon^2)}{2}$	$\frac{(1-2\epsilon+2\epsilon^2)}{2}$		
Guessing strategy $\mathcal{F}$	$(f^3_{v_1}, f^4_{v_2})$			
$\mathbf{P}[\mathrm{Win}(G, 2, \epsilon, \mathcal{F})]$	$\epsilon - \epsilon^2$			

Table 6.5: Guessing strategies of v_1 and v_3 played in $(P_2, 2, \epsilon)$

Guessing strategies of v_i	observes 0	observes 1
$f_{v_i}^1$	0	0
$f_{v_i}^2$	1	1
$f_{v_i}^3$	0	1
$f_{v_i}^4$	1	0

Theorem 30. *For $0 \leq \epsilon < \frac{1}{2}$, the highest probability of winning $(K_2, 2, \epsilon)$ can be obtained by using either guessing strategy $(f_{v_1}^3, f_{v_2}^3)$ or strategy $(f_{v_1}^4, f_{v_2}^4)$. The noisy guessing number is*

$$\mathrm{gn}(G, 2, \epsilon) = 1 - \log_2(1 - 2\epsilon + 2\epsilon^2).$$

When $\epsilon = \frac{1}{2}$, $\mathrm{gn}(G, 2, \epsilon) = 0$.

6.3.2 Noisy guessing games played on P_3

Let us consider another example of a noisy guessing game where $G = P_3$. In this situation, we have three players v_1, v_2 and v_3, each is assigned a number x_1^a, x_2^a and x_3^a from $\{0, 1\}$ uniformly and independently at random. Player v_1 observes a value ${}_1x_2^o$ such that

$$\begin{cases} \mathbf{P}({}_1x_2^o = x_2^a) = 1 - \epsilon \\ \mathbf{P}({}_1x_2^o = (1 - x_2^a) \mod 2) = \epsilon. \end{cases}$$

Similarly for player v_3 who observes a value ${}_3x_2^o$ such that

$$\begin{cases} \mathbf{P}({}_3x_2^o = x_2^a) = 1 - \epsilon \\ \mathbf{P}({}_3x_2^o = (1 - x_2^a) \mod 2) = \epsilon. \end{cases}$$

For player v_2, it observes values ${}_2x_3^o,{}_2 x_1^o$ such that

$$\begin{cases} \mathbf{P}({}_2x_1^o = x_1^a) = 1 - \epsilon \\ \mathbf{P}({}_2x_1^o = (1 - x_1^a) \mod 2) = \epsilon. \end{cases}$$

and

$$\begin{cases} \mathbf{P}({}_2x_3^o = x_3^a) = 1 - \epsilon \\ \mathbf{P}({}_2x_3^o = (1 - x_3^a) \mod 2) = \epsilon. \end{cases}$$

Table 6.5 illustrates all possible guessing strategies we can form for player v_1 and v_3, while all possible guessing strategies we can form for player v_2 are demonstrated in Table 6.6.

Table 6.6: Guessing strategies of v_2 played in $(P_2, 2, \epsilon)$

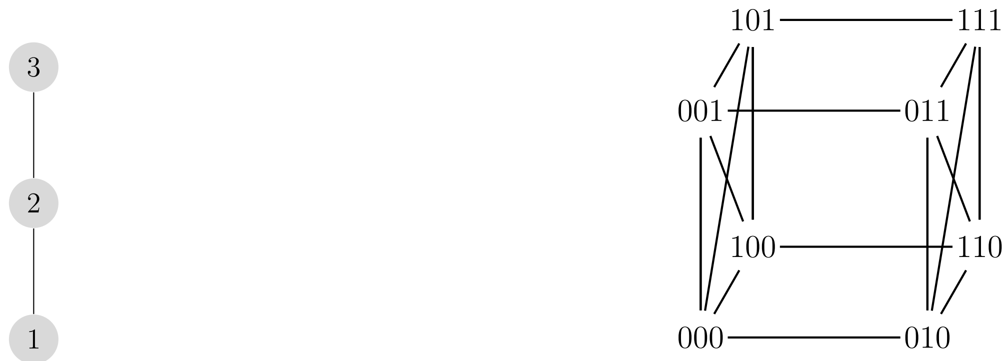
Guessing strategies of v_2	observes 00	observes 01	observes 10	observes 11
$f_{v_2}^1$	0	0	0	0
$f_{v_2}^2$	0	0	0	1
$f_{v_2}^3$	0	0	1	0
$f_{v_2}^4$	0	0	1	1
$f_{v_2}^5$	0	1	0	0
$f_{v_2}^6$	0	1	0	1
$f_{v_2}^7$	0	1	1	0
$f_{v_2}^8$	0	1	1	1
$f_{v_2}^9$	1	0	0	0
$f_{v_2}^{10}$	1	0	0	1
$f_{v_2}^{11}$	1	0	1	0
$f_{v_2}^{12}$	1	0	1	1
$f_{v_2}^{13}$	1	1	0	0
$f_{v_2}^{14}$	1	1	0	1
$f_{v_2}^{15}$	1	1	1	0
$f_{v_2}^{16}$	1	1	1	1

Proposition 12. *The number of fixed points for any guessing strategies played in* $(P_3, 2, \epsilon)$ *can not be greater than* 2.

Proof. According to the definition of fixed points, these are points for which the guessing value is equal to the assigned value under the condition that no error occurred. Therefore, fixed points of a guessing strategy in noisy guessing games $(P_3, 2, \epsilon)$ are also fixed points of the same guessing strategy played in noiseless guessing game $(P_3, 2)$. From Section 3.3, we know that these points are equivalent to independent set of code graph $X(P_3, 2)$. Figure 6.1 illustrates the code graph $X(P_3, 2)$, and it is easy to see that this graph has independent number $\alpha(X(P_3, 2)) = 2$. □

The automorphism group of P_3 is of order 2; therefore we can classify all guessing strategies for $(P_3, 2, \epsilon)$ into equivalent classes similar to what was done in our previous example $(K_2, 2, \epsilon)$. However, since we are only interested in looking at the optimal guessing strategy, we divide guessing strategies for $(P_3, 2, \epsilon)$ into general categories as follows:

- Class $\mathcal{M}_1$ contains strategies of which guessing function used by each player is a constant map.



(a) P_3 (b) $X(P_3, 2)$

Figure 6.1: Guessing game on P_3 and its code graph $X(P_3, 2)$.

Table 6.7: Optimal guessing strategies for $(P_3, 2, \epsilon)$

Assigned value	000	001	010	011	100	101	110	111
$(f^1_{v_1}, f^{13}_{v_2}, f^3_{v_3})$	011	001	010	000	011	001	010	000
$(f^2_{v_1}, f^{13}_{v_2}, f^3_{v_3})$	011	001	110	100	011	001	110	100
$(f^2_{v_1}, f^4_{v_2}, f^2_{v_3})$	000	010	101	111	000	010	101	111
$(f^3_{v_1}, f^{13}_{v_2}, f^3_{v_3})$	111	101	010	000	111	101	010	000
$(f^3_{v_1}, f^4_{v_2}, f^2_{v_3})$	100	110	001	011	100	110	001	011
$(f^4_{v_1}, f^{13}_{v_2}, f^3_{v_3})$	111	101	110	100	111	101	110	100
$(f^4_{v_1}, f^4_{v_2}, f^2_{v_3})$	100	110	101	111	100	110	101	111

- Class $\mathcal{M}_2$ contains strategies of which two players use guessing functions which are constant functions, and the last player guess its value based only on the value of one of its neighbourhood.
- Class $\mathcal{M}_3$ contains strategies of which at least two players collaborate.

Guessing strategies belonging to class $\mathcal{M}_1$, and $\mathcal{M}_2$ are generally similar to strategies covered in the previous example. The winning probabilities for these strategies are $\frac{1}{8}$ and $\frac{(1-\epsilon)}{8}$. The argument for computing the winning probability is similar to the case $(K_2, 2, \epsilon)$. Compared to the winning probability of similar strategies played in $(K_2, 2, \epsilon)$, we see that there is an extra factor of $\frac{1}{2}$ since we have one extra player who guesses its value randomly.

The interesting case is when we have at least two players collaborating with each other. We would expect that the optimal winning probability can be achieved by guessing strategies belonging to $\mathcal{M}_3$. In fact, this is the case. Table 6.7 provides a list of all guessing strategies that obtain an optimal winning probability for $(P_3, 2, \epsilon)$.

The winning probability using any of these strategies is

$$\mathbf{P}[\mathrm{Win}(G,2,\epsilon,\mathcal{F})] = \frac{1}{4}(1-\epsilon)^4 + \frac{1}{2}(1-\epsilon)^3\epsilon + \frac{1}{2}(1-\epsilon)^2\epsilon^2 + \frac{1}{2}(1-\epsilon)\epsilon^3 + \frac{1}{4}\epsilon^4.$$

As we are only concerned about the optimal winning probability, we do not cover other strategies here as they provide a smaller chance of winning compared to strategies listed in Table 6.7. We derived a complete list of winning probabilities for each guessing strategy via computer calculation and the data will be provided upon request.

Following our calculation, we have

Theorem 31. *For $0 \leq \epsilon < \frac{1}{2}$, the highest probability of winning $(P_3, 2, \epsilon)$ can be obtained by using guessing strategies illustrated in Table 6.7. The noisy guessing number is*

$$\mathrm{gn}(G,2,\epsilon) = 3 + \log_2(\frac{1}{4}(1-\epsilon)^4 + \frac{1}{2}(1-\epsilon)^3\epsilon + \frac{1}{2}(1-\epsilon)^2\epsilon^2 + \frac{1}{2}(1-\epsilon)\epsilon^3 + \frac{1}{4}\epsilon^4).$$

When $\epsilon = \frac{1}{2}$, $\mathrm{gn}(G,2,\epsilon) = 0$.

6.3.3 Noisy guessing games played on K_3

Let us consider another example of noisy guessing game where $G = K_3$. In this situation, we have three players v_1, v_2 and v_3, each is assigned a number x_1^a, x_2^a and x_3^a from $\{0,1\}$ uniformly and independently at random. Each player v_i observes values ${}_ix_j^o, {}_ix_k^o$ such that

$$\begin{cases} \mathbf{P}({}_ix_j^o = x_j^a) = 1-\epsilon \\ \mathbf{P}({}_ix_j^o = (1-x_j^a) \mod 2) = \epsilon. \end{cases}$$

and

$$\begin{cases} \mathbf{P}({}_ix_k^o = x_k^a) = 1-\epsilon \\ \mathbf{P}({}_ix_k^o = (1-x_k^a) \mod 2) = \epsilon. \end{cases}$$

Table 6.8 illustrates all 16 possible guessing strategies that we can form for each player v_i $(i = 1,2,3)$.

Proposition 13. *The number of fixed points for any guessing strategy played in $(K_3, 2, \epsilon)$ cannot be greater than 4.*

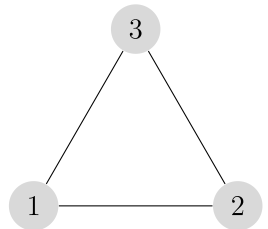
Proof. Based on the definition of fixed points, these are points for which the guessing value equals the assigned value under the condition that no error occurred in observation. Therefore, fixed points of a guessing strategy in noisy guessing games $(K_3, 2, \epsilon)$

Table 6.8: Guessing strategies of v_i played in $(K_3, 2, \epsilon)$

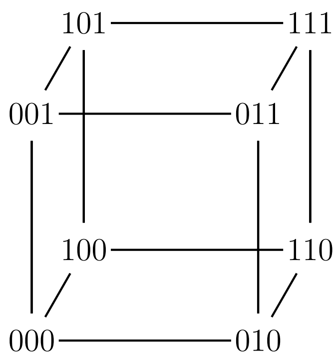
Guessing strategies of v_i	observes 00	observes 01	observes 10	observes 11
$f_{v_i}^1$	0	0	0	0
$f_{v_i}^2$	0	0	0	1
$f_{v_i}^3$	0	0	1	0
$f_{v_i}^4$	0	0	1	1
$f_{v_i}^5$	0	1	0	0
$f_{v_i}^6$	0	1	0	1
$f_{v_i}^7$	0	1	1	0
$f_{v_i}^8$	0	1	1	1
$f_{v_i}^9$	1	0	0	0
$f_{v_i}^{10}$	1	0	0	1
$f_{v_i}^{11}$	1	0	1	0
$f_{v_i}^{12}$	1	0	1	1
$f_{v_i}^{13}$	1	1	0	0
$f_{v_i}^{14}$	1	1	0	1
$f_{v_i}^{15}$	1	1	1	0
$f_{v_i}^{16}$	1	1	1	1

are also fixed points of the same guessing strategy played in noiseless guessing game $(K_3, 2)$. From Section 3.3, we know that these points are equivalent to independent set of code graph $X(K_3, 2)$. Figure 6.1 illustrates the code graph $X(K_3, 2)$, and it is easy to see that this graph has the independent number $\alpha(X(K_3, 2)) = 4$. □

We derived a complete list of winning probability for each guessing strategy via computer calculation and the data will be provided upon request. Interestingly, the optimal guessing strategy for $(K_3, 2, \epsilon)$ depends heavily on the noise level ϵ. We give



(a) K_3



(b) $X(K_3, 2)$

Figure 6.2: Guessing game on K_3 and its code graph $X(K_3, 2)$.

Table 6.9: Representative of optimal guessing strategies for $(K_3, 2, \epsilon)$

Assigned value	000	001	010	011	100	101	110	111
$(f^{10}_{v_1}, f^{10}_{v_2}, f^{10}_{v_3})$	111	001	010	100	100	010	001	111
$(f^{9}_{v_1}, f^{9}_{v_2}, f^{9}_{v_3})$	111	001	010	000	100	000	000	000
$(f^{1}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	011	001	010	000	011	001	010	000

a list of representative guessing strategies for $(K_3, 2, \epsilon)$ in Table 6.9

The winning probability for using strategy $(f^{10}_{v_1}, f^{10}_{v_2}, f^{10}_{v_3})$ is

$$\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] = 2^{-3}\left(4(1-\epsilon)^6 + 12(1-\epsilon)^4\epsilon^2 + 32(1-\epsilon)^3\epsilon^3 + 12(1-\epsilon)^2\epsilon^4 + 4\epsilon^6\right).$$

The winning probability for using strategy $(f^{9}_{v_1}, f^{9}_{v_2}, f^{9}_{v_3})$ is

$$\begin{aligned}\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] &= 2^{-3}(3(1-\epsilon)^6 + 6(1-\epsilon)^5\epsilon + 12(1-\epsilon)^4\epsilon^2 + 20(1-\epsilon)^3\epsilon^3 \\ &\quad + 15(1-\epsilon)^2\epsilon^4 + 6(1-\epsilon)^2\epsilon^4 + 2\epsilon^6).\end{aligned}$$

The winning probability for using strategy $(f^{1}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$ is

$$\begin{aligned}\mathbf{P}[\text{Win}(G, 2, \epsilon, \mathcal{F})] &= 2^{-3}(2(1-\epsilon)^6 + 8(1-\epsilon)^5\epsilon + 14(1-\epsilon)^4\epsilon^2 + 16(1-\epsilon)^3\epsilon^3 \\ &\quad + 14(1-\epsilon)^2\epsilon^4 + 8(1-\epsilon)^2\epsilon^4 + 2\epsilon^6).\end{aligned}$$

We observe a very interesting phenomenon in this example where the optimal guessing strategy depends heavily on the noise level. In particular, when $0 \leq \epsilon < 0.17$, the guessing strategy $(f^{10}_{v_1}, f^{10}_{v_2}, f^{10}_{v_3})$ is invoked. This strategy provides 4 fixed points and it is equivalent to the optimal guessing strategy played in a noiseless guessing game $(K_3, 2)$. When $0.17 \leq \epsilon < 0.35$, the guessing strategy $(f^{9}_{v_1}, f^{9}_{v_2}, f^{9}_{v_3})$ is adopted. Interestingly, this strategy only provides 3 fixed points; therefore, it is non-linear. When $0.35 \leq \epsilon < 0.5$ the guessing strategy $(f^{1}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$ is selected. This guessing strategy provides only 2 fixed points. When $\epsilon = 0.5$, every guessing strategy provides the same result with winning probability equal to $\frac{1}{8}$. Table 6.10 provides a list of all guessing strategies that have a similar winning probability for $(K_3, 2, \epsilon)$. We note that the values 0.17, 0.35 are approximations taken from the data file.

Following our calculation, we have

Theorem 32. *For $0 \leq \epsilon < 0.17$, the highest probability of winning $(K_3, 2, \epsilon)$ can be obtained by using either guessing strategy $(f^{10}_{v_1}, f^{10}_{v_2}, f^{10}_{v_3})$ or its equivalences (presented*

Table 6.10: Classes of optimal guessing strategies for $(K_3, 2, \epsilon)$

Representative	Guessing strategies			Range of ϵ
$(f^{10}_{v_1}, f^{10}_{v_2}, f^{10}_{v_3})$	$(f^{10}_{v_1}, f^{10}_{v_2}, f^{10}_{v_3})$	$(f^{7}_{v_1}, f^{7}_{v_2}, f^{7}_{v_3})$		$0 \leq \epsilon < 0.17$
$(f^{9}_{v_1}, f^{9}_{v_2}, f^{9}_{v_3})$	$(f^{15}_{v_1}, f^{15}_{v_2}, f^{15}_{v_3})$	$(f^{14}_{v_1}, f^{2}_{v_2}, f^{12}_{v_3})$	$(f^{12}_{v_1}, f^{12}_{v_2}, f^{2}_{v_3})$	$0.17 \leq \epsilon < 0.35$
	$(f^{8}_{v_1}, f^{5}_{v_2}, f^{5}_{v_3})$	$(f^{9}_{v_1}, f^{9}_{v_2}, f^{9}_{v_3})$	$(f^{5}_{v_1}, f^{8}_{v_2}, f^{9}_{v_3})$	
	$(f^{2}_{v_1}, f^{14}_{v_2}, f^{14}_{v_3})$	$(f^{3}_{v_1}, f^{3}_{v_2}, f^{8}_{v_3})$		
$(f^{1}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{13}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{13}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{11}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$0.35 \leq \epsilon \leq 0.5$
	$(f^{11}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{2}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{10}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	
	$(f^{10}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{7}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{7}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	
	$(f^{6}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{6}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{4}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	
	$(f^{4}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{15}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{15}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	
	$(f^{14}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{14}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{12}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	
	$(f^{12}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{8}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{8}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	
	$(f^{16}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{16}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{1}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	
	$(f^{1}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{9}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{9}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	
	$(f^{5}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	$(f^{5}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{3}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$	
	$(f^{3}_{v_1}, f^{4}_{v_2}, f^{4}_{v_3})$	$(f^{2}_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$		

in Table 6.10) The noisy guessing number in this case is

$$\begin{aligned}\text{gn}(K_3, 2, \epsilon) = \log_2(&4(1-\epsilon)^6 + 12(1-\epsilon)^4\epsilon^2 \\ &+ 32(1-\epsilon)^3\epsilon^3 + 12(1-\epsilon)^2\epsilon^4 + 4\epsilon^6).\end{aligned}$$

For $0.17 \leq \epsilon < 0.35$, *the highest probability of winning* $(K_3, 2, \epsilon)$ *can be obtained by using either guessing strategy* $(f^9_{v_1}, f^9_{v_2}, f^9_{v_3})$ *or its equivalences (presented in Table 6.10) The noisy guessing number in this case is*

$$\begin{aligned}\text{gn}(K_3, 2, \epsilon) = \log_2(&3(1-\epsilon)^6 + 6(1-\epsilon)^5\epsilon + 12(1-\epsilon)^4\epsilon^2 \\ &+ 20(1-\epsilon)^3\epsilon^3 + 15(1-\epsilon)^2\epsilon^4 + 6(1-\epsilon)^2\epsilon^4 + 2\epsilon^6.).\end{aligned}$$

For $0.35 \leq \epsilon < 0.5$, *the highest probability of winning* $(K_3, 2, \epsilon)$ *can be obtained by using either guessing strategy* $(f^1_{v_1}, f^{13}_{v_2}, f^{13}_{v_3})$ *or its equivalences (presented in Table 6.10) The noisy guessing number in this case is*

$$\begin{aligned}\text{gn}(K_3, 2, \epsilon) = \log_2(&2(1-\epsilon)^6 + 8(1-\epsilon)^5\epsilon + 14(1-\epsilon)^4\epsilon^2 \\ &+ 16(1-\epsilon)^3\epsilon^3 + 14(1-\epsilon)^2\epsilon^4 + 8(1-\epsilon)^2\epsilon^4 + 2\epsilon^6).\end{aligned}$$

When $\epsilon = \frac{1}{2}$, $\text{gn}(G, 2, \epsilon) = 0$.

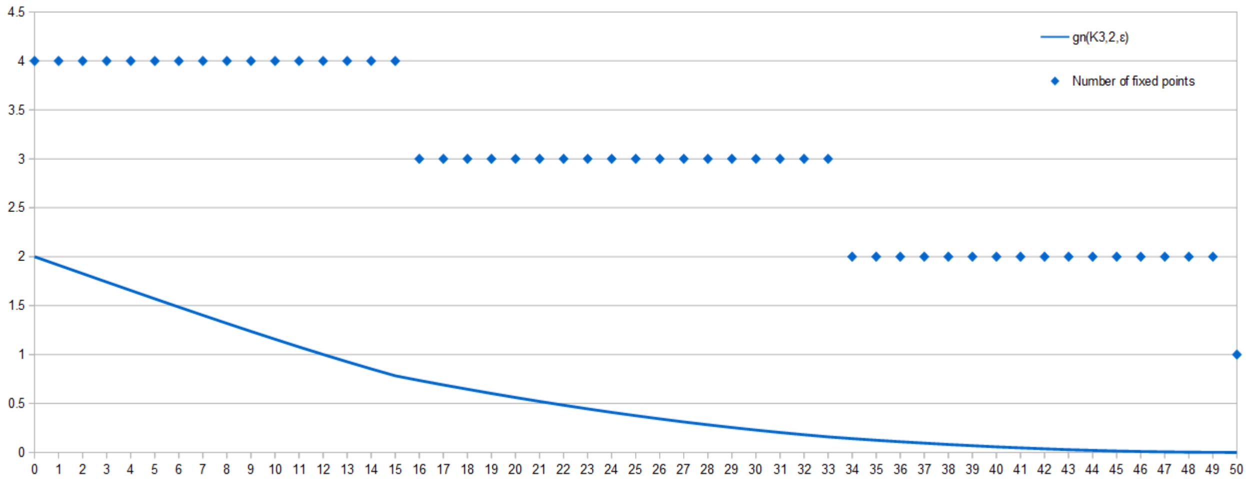


Figure 6.3: Noisy guessing number/Number of fixed points vs Noise level (%)

Chapter 7

Conclusion

7.1 Summary

The guessing game played on digraphs is a combinatorial framework that is based on a simplified version of multiple-unicast network coding. This game lies in the intersection of graph theory, network coding, and circuit complexity theory. In particular, it connects the solvability of a multiple-unicast network with the value of a combinatorial number known as guessing number. Moreover, the guessing number is proved to be equal to the order of the entropy of the same digraph. Hence, the nature of this problem is strongly influenced by fundamental characteristics of information theory.

In this paper, we have shown that the guessing game, even in its simplest form which is a game being played on an undirected graph, possesses 'conjecturally' all informatic properties that are admitted by a general network coding problem. In particular, we have established examples of guessing games played on (un)directed graphs possessing the property that there are gaps between different bounds provided by different families of information inequalities. These graphs add new test cases to the list of graphs with special informatic properties which previously contained the Vámos network and its variations. Moreover, our examples are of small order, therefore, the calculation and verification can be done entirely by computer. In fact, the process of deriving these bounds is achieved by computer. In addition, using the method of constructing a new graph based on graph product, our example leaves room for improvement in terms of extending these gaps further.

Another question that we improved is the existence of an irreversible guessing game on digraph. Even-though we could not find an answer to this problem in its strongest form, we established an example which justifies the conjectured. Our example R^S has the property that the Shannon bound of R^S and the Shannon bound of Reverse(R^S) did not match.

Unfortunately, we made no serious improvement to the problem of tracing for an algorithm that can compute the guessing number of an arbitrary graph. Limited by the number of tools at our disposal, we added to the literature new families of graphs with their exact guessing numbers derived. In addition, we managed to disprove a conjecture which links the guessing number with a combinatorial value defined purely based on the graph topology. In establishing our counter-examples, we also showed the behaviour of different lower-bounds which are coming from different families of codes. To construct some of our examples, combinatorial structures known as the Steiner system were invoked. These structures are very interesting on its own, which have found applications in representation theory – an influential branch of pure mathematics. The existence of a generalization of the Steiner system was proved by Keevash [60]. This vital work opens the door for us towards constructing many more instances of guessing games with interesting combinatorial properties.

Last but not least, we initialize the study of noisy guessing games. Though our results are simple and naive, they are remarks of an effort of dealing with 'practical' problems in network coding. Notably, many interesting phenomena have already appeared in our studies, which are worth investigation further.

7.2 Open questions and future research

At this point, we can see clearly that there are many open questions can be raised in our study ranging from specific examples to broadly defined ideas, e.g.

- What is the exact value of the guessing number of R^-? We emphasise that R^- is the only undirected graph on at most 10 vertices whose guessing number remains undetermined in our calculation. Any lower bound that implies $\text{gn}(R^-) > 20/3$ would show that there exists a non-linear guessing strategy that outperforms the fractional clique cover strategy for R^-.

- Whether there exists an irreversible guessing game, i.e. a guessing game G such that $\text{gn}(G) \neq \text{gn}(\text{Reverse}(G))$. This can be answered in the affirmative if $\text{gn}(R^L)$ can be shown to be strictly larger than $27/4 = 6.75$. Unfortunately, this might be hard to prove as it would establish the existence of a non-linear guessing strategy that improves the lower bound we derived.

- Can we compute the guessing numbers of the Higman–Sims graph, Clebsch graph, Hoffman–Singleton graph, Gewirtz graph, and M_{22}? We notice that these graphs are large and therefore we have no powerful computation system to derive the upper-bound of these graphs using the entropy arguments. However, these graphs are triangle-free and strongly regular, which implies that the associated code graphs possess interesting symmetrical structures, which may help us derive the guessing number in a combinatorial way.

- Can we find an algorithm to compute the guessing number? Or can we prove that no such algorithm exists?

- Can we define a noisy guessing number independently from the alphabet size assuming that the communication channels are symmetric? Can we define noisy guessing numbers for more general types of communication channels?

Each of these questions are directly concerned with some of the deepest open problems in information theory. Therefore, it is unlikely to expect an answer to these problems within a short period of time. However, there are reasonable yet non-trivial open questions that we are also interested in. These questions arise from our works of computing exact guessing numbers of families of graphs appeared in Chapter 5. It is noticeable that all of the counter-examples to the fractional clique cover conjecture that appeared in this thesis are non-planar. This fact can be easily proved following the Kuratowski's theorem saying that:

Theorem 33 (Kuratowski's theorem). *A finite graph is planar if and only if it contains no subgraph homeomorphic to K_5 (complete graph on five vertices) or $K_{3,3}$ (complete bipartite graph on six vertices, three of which connect to each of the other three).*

Obviously, our graphs R and R^- contain subgraphs that are homeomorphic to $K_{3,3}$. Moreover, each strongly regular triangle-free planar graph appeared in Section 4.3 contains copies of the Petersen graph which in turns contains a minor isomorphic to the $K_{3,3}$ graph.

In contrast, the families of graphs that we derived their exact guessing numbers in Chapter 5 are planar triangle-free graphs. This observation motivates our following conjecture:

Conjecture 15. The fractional clique cover conjecture holds if the underlying graph is a planar triangle-free graph.

We can even hope for a more general statement:
Conjecture 16. The fractional clique cover conjecture holds if the underlying graph is a planar graph.

Note that this conjecture holds when our graph is K_4 which is a planar graph but is not a triangle-free graph.

These conjectures, by far, are the most general conjectured statements about guessing numbers of families of undirected graphs which remain open.

Another problem that might be within our reach is about building asymptotic statements about guessing numbers of families of graphs. When we say "a family of graphs," we mean a sequence of (un)directed graphs $\{G_1, G_2, \ldots, G_n, \ldots\}$, where G_i is isomorphic to a subgraph of G_{i+1} plus some extra conditions. An asymptotic behaviour is a behaviour of the sequence $\{\text{gn}(G_1), \text{gn}(G_2), \ldots, \text{gn}(G_n), \ldots\}$ under different constraints in defining the family $\{G_1, G_2, \ldots, G_n, \ldots\}$.

References

[1] A. Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Information Theory Workshop, 2004. IEEE*, pages 247–249, Oct 2004.

[2] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. Network information flow. *Information Theory, IEEE Transactions on*, 46(4):1204–1216, Jul 2000.

[3] N. Alon, E. Lubetzky, U. Stav, A. Weinstein, and A. Hassidim. Broadcasting with side information. In *Foundations of Computer Science, 2008. FOCS '08. IEEE 49th Annual IEEE Symposium on*, pages 823–832, Oct 2008.

[4] Rahil Baber, Demetres Christofides, Anh N. Dang, Søren Riis, and Emil R. Vaughan. Multiple unicasts, graph guessing games, and non-shannon inequalities. In *International Symposium on Network Coding, NetCod 2013, Calgary, AB, Canada, June 7-9, 2013*, pages 1–6, 2013.

[5] Rahil Baber, Demetres Christofides, Anh Nhat Dang, Søren Riis, and Emil R. Vaughan. Graph guessing games and non-shannon information inequalities. *CoRR*, abs/1410.8349, 2014.

[6] Hossein Bahramgiri and Farshad Lahouti. Robust network coding against path failures. *Communications, IET*, 4(3):272–284, 2010.

[7] Z. Bar-Yossef, Y. Birk, T.S. Jayram, and T. Kol. Index coding with side information. *Information Theory, IEEE Transactions on*, 57(3):1479–1494, March 2011.

[8] Ziv Bar-Yossef, Yitzhak Birk, TS Jayram, and Tomer Kol. Index coding with side information. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 197–206. IEEE, 2006.

[9] R. Bassoli, H. Marques, J. Rodriguez, K.W. Shum, and R. Tafazolli. Network coding theory: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1950–1978, Fourth 2013.

[10] Claude Berge. Perfect graphs. In Six Papers on Graph Theory, Indian Statistical Institute, 1963., 1963.

[11] Y. Birk and T. Kol. Coding on demand by an informed source (iscod) for efficient broadcast of different supplemental data to caching clients. *Information Theory, IEEE Transactions on*, 52(6):2825–2830, June 2006.

[12] A. Blasiak, R. Kleinberg, and E. Lubetzky. Broadcasting with side information: Bounding and approximating the broadcast rate. *Information Theory, IEEE Transactions on*, 59(9):5811–5823, Sept 2013.

[13] Anna Blasiak. *A Graph-Theoretic Approach to Network Coding*. PhD thesis, Cornell University, 2013.

[14] Anna Blasiak and Robert Kleinberg. The serializability of network codes. In *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming: Part II*, ICALP'10, pages 100–114, Berlin, Heidelberg, 2010. Springer-Verlag.

[15] Anna Blasiak, Robert Kleinberg, and Eyal Lubetzky. Lexicographic products and the power of non-linear network coding. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 609–618, 2011.

[16] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 1998.

[17] Martin Bossert and Ernst M Gabidulin. One family of algebraic codes for network coding. In *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory-Volume 4*, pages 2863–2866. IEEE Press, 2009.

[18] A. Brouwer. Graph descriptions.

[19] A. E. Brouwer and C. A. Van Eijl. On the p-rank of the adjacency matrices of strongly regular graphs. *Journal of Algebraic Combinatorics 1*, 4:329–346, 1992.

[20] Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoss, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6:1–6:33, February 2013.

[21] Ning Cai and Raymond W Yeung. Network coding and error correction.

[22] Ning Cai, Raymond W Yeung, et al. Network error correction, ii: Lower bounds. *Communications in Information & Systems*, 6(1):37–54, 2006.

[23] Peter J. Cameron, Anh N. Dang, and Søren Riis. Guessing games on triangle-free graphs. *CoRR*, abs/1410.2405, 2014.

[24] Peter J Cameron and Jacobus Hendricus Van Lint. *Designs, graphs, codes, and their links*. Cambridge University Press, 1992.

[25] T. Chan and A. Grant. Dualities between entropy functions and network codes. *Information Theory, IEEE Transactions on*, 54(10):4470–4487, Oct 2008.

[26] Terence Chan and Alex Grant. Non-linear information inequalities. *Entropy*, 10(4):765–775, 2008.

[27] Terence H Chan. Capacity regions for linear and abelian network codes. In *Information Theory and Applications Workshop, 2007*, pages 73–78. IEEE, 2007.

[28] Demetres Christofides and Klas Markström. The guessing number of undirected graphs. *Electr. J. Comb.*, 18(1), 2011.

[29] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Ann. Math. (2)*, 164(1):51–229, 2006.

[30] Thomas M. Cover and Joy A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[31] M. Di Renzo, M. Iezzi, and F. Graziosi. Beyond routing via network coding: An overview of fundamental information-theoretic results. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 2745–2750, Sept 2010.

[32] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[33] A.G. Dimakis, K. Ramchandran, Y. Wu, and Changho Suh. A survey on network codes for distributed storage. *Proceedings of the IEEE*, 99(3):476–489, March 2011.

[34] R. Dougherty, C. Freiling, and K. Zeger. Insufficiency of linear coding in network information flow. *Information Theory, IEEE Transactions on*, 51(8):2745–2759, Aug 2005.

[35] R. Dougherty, C. Freiling, and K. Zeger. Six new non-shannon information inequalities. In *Information Theory, 2006 IEEE International Symposium on*, pages 233–236, July 2006.

[36] R. Dougherty, C. Freiling, and K. Zeger. Networks, matroids, and non-shannon information inequalities. *Information Theory, IEEE Transactions on*, 53(6):1949–1969, June 2007.

[37] Randall Dougherty, Chris Freiling, and Kenneth Zeger. Non-shannon information inequalities in four random variables. *arXiv preprint arXiv:1104.3602*, 2011.

[38] Randall Dougherty, Christopher F. Freiling, and Kenneth Zeger. Network coding and matroid theory. *Proceedings of the IEEE*, 99(3):388–405, 2011.

[39] Randall Dougherty and Kenneth Zeger. Nonreversibility and equivalent constructions of multiple-unicast networks. *Information Theory, IEEE Transactions on*, 52(11):5067–5077, 2006.

[40] S. El Rouayheb, A. Sprintson, and C. Georghiades. On the index coding problem and its relation to network coding and matroid theory. *Information Theory, IEEE Transactions on*, 56(7):3187–3195, July 2010.

[41] Christina Fragouli, Jean-Yves Le Boudec, and Jörg Widmer. Network coding: An instant primer. *SIGCOMM Comput. Commun. Rev.*, 36(1):63–68, January 2006.

[42] Christina Fragouli and Emina Soljanin. Network coding applications. *Foundations and Trends in Networking*, 2(2):135–269, 2007.

[43] EM Gabidulin and Martin Bossert. Algebraic codes for network coding. *Problems of Information Transmission*, 45(4):343–356, 2009.

[44] Maximilien Gadouleau and Søren Riis. Computing without memory. *CoRR*, abs/1111.6026, 2011.

[45] Maximilien Gadouleau and Søren Riis. Graph-theoretical constructions for graph entropy and network coding based communications. *IEEE Transactions on Information Theory*, 57(10):6703–6717, 2011.

[46] Maximilien Gadouleau and Zhiyuan Yan. Packing and covering properties of subspace codes for error control in random linear network coding. *Information Theory, IEEE Transactions on*, 56(5):2097–2108, 2010.

[47] Xuan Guang, Fang-Wei Fu, and Zhen Zhang. Construction of network error correction codes in packet networks. *Information Theory, IEEE Transactions on*, 59(2):1030–1047, 2013.

[48] Nicholas J. A. Harvey and et al. Comparing network coding with multicommodity flow for the k-pairs communication problem, 2004.

[49] Nicholas J. A. Harvey and Robert Kleinberg. Tighter cut-based bounds for k-pairs communication problems, 2005.

[50] Nicholas J.A. Harvey, R. Kleinberg, and A.R. Lehman. On the capacity of information networks. *Information Theory, IEEE Transactions on*, 52(6):2345–2364, June 2006.

[51] N.J.A. Harvey, R. Kleinberg, C. Nair, and Yunnan Wu. A "chicken & egg" network coding problem. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 131–135, June 2007.

[52] Donald G Higman and Charles C Sims. A simple group of order 44,352,000. *Mathematische Zeitschrift*, 105(2):110–113, 1968.

[53] Tracey Ho, M. Medard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong. A random linear network coding approach to multicast. *Information Theory, IEEE Transactions on*, 52(10):4413–4430, Oct 2006.

[54] S. Jaggi, P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain, and L.M.G.M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *Information Theory, IEEE Transactions on*, 51(6):1973–1982, June 2005.

[55] Sidharth Jaggi, Michelle Effros, T Ho, and Muriel Médard. On linear network coding. *Proc. of the 42nd Allerton Conference*, 2004.

[56] K. Jain, Vijay V. Vazirani, R. Yeung, and G. Yuval. On the capacity of multiple unicast sessions in undirected graphs. In *Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on*, pages 563–567, Sept 2005.

[57] Kamal Jain, Mohammad Mahdian, and Mohammad R. Salavatipour. Packing steiner trees. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium*

on Discrete Algorithms, SODA '03, pages 266–274, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[58] Richard M Karp. *Reducibility among combinatorial problems.* Springer, 1972.

[59] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *Networking, IEEE/ACM Transactions on*, 16(3):497–510, June 2008.

[60] Peter Keevash. The existence of designs. *arXiv preprint arXiv:1401.3665*, 2014.

[61] Mikhail H. Klin and Andrew J. Woldar. Dale mesner, higman & sims, and the strongly regular graph with parameters $(100, 22, 0, 6)$. *Bulletin of the Institute of Combinatorics and its Applications*, 63:13–35, 2011.

[62] Ralf Koetter and Frank R Kschischang. Coding for errors and erasures in random network coding. *Information Theory, IEEE Transactions on*, 54(8):3579–3591, 2008.

[63] Ralf Koetter and Muriel Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking (TON)*, 11(5):782–795, 2003.

[64] J Körner. Coding of an information source having ambiguous alphabet and the entropy of graphs. *6th Prague conference on information theory*, pages 411–425, 1973.

[65] Gerhard Kramer and SerapA. Savari. Edge-cut bounds on network coding rates. *Journal of Network and Systems Management*, 14(1):49–67, 2006.

[66] M. Langberg and A. Sprintson. On the hardness of approximating the network coding capacity. In *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, pages 315–319, July 2008.

[67] April Rasala Lehman. *Network Coding.* PhD thesis, Massachusetts Institute of Technology.

[68] 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 '04, pages 142–150, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[69] Radim Lněnička. On the tightness of the zhang-yeung inequality for gaussian vectors. *Communications in information and systems*, 3(1):41–46, 2003.

[70] L LovÕász. A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95 – 98, 1972.

[71] E. Lubetzky and U. Stav. Nonlinear index coding outperforming the linear optimum. *Information Theory, IEEE Transactions on*, 55(8):3544–3551, Aug 2009.

[72] Konstantin Makarychev, Yury Makarychev, Andrei Romashchenko, and Nikolai Vereshchagin. A new class of non-shannon-type inequalities for entropies. *Communications in Information and Systems*, 2:147–165, 2002.

[73] Takahiro Matsuda, Taku Noguchi, and Tetsuya Takine. Survey of network coding and its applications. *IEICE transactions on communications*, 94(3):698–717, 2011.

[74] Ryutaroh Matsumoto. Construction algorithm for network error-correcting codes attaining the singleton bound. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 90(9):1729–1735, 2007.

[75] F. Matús. Infinitely many information inequalities. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 41–44, June 2007.

[76] Muriel Medard, Michelle Effros, David Karger, and Tracey Ho. On coding for non-multicast networks. In *IN PROC. 41ST ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL AND COMPUTING*, 2003.

[77] Dale M. Mesner. PhD thesis, 1956.

[78] James G. Oxley. *Matroid theory.* Oxford: Oxford University Press, 1992.

[79] Etienne Perron Rethnakaran Pulikkoonattu and Suhas Diggavi. X information theoretic inequality prover (xitip), software.

[80] Søren Riis. Linear versus non-linear boolean functions in network flow. *38th Annual Conference on Information Science and Systems (CISS), Princeton, NJ*, 2004.

[81] Søren Riis. Graph entropy, network coding and guessing games. *CoRR*, abs/0711.4175, 2007.

[82] Søren Riis. Information flows, graphs and their guessing numbers. *Electr. J. Comb.*, 14(1), 2007.

[83] Søren Riis. Reversible and irreversible information networks. *IEEE Transactions on Information Theory*, 53(11):4339–4349, 2007.

[84] Michele Sanna and Ebroul Izquierdo. A survey of linear network coding and network error correction code constructions and algorithms. *International Journal of Digital Multimedia Broadcasting*, 2011, 2011.

[85] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[86] Danilo Silva and Frank R Kschischang. On metrics for error correction in network coding. *Information Theory, IEEE Transactions on*, 55(12):5479–5490, 2009.

[87] Lihua Song, R.W. Yeung, and Ning Cai. Zero-error network coding for acyclic networks. *Information Theory, IEEE Transactions on*, 49(12):3129–3139, Dec 2003.

[88] Y. Sun. *Network coding and graph entropy.* PhD thesis, Queen Mary, University of London.

[89] Leslie G Valiant. Why is boolean complexity theory difficult? In *Poceedings of the London Mathematical Society symposium on Boolean function complexity*, pages 84–94. Cambridge University Press, 1992.

[90] Taoyang Wu, Peter J. Cameron, and Søren Riis. On the guessing number of shift graphs. *J. Discrete Algorithms*, 7(2):220–226, 2009.

[91] Weidong Xu, Jia Wang, and Jun Sun. A projection method for derivation of non-shannon-type information inequalities. In *2008 IEEE International Symposium on Information Theory*, pages 2116–2120, 2008.

[92] Shenghao Yang, Raymond W Yeung, and Chi Kin Ngai. Refined coding bounds and code constructions for coherent network error correction. *Information Theory, IEEE Transactions on*, 57(3):1409–1424, 2011.

[93] Shuo yen Robert Li, Senior Member, Raymond W. Yeung, and Ning Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49:371–381, 2003.

[94] R. W. Yeung and Y. O. Yan. Information theory inequality prover (itip), software.

[95] Raymond W. Yeung. *A First Course in Information Theory (Information Technology: Transmission, Processing and Storage)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[96] Raymond W Yeung, Ning Cai, et al. Network error correction, i: Basic concepts and upper bounds. *Communications in Information & Systems*, 6(1):19–35, 2006.

[97] R.W. Yeung and Z. Zhang. Distributed source coding for satellite communications. *Information Theory, IEEE Transactions on*, 45(4):1111–1120, May 1999.

[98] Z. Zhang and R.W. Yeung. On characterization of entropy function via information inequalities. *Information Theory, IEEE Transactions on*, 44(4):1440–1452, Jul 1998.

[99] Zhen Zhang. On a new non-shannon type information inequality. *Communications in Information and Systems*, 3(1), 2003.

[100] Zhen Zhang. Linear network error correction codes in packet networks. *Information Theory, IEEE Transactions on*, 54(1):209–218, 2008.

[101] Zhen Zhang. Theory and applications of network error correction coding. *Proceedings of the IEEE*, 99(3):406–420, 2011.

[102] Zhen Zhang and R.W. Yeung. A non-shannon-type conditional inequality of information quantities. *Information Theory, IEEE Transactions on*, 43(6):1982–1986, Nov 1997.