5-2018

Graphs with few spanning substructures

Jessica De Silva
University of Nebraska - Lincoln, jessica.desilva@huskers.unl.edu

GRAPHS WITH FEW SPANNING SUBSTRUCTURES

by

Jessica C. De Silva

A DISSERTATION

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Doctor of Philosophy

Major: Mathematics

Under the Supervision of Professor Jamie Radcliffe

Lincoln, Nebraska

August, 2018

GRAPHS WITH FEW SPANNING SUBSTRUCTURES

Jessica C. De Silva, Ph.D.

University of Nebraska, 2018

Advisor: Professor Jamie Radcliffe

In this thesis, we investigate a number of problems related to spanning substructures of graphs. The first few chapters consider extremal problems related to the number of forest-like structures of a graph. We prove that one can find a threshold graph which contains the minimum number of spanning pseudoforests, as well as rooted spanning forests, amongst all graphs on n vertices and e edges. This has left the open question of exactly which threshold graphs have the minimum number of these spanning substructures. We make progress towards this question in particular cases of spanning pseudoforests.

The final chapter takes on a different flavor—we determine the complexity of a problem related to Hamilton cycles in hypergraphs. Dirac's theorem states that graphs with minimum degree at least half the size of the vertex set are guaranteed to have a Hamilton cycle. In 1993, Karpińksi, Dahlhaus, and Hajnal proved that for any $c < \frac{1}{2}$, the problem of determining whether a graph with minimum degree at least cn has a Hamilton cycle is NP-complete. The analogous problem in hypergraphs, for both a Dirac-type condition and complexity, are just as interesting. We prove that for classes of hypergraphs with certain minimum vertex degree conditions, the problem of determining whether or not they contain an ℓ-Hamilton cycle is NP-complete.

ACKNOWLEDGMENTS

First, thank you to Jamie and Michael for being my advisors/mentors throughout my graduate school experience. Jamie always allowed me to let my interests wander, and taught me that one should always do small examples *first*, before you attempt to prove a conjecture. Michael, thank you for being my rockstar and believing in me the entire way more than anyone else I know. I'd also like to thank my committee: Susan Hermiller, Judy Walker, Christine Kelley, and Vinod Variyam. I hadn't even realized until the end that my committee was more than half outstanding female mathematicians who I continue to look up to. Also, thank you to the staff in UNL Graduate Studies and to my other mentors: Matthias Beck, Aris Winger, David Offner, and Bill Hrusa.

I'd also like to thank my Mom, Dad, and brother for supporting me from many miles away during graduate school. Thank you to the various officemates and amazing friends that I've met while in graduate school, including: Jessalyn Bolkema, Lauren Keough, Rachel Kirsch, Carolyn Mayer, Shimin Deng, Lara Ismert, Jessie Jamieson, Stephanie Prahl, Laura White, Allison Beemer, Josh Pollitz, Corbin Groothius, JD Nir, Nick Packauskas, Mohsen Gheibi, my GRWC Wolf Pack, UNL Graduate Student Ambassadors, and Carnegie Mellon friends.

I'd especially like to thank all of the programs I have participated in over the years, especially the McNair Scholars Program, GRWC, IAS Women and Mathematics Program, and MSRI-UP. I am thankful for the faculty in the Department of Mathematics at California State University, Stanislaus, who have formed the foundation of who I am as a mathematician and who now soon be my future colleagues.

From the bottom of my heart, Paulo deserves the biggest thank you for being my rock, my love, and the source of all of my laughter for the past five years in Lincoln.

GRANT INFORMATION

The author was partially supported by the National Science Foundation Graduate Research Fellowship Program NSF DGE-1041000.

Table of Contents

List of Figures

Chapter 1

Introduction

In this dissertation, we investigate several problems related to extremal graph and hypergraph theory. A well-known result in this area is that of Turán, who determined the maximum number of edges in a graph which does not contain a complete subgraph of a given size [31]. Questions in extremal graph theory are often in the same spirit, asking what is the maximum or minimum value of a combinatorial property of a graph subject to a given constraint. The result of Turán maximizes the number of edges while fixing the number of vertices and the property 'Does not contain a K_{r+1}'. Throughout, we aim to develop more information about graphs with a number of fixed properties (e.g., size, order, minimum degree) which minimize the number of certain spanning substructures.

All graphs considered here are finite, undirected, and simple. We use standard graph theory notation (see [3]) and a glossary of terms can be found in Chapter 6. Given a graph G with vertex set $V(G)$ and edge set $E(G)$, let $n(G) := |V(G)|$ and $e(G) := |E(G)|$. For $S \subseteq V(G)$, define $G[S]$ as the subgraph of G induced by the vertex set S. We will let $G \backslash S$ denote the graph $G[V(G) \backslash S]$. The closed neighborhood of $v \in V(G)$ will be written as $N[v] = N(v) \cup \{v\}$. The disjoint union of two graphs

G and H will be written as $G \cup H$. For $H \subseteq G$ and $v \in V(H)$, let

$$N_H(v) = \{u \in V(H) : uv \in E(H) \subseteq E(G)\}.$$

For $H \subseteq G$, we call H a *spanning subgraph* of G if $V(H) = V(G)$.

1.1 Minimizing Spanning Forest-like Structures

In recent years, many problems in extremal graph theory have been considered concerning maximizing or minimizing the number of substructures in a graph while fixing the number of vertices and edges. We will denote the class of graphs with n vertices and e edges by $\mathcal{G}_{n,e}$. One aspect of these types of problems is to characterize the extremal graphs, that is, the graphs which attain the bounds. In many problems of this type, it has been shown that the maximum or minimum value of the parameter in question is attained within the class of threshold graphs.

Definition 1.1.1. A graph $G = (V, E)$ is called a *threshold graph* when there exist non-negative reals w_v, $v \in V$ and t such that for $U \subseteq V$,

$$\sum_{v \in U} w_v \leq t$$

if and only if U is an independent set.

The name *threshold* comes from the idea that taking U to be a pair of vertices, t acts as a threshold for whether or not those two vertices are adjacent. There are many equivalent definitions of threshold graphs, two of which will be used in the following chapters.

Theorem 1.1.1 ([10, 12])**.** *For a graph $G = (V, E)$, the following are equivalent:*

1. *G is a threshold graph;*

2. *For every pair of vertices $x, y \in V$,*

$$N(x)\backslash\{y\} \subseteq N(y)\backslash\{x\} \text{ or } N(y)\backslash\{x\} \subseteq N(x)\backslash\{y\};$$

3. *G can be constructed from a single vertex by adding vertices one at a time that are either isolated or dominating.*

Chapters 2 and 4 will utilize Part 3.2 of Theorem 1.1.1 and in Chapter 3 we will make use of Part 3.

Example 1.1.1. *Note that using Part 3.2 of Theorem 1.1.1, we can show that the graph G shown in Figure 1.1.1 is threshold. We can check that $N(v_0) = N(v_1)$ and $N(v_0)\backslash\{v_2\} = \emptyset \subseteq N(v_2)\backslash\{v_0\}$ and similarly for v_0 and v_2.*

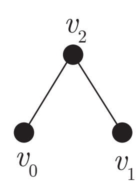


Figure 1.1: A threshold graph G.

Using Part 3 of Theorem 1.1.1, we can create a new threshold graph from G by sequentially adding isolated or dominating vertices. The threshold graph in Figure 1.1.1 is derived from G by first adding v_3 as an isolated vertex, and then adding v_4 as a dominating vertex.

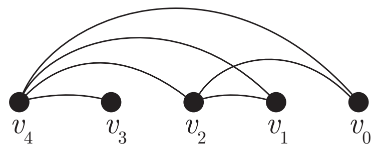


Figure 1.2: A threshold graph constructed by adding an isolated and then a dominating vertex to the threshold graph G.

In particular, threshold graphs have been shown to contain the minimizer of the number of certain spanning substructures such as spanning trees and perfect matchings ([34],[24]). Motivated by these results, Chapters 2 and 4 prove that threshold graphs also contain the minimizer of the number of spanning pseudoforests and rooted spanning forests, respectively, of a fixed size. In Chapter 3, we dive deeper into the specific case of size 5 pseudoforests and determine properties of threshold graphs which contain the maximum number of size 5 pseudoforests.

1.2 Hamilton Cycles in Hypergraphs

In the final chapter, we study a problem in extremal hypergraph theory from the perspective of computational complexity theory. Generally speaking, in computational complexity theory we ask how fast can an algorithm solve a given problem relative to the length of the input. A problem is in P if it can be solved in polynomial time, while a problem is in NP if solutions to the problem can be verified in polynomial time. While it is clear that problems in P are contained in NP, one of the most famous unsolved problems of mathematics is whether or not there exist problems in NP which are not in P. We call a problem NP-complete if it is in NP and all other problems in NP can be reduced to an instance of it in polynomial time. A polynomial time algorithm for an NP-complete problem would then imply that NP is contained in P, and so NP-complete problems are of particular interest in this area. A general outline of how one shows that a given decision problem is NP-complete is discussed in Chapter 5.1.

Let's assume that the problem of determining whether or not a graph has property Q is NP-complete. Suppose there exists an extremal graph theory result which states that for some graph parameter $\lambda(\cdot)$, k is minimum such that every graph G with $\lambda(G) \geq k$ has property Q. If we consider the decision problem of determining whether or not a graph has property Q, then the extremal result implies that this problem restricted to graphs G with $\lambda(G) \geq k$ can be solved in polynomial time by always returning 'YES'. Since the problem with no restrictions on λ is NP-complete, it is interesting to determine the threshold at which the problem becomes hard. Thus, for some $i < k$, one may ask what is the complexity of deciding whether or not a graph G with $\lambda(G) \geq i$ has property Q.

The problem we will investigate is a hypergraph analogue to Dirac's theorem. A *hypergraph* with vertex set V is a generalization of a graph in which the edge set, E, is any collection of subsets of V. Recall that a Hamilton cycle in a graph G is a spanning subgraph of G which is a cycle. Dirac's theorem states that any n-vertex graph with minimum degree at least $n/2$ is guaranteed to contain a Hamilton cycle. The problem of determining whether or not a graph contains a Hamilton cycle is one of Karp's celebrated 21 NP-complete problems [20]. On the other hand, Dirac's theorem implies that restricting this problem to n-vertex graphs with minimum degree at least $n/2$ can be done in polynomial time by simply returning 'YES'. In 1993, Dahlhaus, Hajnal, and Karpiński studied the complexity of the problem of determining whether or not a graph with minimum degree below $n/2$ contains a Hamilton cycle. Interestingly enough, they showed that restricting this problem to graphs *anywhere* below this minimum degree threshold is NP-complete. Let $\mathrm{HAM}(2, c)$ be the problem of determining whether or not an n-vertex graph with minimum degree at least cn contains a Hamilton cycle.

Theorem 1.2.1 ([8])**.** *For all $c < \frac{1}{2}$, HAM$(2, c)$ is NP-complete.*

The problem of generalizing Dirac's theorem to the hypergraph setting has been of increased interest in recent years. In particular we will be considering k-uniform hypergraphs, that is a hypergraph whose edges are all of size k. There are many ways in which one may define a cycle in a hypergraph—we will focused the notion of an ℓ-cycle. A k-uniform hypergraph, also called a k-graph, C is an *ℓ-cycle* if its vertices can be cyclically ordered in such a way that each edge of C consists of k consecutive vertices and each pair of consecutive edges overlaps in exactly ℓ vertices. Thus a Hamilton ℓ-cycle of a k-uniform hypergraph H is a subhypergraph of G that is an ℓ-cycle and contains all of the vertices of H. For $n \geq 1$, let $[n]$ denote the set of integers $\{1, \ldots, n\}$.

Example 1.2.1. *The following three hypergraphs are all on the vertex set* $[6]$.

(i) $E(H_1) = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}\}$

H_1 *is a* 1*-cycle in a* 2*-graph (i.e., a graph).*

(ii) $E(H_2) = \{\{1, 2, 3\}, \{3, 4, 5\}, \{5, 6, 1\}\}$

H_2 *is a* 1*-cycle in a* 3*-graph.*

(iii) $E(H_3) = \{\{1, 2, 3\}, \{2, 3, 4\}, \{3, 4, 5\}, \{4, 5, 6\}, \{5, 6, 1\}, \{6, 1, 2\}\}$

H_3 *is a* 2*-cycle in a* 3*-graph.*

The notion of minimum degree also has various interpretations. For a k-uniform hypergraph H let

$$\delta_i(H) = \min_{S \subseteq V(H),\, |S| = i} |\{e \in E(H) \,:\, S \subseteq e\}|.$$

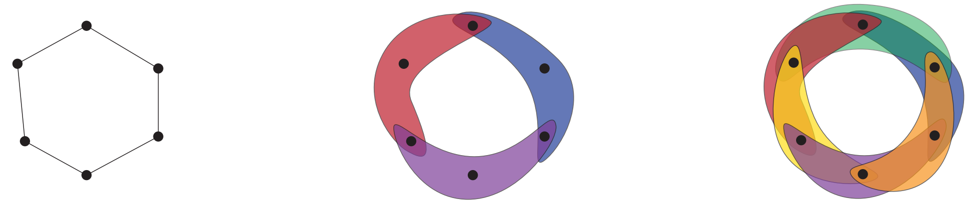


Figure 1.3: Hypergraphs H_1, H_2, and H_3 respectively.

The special cases of $i = k-1$ and $i = 1$ are often referred to as the minimum *codegree* and the minimum *vertex degree* respectively. The following theorem is an approximate hypergraph analogue to Dirac's theorem. Various cases of the theorem were proved by Rödl, Ruciński, and Szemerédi [32, 33], Kühn and Osthus [28], Keevash, Kühn, Mycroft, and Osthus [22], Hàn and Schacht [15], Kühn, Mycroft, and Osthus [27].

Theorem 1.2.2 ([32, 33, 28, 22, 15, 27])**.** *For any $k \geq 3$, $1 \leq \ell \leq k-1$, and $\eta > 0$, there exists n_0 such that if $n \geq n_0$ is divisible by $k-\ell$ and H is a k-uniform hypergraph on n vertices with*

$$\delta_{k-1}(G) \geq \begin{cases} \left(\frac{1}{2} + \eta\right) n, & \text{if } k-\ell \text{ divides } k, \\ \left(\frac{1}{\lceil \frac{k}{k-\ell} \rceil (k-\ell)} + \eta\right) n, & \text{otherwise,} \end{cases}$$

then H contains a Hamilton ℓ-cycle.

The corresponding complexity problem has also been of recent interest. In [21], Karpiński, Ruciński, and Szymańska show that the problem of determining a Hamilton $(k-1)$-cycle in a k-uniform hypergraph with minimum codegree $\left(\frac{1}{k} - \epsilon\right) n$ is NP-complete. This leaves a gap between $\frac{n}{k}$ and $\frac{n}{2}$ for the complexity result and the Dirac threshold given in Theorem 1.2.2. Very recently, Garbe and Mycroft closed this gap with the following result.

Theorem 1.2.3 ([11]). *For any $k \geq 3$, there exists a constant C such that the Hamilton $(k-1)$-cycle decision problem remains NP-complete when restricted to n-vertex k-uniform hypergraphs H with $\delta_{k-1}(H) \geq \frac{n}{2} - C$.*

Here, we focus on Dirac-type conditions regarding the minimum vertex degree, as opposed to minimum codegree. There are very few cases in which the analogue to Dirac's theorem has been proven for vertex degree. Specifically, Han and Zhao in [16] determine lower bounds for what the Dirac-type condition should be but it is not proven to be tight. Let $h_\ell(k,n)$ denote the smallest integer h such that every n-vertex k-uniform hypergraph H satisfying $\delta_1(H) \geq h$ contains a Hamilton ℓ-cycle. In 2016, Han and Zhao proved the following bounds on h.

Theorem 1.2.4 ([16]). *Let $1 \leq \ell \leq k-1$ and $t = k-1$, then*

$$h_\ell(k,n) \geq \begin{cases} \left(1 - \binom{t}{\lfloor \frac{t}{2} \rfloor} \frac{\lceil \frac{t}{2} \rceil^{\lceil \frac{t}{2} \rceil} (\lfloor \frac{t}{2} \rfloor + 1)^{\lfloor \frac{t}{2} \rfloor}}{(t+1)^t} + o(1)\right) \binom{n}{t}, & \text{if } \ell = k-1, \\ (1 - b_{t,k-\ell} 2^{-t} + o(1)) \binom{n}{t}, & \text{otherwise,} \end{cases}$$

where $b_{t,k-\ell}$ equals the largest sum of $k-\ell$ binomial coefficients taken from $\{\binom{t}{0}, \ldots, \binom{t}{t}\}$.

We prove that when restricted to graphs with certain vertex degree thresholds, the problem of determining Hamiltonicity with respect to 1-cycles and 2-cycles is NP-complete. As in the case of Karpiński, Ruciński, and Szymańska, our results leave us with a hardness gap for which the hardness of the problem remains unknown.

Chapter 2

Minimizing Spanning Pseudoforests

In this chapter we answer an extremal question concerning forest-like structures called pseudoforests. We are particularly interested in determining which graphs in $\mathcal{G}_{n,e}$, the set of all graphs on n vertices and e edges, have the minimum number of spanning pseudoforests. The main result is that we can find a threshold graph which minimizes the number of spanning pseudoforests with a fixed number of edges.

2.1 Definitions and Results

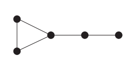
Recall that a forest is a graph such that every (connected) component is a tree. A family of graphs that is closely related to trees is the family of unicyclic graphs. A unicyclic graph H is *almost* a tree since it can be written as $H = T + e$ for some tree T and some edge $e \notin E(T)$. Moreover, such a graph contains exactly one cycle. The following notion of a pseudoforest, attributed to Dantzig [9], defines a forest-like graph that allows for components to be trees or *almost* trees.

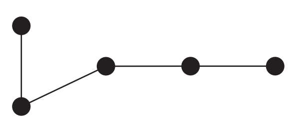
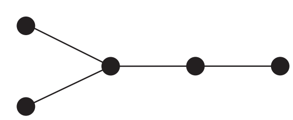
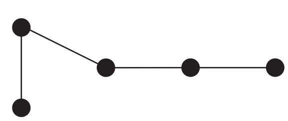
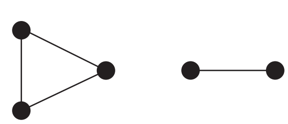
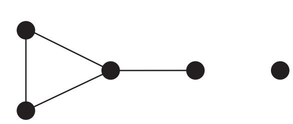
Definition 2.1.1. A *pseudoforest* is a graph such that every component is either a tree or unicyclic.

We are particularly interested in pseudoforest subgraphs of some host graph G. As is often used for trees and forests, we can define a notion of what it means for a pseudoforest to be *spanning* in G.

Definition 2.1.2. The *spanning pseudoforests* of a graph G are the pseudoforest subgraphs which contain all the vertices of G.

One way to partition the set of all spanning pseudoforests of G is according to the number of edges. Let $\mathcal{P}_\ell(G)$ denote the set of spanning pseudoforests of G with exactly ℓ edges. In the case of a spanning forest, the number of edges determines the number of components. In particular, a forest of order n with $n-k$ edges has exactly k components. When we consider the set of spanning pseudoforests of G with exactly ℓ edges, this is *not* necessarily the same as the set of spanning pseudoforests of G with exactly $n-\ell$ components. In fact, these two sets of graphs are not necessarily comparable with respect to containment as illustrated in Example 2.1.1.

Example 2.1.1.
The following table contains the elements of $\mathcal{P}_4\Bigg($  $\Bigg)$.

Spanning Trees		
		
Other		

The original graph itself is a 1-component spanning pseudoforest although it does not lie in $\mathcal{P}_4(G) = \mathcal{P}_{n-1}(G)$.

For a clearer picture of of this partition, we will determine if $\mathcal{P}_\ell(G)$ is nontrivial for a given ℓ. The set $\mathcal{P}_0(G)$ is exactly the set containing the empty graph on the vertex set $V(G)$. An interesting question one may then ask is "What is the maximum ℓ such that $\mathcal{P}_\ell(G) \neq \emptyset$?" Certainly we should only consider $\ell \leq e(G)$ and note that for $\ell \geq 1$, $\mathcal{P}_\ell(G) \neq \emptyset$ implies $\mathcal{P}_{\ell-1}(G) \neq \emptyset$ since the family of pseudoforests is closed under edge deletion.

Lemma 2.1.1. *Let G be a graph on n vertices, then*

$$\max\{\ell : \mathcal{P}_\ell(G) \neq \emptyset\} = n - c_t(G)$$

where $c_t(\cdot)$ counts the number of tree components of G.

Proof. The result is clear if G is a tree since G is itself a pseudoforest and $e(G) = n - c_t(G)$. Now suppose G is connected and not a tree, then it contains some spanning tree T. T is acyclic and G contains at least one cycle, hence there exists some edge $e \notin E(T)$ that lies on a cycle in G. Then $T + e$ is a spanning pseudoforest of G with exactly $e(T) + 1 = n$ edges, therefore $\mathcal{P}_n(G) \neq \emptyset$.

Let P be a spanning pseudoforest of G. Note that P can be derived from some spanning forest of G by adding at most one edge per component. Fix one such spanning forest, F, and let $c(F)$ denote the number of components of F. Recall that this implies F has $n - c(F)$ edges, therefore

$$e(P) \leq e(F) + c(F) = n.$$

Finally suppose G is disconnected and let $G_1, \ldots, G_k$ be the components of G. A spanning pseudoforest P of G can be written as $P = \cup_{i=1}^k P_i$ where P_i is a spanning pseudoforest of G_i for each i. From above we have $\mathcal{P}_\ell(G_i)$ is nonempty if $\ell \leq n(G_i) - 1$

when G_i is a tree and $\ell \leq n(G_i)$ otherwise. Therefore $\mathcal{P}_\ell(G)$ is nonempty if

$$\ell \leq \sum_{G_i \text{ is a tree}} (n(G_i) - 1) + \sum_{G_i \text{ is not a tree}} n(G_i) = n - c_T(G).$$

$\square$

We are interested in determining which graphs with a fixed number of vertices and edges minimize $|\mathcal{P}_\ell|$. The main result of this chapter is the following.

Theorem 2.1.1. *For any n and e and any $\ell \in \{0, \ldots, n\}$, there exists a threshold graph $H \in \mathcal{G}_{n,e}$ such that*

$$|\mathcal{P}_\ell(H)| = \min_{G \in \mathcal{G}_{n,e}} |\mathcal{P}_\ell(G)|.$$

We will postpone the proof of this theorem to Section 2.5. Our main tool will be a graph operation referred to as compression, or the Kelmans transformation (see, e.g., [5]).

Definition 2.1.3. Let x and y be two vertices of a graph G. The *compression of G from x to y*, denoted $G_{x \to y}$, is the graph obtained from G by removing all edges between x and $N(x) \backslash N[y]$ and adding all edges between y and $N(x) \backslash N[y]$.

The important outcome of this size-preserving operation is that it alters the neighborhoods of two vertices so that one neighborhood contains the other. In light of this, it is well-known that threshold graphs can be obtained from any graph via a series of compressions (see Corollary 2.2.2). We also remark that compressing G from x to y will result in the same graph, under isomorphism, as compressing from y to x. That is, $G_{x \to y} \cong G_{y \to x}$.

The following result of Satyanarayana, Schoppmann, and Suffel [34] shows that compression decreases the number of spanning trees.

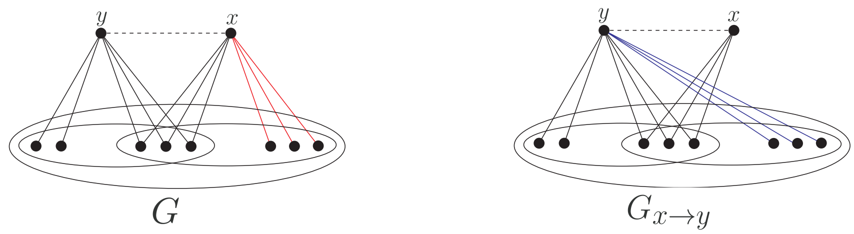


Figure 2.1: Compression of G from x to y.

Theorem 2.1.2 ([34]). *For $x, y \in V(G)$,*

$$|\mathcal{T}(G_{x\to y})| \leq |\mathcal{T}(G)|.$$

where $\mathcal{T}(\cdot)$ denotes the number of spanning trees.

We take a similar approach by showing that compression also decreases the number of spanning pseudoforests with a fixed number of edges. The proof of Theorem 2.1.1 serves to be interesting as it utilizes Proposition 2.3.1 which is a more precise result of Theorem 2.1.2.

2.2 Notation and Preliminary Lemmas

We will first set up notation and note some properties of compression that will be used throughout. Fix $G \in \mathcal{G}_{n,e}$ and $x, y \in V(G)$. Since the following subsets of neighborhoods of x and y will be referred to often, we will denote them as

$$A_{x\overline{y}} = N_G(x) \backslash N_G(y)$$
$$A_{\overline{x}y} = N_G(y) \backslash N_G(x)$$
$$A_{xy} = N_G(x) \cap N_G(y).$$

We define two edge replacement functions c, d. Let $c : E(G) \to E(G_{x\to y})$ and d :

$E(G_{x\to y}) \to E(G)$ be given by

$$c(e) = \begin{cases} e\Delta\{x,y\} & \text{if } e \notin E(G_{x\to y}) \\ e & \text{otherwise} \end{cases} \qquad \text{and} \qquad d(e) = \begin{cases} e\Delta\{x,y\} & \text{if } e \notin E(G) \\ e & \text{otherwise.} \end{cases}$$

Note that Δ is the standard symmetric difference operator. We remark that these two functions do indeed depend on G, x, y, that is, $c = c_{G,x,y}$ and $d = d_{G,x,y}$. For simplicity of notation, we will proceed by suppressing the dependence on G, x, y. For $H \subseteq G$, let $c(H)$ be the graph with vertex set $V(H)$ and edge set $\{c(e) : e \in E(H)\}$ and similarly define $d(H)$. Note that $c(G) = G_{x\to y}$ and $d(G_{x\to y}) = G$, henceforth we will denote $G_{x\to y}$ as $c(G)$. We will refer to c as *compression* and d as *decompression*. As the names imply, the following lemma states that c and d are indeed inverse functions of one another.

Lemma 2.2.1. *Let $H \subseteq G$ and $H' \subseteq c(G)$. Then $d(c(H)) = H$ and $c(d(H')) = H'$.*

Proof. By definition of c and d, $V(d(c(H))) = V(c(H)) = V(H)$ and

$$E(d(c(H))) = \{d(e) : e \in E(c(H))\} = \{d(c(e)) : e \in E(H)\}.$$

Thus, it suffices to show $d(c(e)) = e$ for all $e \in E(H)$. Note

$$d(c(e)) = \begin{cases} c(e)\Delta\{x, y\} & \text{if } c(e) \notin E(G) \\ c(e) & \text{otherwise} \end{cases}$$

$$= \begin{cases} (e\Delta\{x, y\})\Delta\{x, y\} & \text{if } e \notin E(G_{x\to y}) \text{ and } (e\Delta\{x, y\}) \notin E(G) \\ e\Delta\{x, y\} & \text{if } e \in E(G_{x\to y}) \text{ and } e \notin E(G) \\ e\Delta\{x, y\} & \text{if } e \notin E(G_{x\to y}) \text{ and } (e\Delta\{x, y\}) \in E(G) \\ e & \text{if } e \in E(G_{x\to y}) \text{ and } e \in E(G) \end{cases}$$

and since $e \in E(H) \subseteq E(G)$, e does not fall into cases 2 nor 3. Thus,

$$d(c(e)) = \begin{cases} (e\Delta\{x, y\})\Delta\{x, y\} & \text{if } e \notin E(G_{x\to y}) \text{ and } (e\Delta\{x, y\}) \notin E(G) \\ e & \text{if } e \in E(G_{x\to y}) \text{ and } e \in E(G) \end{cases}$$

$$= e$$

which concludes the proof for H. Similarly, the result follows for H'. □

Corollary 2.2.1. *Suppose $H_1, H_2 \subseteq G$ and $H_1', H_2' \subseteq c(G)$. If $H_1 \neq H_2$ and $H_1' \neq H_2'$, then $c(H_1) \neq c(H_2)$ and $d(H_1') \neq d(H_2')$.*

Proof. This follows directly from Lemma 2.2.1. □

An important property about compression and decompression is that they both behave nicely under induced subgraphs. For example, any edge which does not contain x nor y is fixed by both c and d.

Lemma 2.2.2. *For $S \subseteq V(G)$,*

1. $c(G)[S] = c(G[S])$.

2. $d(G)[S] = d(G[S])$.

3. *If $\{x, y\} \subseteq S$, then $c(G\backslash S) = d(G\backslash S) = G\backslash S$.*

Proof. 1. Note that $V(c(G)[S]) = S = V(c(G[S]))$. We also have

$$\begin{aligned} E(c(G)[S]) &= \{e \in E(c(G)) : e \subseteq S\} \\ &= \{c(e) : e \in E(G) \text{ and } e \subseteq S\} \\ &= \{c(e) : e \in E(G[S])\} \\ &= E(c(G[S])) \end{aligned}$$

since c only depends on G, x, y.

2. Follows similarly from the proof of 1.

3. Since $\{x, y\} \subseteq S$, we have $G\backslash S \subseteq G$ and $G\backslash S \subseteq c(G)$. Thus by definition of c and d, for $e \in E(G\backslash S)$ we have $c(e) = d(e) = e$. □

It is also handy to note how compression and decompression affects connectivity of subgraphs (of both G and $c(G)$). For $H \subseteq G$, it is clear that $N_H(\{x, y\}) = N_{c(H)}(\{x, y\})$, but the following lemma states that vertices in a component with either x or y remain in a component with either x or y under compression and decompression. Given $v \in V(H)$, let H_v be the connected component of H containing v.

Lemma 2.2.3. *Let $S \subseteq E(G)$ such that for $e \in S$ either $x \in e$ or $y \in e$. Define a function q on $E(G)$ by*

$$q(e) = \begin{cases} e\Delta\{x, y\} & \text{if } e \in S \\ e & \text{otherwise.} \end{cases}$$

Then for $H \subseteq G$, $q(H_x \cup H_y) = q(H_x \cup H_y)_x \cup q(H_x \cup H_y)_y$. A similar statement holds for $S \subseteq E(c(G))$ and $H \subseteq c(G)$.

Proof. Let $H \subseteq G$. First suppose $z \in V(H_y)$ and let P_{zy} be the shortest path in H_y from z to y. If $q(P_{zy}) \neq P_{zy}$, then there exists some $e \in S$ such that $e \in E(P_{zy})$. If $P_{zy} = zw_1 \cdots w_k y$ with $w_i \neq x$ for $i = 1, \ldots, k$ and $w_k y \in S$, then z is connected to x via the path $q(P_{zy}) = zw_1 \cdots w_k x$. Otherwise, $x \in V(P_{zy})$ and so P_{zy} contains the subgraph $P'_{zy} = zw_1 \cdots w_j x$ with $w_i \neq x$ for $i = 1, \ldots, j$ and $w_j x \in S$. Then z is in $q(H_x \cup H_y)_y$ since it is connected to y via the path $q(P'_{zy}) = zw_1 \cdots w_j y$.

Now suppose $z \in V(H_x)\backslash V(H_y)$ and let P_{zx} be the shortest path in H_x from z to x. If $q(P_{zx}) \neq P_{zx}$, then $P_{zx} = zv_1 \cdots v_k x$ with $v_i \neq y$ for $i = 1, \ldots, k$ and $v_k x \in S$. Then z is in $c(H_x \cup H_y)_y$ since it is connected to y via the path $c(P_{zx}) = zv_1 \cdots v_k y$. Thus, $c(H_x \cup H_y) \subseteq c(H_x \cup H_y)_x \cup c(H_x \cup H_y)_y$. Applying the same proof with $S' = \{e\Delta\{x, y\} \,|\, e \in S\}$ implies that equality holds. $\square$

Perhaps most importantly, any graph can be transformed into a threshold graph by repeated application of compression. Note that if G is threshold, then for all $x, y \in V(G)$, $G_{x \to y} \cong G$. Corollary 2.2.2 given below appears in [24], but we will include the proof for completeness. To prove this, we will use Lemma 2.2.4 which states that the sum of the squares of the degrees, denoted by

$$d_2(G) = \sum_{v \in V(G)} d(v)^2,$$

strictly increases under a non-trivial compression.

Lemma 2.2.4 ([6])**.** *If $x, y \in V(H)$ such that $N(x) \backslash \{y\} \not\subseteq N(y)$ and $N(y) \backslash \{x\} \not\subseteq N(x)$, then $d_2(H) < d_2(H_{x \to y})$.*

Corollary 2.2.2 ([24])**.** *Let $\mathcal{G}$ be a collection of graphs such that if $H \in \mathcal{G}$ and $x, y \in V(H)$, then $H_{x \to y} \in \mathcal{G}$. If $H \in \mathcal{G}$ satisfies*

$$d_2(H) = \max_{H' \in \mathcal{G}} d_2(H'),$$

then H is threshold.

Proof. Suppose otherwise, then there exists $u, v \in V(H)$ such that $N(u) \backslash \{v\}$ and $N(v) \backslash \{u\}$ are incomparable. Then $H_{u \to v} \in \mathcal{G}$ and by Lemma 2.2.4, $d_2(H) < d_2(H_{u \to v})$. This contradicts the assumption that H attains the maximum value of d_2 and therefore H is threshold. $\square$

2.3 Compression Acting on Spanning Trees

In what follows we will use the previous setup where x and y are fixed vertices of G and we are considering the compression $c : E(G) \to E(G_{x \to y})$. In order to prove Theorem 2.1.1, we will need a more precise result of Theorem 2.1.2. Given $T \in \mathcal{T}(G)$, note that $c(T)$ is either a tree or unicyclic. If $c(T)$ is unicyclic, let C_T denote its unique cycle and note that by definition of c, C_T contains y. Let $\mathcal{T}^*(G)$ denote the set of spanning trees T whose image under compression is unicyclic and C_T contains an edge of the form yw for $w \in A_{\overline{x}y}$.

Proposition 2.3.1. $|\mathcal{T}(G)| - |\mathcal{T}(c(G))| = |\mathcal{T}^*(G)|$.

Proof. Let

$$\begin{aligned}
\mathcal{T}_1(G) &:= \{T \in \mathcal{T}(G) \,|\, c(T) \in \mathcal{T}(c(G))\} \\
\mathcal{T}_2(G) &:= \mathcal{T}(G) \backslash \mathcal{T}_1(G) = \{T \in \mathcal{T}(G) \,|\, c(T) \notin \mathcal{T}(c(G))\} \\
\mathcal{T}_1(c(G)) &:= \{T \in \mathcal{T}(c(G)) \,|\, d(T) \in \mathcal{T}(G)\} \\
\mathcal{T}_2(c(G)) &:= \mathcal{T}(c(G)) \backslash \mathcal{T}_1(c(G))) = \{T \in \mathcal{T}(c(G)) \,|\, d(T) \notin \mathcal{T}(G)\}.
\end{aligned}$$

By Corollary 2.2.1,

$$|\mathcal{T}_1(G)| = |\mathcal{T}_1(c(G))|.$$

For $T \in \mathcal{T}_2(G)$, since $c(T)$ contains $n-1$ edges and $c(T) \notin \mathcal{T}(c(G))$ then $c(T)$ is not connected. Lemma 2.2.3 implies $c(T)$ contains exactly two components $c(T)_x$ and $c(T)_y$ such that $c(T)_y$ contains exactly one cycle. In particular, the cycle contains two edges yz and yw for $w \in A_{x\overline{y}}$ and $z \in A_{\overline{x}y} \cup A_{xy}$. We will partition $\mathcal{T}_2(G)$ in the following way:

$$\begin{aligned}
\mathcal{T}^{**}(G) &:= \{T \in \mathcal{T}_2(G) \,|\, yz \in E(C_T) \text{ where } z \in A_{xy}\} \\
\mathcal{T}^*(G) &:= \mathcal{T}_2(G) \backslash \mathcal{T}^{**}(G).
\end{aligned}$$

It remains to prove

$$|\mathcal{T}^{**}(G)| = |\mathcal{T}_2(c(G))|.$$

For $T \in \mathcal{T}^{**}(G)$, let $w_T \in A_{xy}$ and $z_T \in A_{x\overline{y}}$ such that $yw_T, yz_T \in E(C_T)$. Consider the following edge replacement function $s : E(T) \to E(c(G))$ given by

$$s(e) = \begin{cases} e\Delta\{x,y\} & \text{if } e \notin E(c(G)) \\ e\Delta\{x,y\} & \text{if } e = yw_T \\ e & \text{otherwise.} \end{cases}$$

We will show s is a bijection from $\mathcal{T}^{**}(G)$ to $\mathcal{T}_2(c(G))$.

Claim. *$s(T) \in \mathcal{T}_2(c(G))$.*

Proof. By Lemma 2.2.3, every vertex is connected to either x or y in $s(T)$. Also, $s(T)$ contains the path from x to y given by $C_T - yw_T + xw_T$ so that $s(T)$ has exactly one component. The number of edges is preserved under s, therefore $s(T)$ is a spanning tree. It is clear the $s(T) \in \mathcal{T}_2(c(G))$, since $d(s(T))$ contains the cycle $C_T - yw_T - yz_T + xw_T + xz_T$. $\square$

We first show s is injective. If for $T, T' \in \mathcal{T}^{**}(G)$, $T \neq T'$, we have $s(T) = s(T')$ then either $w_T \neq w_{T'}$ or $z_T \neq z_{T'}$. If $z_T = z_{T'}$, then $s(T)$ contains two paths from x to z_T via $C_T - yw_T + xw_T$ and $C_{T'} - yw_{T'} + xw_{T'}$. Similarly, if $w_T = w_{T'}$ then there are two paths from y to w_T via $C_T - yw_T$ and $C_{T'} - yw_{T'}$. Finally, if $w_T \neq w_{T'}$ and $z_T \neq z_{T'}$ then there are two paths from x to y via $C_T - yw_T + xw_T$ and $C_{T'} - yw_{T'} + xw_{T'}$. Therefore $|\mathcal{T}^{**}(G)| \leq |\mathcal{T}_2(c(G))|$.

Now we will show for every $T^* \in \mathcal{T}_2(c(G))$, $T^* = s(T)$ for some $T \in \mathcal{T}^{**}(G)$. If $T^* \in \mathcal{T}_2(c(G))$, then $d(T^*)$ is not a spanning tree of G. By Lemma 2.2.3, it must be that $d(T^*)$ contains exactly two components $d(T^*)_x$ and $d(T^*)_y$. Furthermore, the unique cycle in $d(T^*)$ contains two edges incident to x: xz and xw for $z \in A_{x\overline{y}}$ and $w \in A_{\overline{x}y}$. Since y and x are not in the same connected component in $d(T^*)$, this implies $yw \notin E(d(T^*))$. Therefore $d(T^*) - xw + yw$ contains a path from y to x via z

and w so that $d(T^*) - xw + yw \in \mathcal{T}(G)$ and in particular $d(T^*) - xw + yw \in \mathcal{T}^{**}(G)$ and $s(d(T^*) - xw + yw) = T^*$. We may now conclude $|\mathcal{T}^{**}(G)| = |\mathcal{T}_2(c(G))|$. $\square$

2.4 Compression Acting on Spanning Pseudoforests

We will now prove that compression decreases the number of spanning pseudoforests by carefully mapping elements of $\mathcal{P}_\ell(c(G))$ into $\mathcal{P}_\ell(G)$ injectively.

Theorem 2.4.1. *For each $\ell \in [n]$ and $x, y \in V(G)$,*

$$|\mathcal{P}_\ell(G_{x\to y})| \leq |\mathcal{P}_\ell(G)|.$$

Proof. For $i = 1, 2$, let $\mathcal{P}_\ell^{(i)}(G)$ and $\mathcal{P}_\ell^{(i)}(c(G))$ denote the pseudoforests in $\mathcal{P}_\ell(G)$ and $\mathcal{P}_\ell(c(G))$ respectively such that x and y are in exactly i components. Given P a pseudoforest of $c(G)$, let $P^*(x, y)$ denote the set of all paths from x to y in P such that the neighbor of y along that path is in $A_{x\overline{y}}$. Define

$$E^*(P) = \{xw \mid xw \in E(S) \text{ where } S \in P^*(x, y)\}$$

and the edge replacement function $t_P : E(c(G)) \to E(G)$

$$t_P(e) = \begin{cases} e\Delta\{x, y\} & \text{if } e \in E^*(P) \\ e\Delta\{x, y\} & \text{if } e \notin E(G) \\ e & \text{otherwise.} \end{cases}$$

Given from Proposition 2.3.1, let $\varphi : \mathcal{T}(c(G)) \to (\mathcal{T}(G)\backslash\mathcal{T}^*(G))$ be a bijective function. Now we define $\Psi : \mathcal{P}_k^{(1)}(c(G)) \to \mathcal{P}_k^{(1)}(G)$ as

$$\Psi(P) = \begin{cases} \varphi(P) & \text{if } P \in \mathcal{T}(c(G)) \\ d(P) & \text{if } d(P) \text{ is connected and unicyclic} \\ t_P(P) & \text{otherwise.} \end{cases}$$

If P is unicyclic and $d(P)$ is disconnected, then all paths in P from x to y of the form $xw_1 \cdots w_k y$ such that $w_1 \in A_{xy}$ and $w_k \in A_{x\overline{y}}$. Then $t_P(xw_1 \cdots w_k y) = yw_1 \cdots w_k x$ and so x and y are in the same component. By Lemma 2.2.3, this implies $t_P(P)$ is connected and since t_P preserves the number of edge, $t_P(P)$ is unicyclic.

We will now show Ψ is injective.

Lemma 2.4.1. *If P is unicyclic and $d(P)$ is not connected, then all paths between x and y in $t_P(P)$ are of the form*

$$xw_1 \cdots w_k y$$

such that $w_1 \in A_{x\overline{y}}$, $w_k \in A_{xy}$, and $xw_k \in E^*(P)$. Furthermore if $P_1 \neq P_2$, then $s_{P_1}(P_1) \neq s_{P_2}(P_2)$.

Proof. Note that every path from x to y in $s_P(P)$ must use an edge e such that

$$e\Delta\{x, y\} \in E^*(P).$$

If not, then that path is also in $d(P)$ and so $d(P)$ is connected. Hence it suffices to check that no path is of the form

$$xw_1 \cdots w_k y$$

such that $xw_k \in E^*(P)$ and $w_1 \in A_{xy}$. Since $xw_k \in E^*(P)$, the exists some $z \in A_{x\overline{y}}$ such that

$$xw_k v_1 \cdots v_k zy$$

is a path in P. Then $xw_1 \cdots w_k v_1 \cdots v_k zy$ is a walk in P. This then implies there exists a path in P from x to y of the form $xw_1 \cdots zy$, and since $z \in A_{x\overline{y}}$ it must be that $w_1 \in E^*(P)$. Hence $xw_1 \notin s_P(P)$ which yields a contradiction.

From this, we can define an inverse of s_P. Given $\hat{P} = s_P(P)$, let $\hat{P}(x,y)$ be the set of all paths between x and y and set

$$E^{-1}(P) = \{yw \,|\, yw \in E(S) \text{ where } S \in \hat{P}(x,y)\}.$$

Then

$$s'_{\hat{P}}(P) = \begin{cases} e\Delta\{x,y\} & \text{if } e \in E^{-1}(P) \\ e\Delta\{x,y\} & \text{if } e \notin E(c(G)) \\ e & \text{otherwise.} \end{cases}$$

Since $E^{-1}(P) = \{e\Delta\{x,y\} \,|\, e \in E^*(P)\}$, we have $s'_{\hat{P}}(s_P(P)) = P$. $\square$

Suppose $P_1 \neq P_2$ and $\Psi(P_1) = \Psi(P_2) = \hat{P}$. By Lemma 2.4.1, we may assume $\Psi(P_1) = s_{P_1}(P_1)$ and $\Psi(P_1) = d(P_2)$. $\hat{P}$ is connected and therefore by Lemma 2.4.1 contains a path of the form

$$yw_1 \cdots w_k x$$

such that $w_1 \in A_{xy}$ and $w_k \in A_{x\overline{y}}$. Since P_2 is connected, then by Lemma 2.4.1 there exists a path between x and y in P_2 of the form

$$xz_1 \cdots z_j y$$

such that $z_1 \in A_{xy}$ and $z_j \in A_{x\overline{y}}$. Note $w_1 \neq z_1$ since $yw_1 \in E(\hat{P})$ implies $yw_1 \in E(P_2)$, but yw_1x cannot be a path in P_2.

Case 1. Assume $w_k = z_j$. Then $yw_1 \cdots w_k = z_j \cdots z_1 x$ is a walk from y to x in $\hat{P}$. This implies that there exists a path from y to x of the form $yw_1 \cdots z_1 x$ and since $w_1, z_2 \in A_{xy}$ this contradictions Lemma 2.4.1.

Case 2. Assume $w_k \neq z_j$. Notice $xw_k \cdots w_1 y$ is a path in P_2. Since $w_k \in A_{xy}$ and $w_1 \in A_{x\overline{y}}$, this implies $xw_k \in E^*(P_2)$ so that $xw_k \notin E(s_{P_2}(P_2)) = E(\hat{P})$ which is a contradiction.

Let $P \in \mathcal{P}^{(2)}(c(G)) \cup \mathcal{P}^{(2)}(G)$. Call P **Type 1** if P_x is unicyclic and x lies on the unique cycle in P_x. In this case, let x_1^C and x_2^C be the neighbors of x in the cycle and $E^C(P_x) = \{xx_1^C, xx_2^C\}$. Call P **Type 2** if P_x is unicyclic and P is not Type 1. Let $\hat{x}$ be the unique neighbor of x in P_x such that $(P_x \backslash \{x\})_{\hat{x}}$ is unicyclic and let $E^S(P_x) = \{x\hat{x}\}$. If P_y is unicyclic, we similarly define **Type 3**, y_1^C, y_2^C, $E^C(P_y)$, **Type 4**, $\hat{y}$, and $E^S(P_y)$. If P_x and P_y are trees, call P **Type 5**.

Let $j_P : E(c(G)) \to E(K_n)$ be defined as

$$j_P(e) = \begin{cases} e\Delta\{x,y\} & \text{if } e \notin E(G) \\ e\Delta\{x,y\} & \text{if } e \in E^C(P_x) \cup E^S(P_x) \cup E^C(P_y) \cup E^S(P_y) \\ e & \text{otherwise.} \end{cases}$$

Then define $\Psi : \mathcal{P}_k^{(2)}(c(G)) \to \mathcal{P}_k(G)$ by

$$\Psi(P) = \begin{cases} j_P(P) & \text{if } P \text{ is Type 1 or 2 and } \hat{y} \in A_{x\overline{y}} \\ j_P(P) & \text{if } P \text{ is Type 1 or 2 and } y_1^C, y_2^C \in A_{x\overline{y}} \\ j_P(P) & \text{if } y_1^C \in A_{x\overline{y}} \text{ and } y_2^C \in A_{xy} \\ d(P) & \text{otherwise.} \end{cases}$$

Claim. *For $P_1, P_2 \in \mathcal{P}^{(2)}(c(G))$, if $P_1 \neq P_2$, then $\Psi(P_1) \neq \Psi(P_2)$.*

Proof. If P is Type 1 or 2 and $\hat{y} \in A_{x\overline{y}}$, then $j_P(P)$ is Type 1 and $(c(j_P(P)))_y$ contains two cycles so that $j_P(P) \neq d(P')$ for $P' \in \mathcal{P}_k^{(2)}(c(G))$. If P is Type 1 or Type 2 and $y_1^C, y_2^C \in A_{x\overline{y}}$, then $j_P(P)$ is Type 2 and both neighbors of x on the cycle of $j_P(P)$ are in $A_{x\overline{y}}$. In this case, $(c(j_P(P)))_y$ also contains two cycles. If P is Type 3 and $y_1^C \in A_{x\overline{y}}$ and $y_2^C \in A_{xy}$, then $j_P(P)$ is Type 2 and x has a neighbor on the cycle of $j_P(P)$ which is in A_{xy}. Also, if $j_P(P) = d(P')$ then since $c(j_P(P))$ contains a path from x to y via the edges yy_1^C and xy_2^C, $P \in \mathcal{P}^{(1)}(c(G))$. □

It suffices to show that if $P \in \mathcal{P}_k^{(2)}(c(G))$ such that

$$\Psi(P) = s(P) \in \mathcal{P}_k^{(1)}(G),$$

then $\Psi(P) \neq \Psi(P')$ for $P' \in \mathcal{P}_k^{(1)}(c(G))$.

Case 1. $\Psi(P) = T \in \mathcal{T}(G)$. Then P_x is a tree and P_y is unicyclic such that $y_1^C \in A_{\overline{x}y}$ and $y_2^C \in A_{x\overline{y}}$. Then $c(T)$ contains a cycle of the form $yy_1^C \cdots y_2^C y$ with $y_1^C \in A_{xy}$ and $y_2^C \in A_{\overline{x}y}$. This implies $T \in \mathcal{T}^*(G)$ and thus by definition of φ, $T \neq \Psi(P')$ for $P' \in \mathcal{P}^{(1)}(c(G))$.

Case 2. P_x and P_y are unicyclic such that $y_1^C \in A_{\bar{x}y}$ and $y_2^C \in A_{x\bar{y}}$. If $P' \in \mathcal{P}^{(1)}(c(G))$ such that $\Psi(P') = \Psi(P)$, then $c(\Psi(P')) = P_x \sqcup P_x$ disconnected implies $\Psi(P') = s_{P'}(P')$. But $yy_1^C \cdots y_2^C x$ is a path in $s_{P'}(P')$ with $y_1^C \in A_{\bar{x}y}$ and so this contradicts Lemma 2.4.1. □

2.5 Proof of Theorem 2.1.1

Recall that Theorem 2.1.1, our main theorem in this chapter, is that there exists a threshold graph in $\mathcal{G}_{n,e}$ with the minimum number of size k spanning pseudoforests for all k.

Proof of Theorem 2.1.1. Select a graph H from $\mathcal{G}_{n,e}$ with $|\mathcal{P}_k(H)|$ minimal. By Theorem 2.4.1, compression does not increase the number of spanning pseudoforests and so any compression of H is also minimal. Among these, pick H with $d_2(H)$ maximal. By Corollary 2.2.2, H is threshold. □

2.6 Future Directions

There are still many open problems related to this question. For instance, it would be very interesting to know which graphs attain the minimum value of $|\mathcal{P}_\ell|$. For small ℓ the problem is trivial since the smallest connected graph (in terms of size) that is neither a tree nor unicyclic is $K_4 - e$, often referred to as the *diamond graph*. Thus for $\ell \in \{0, 1, \ldots, 4\}$, any subgraph of G with ℓ edges will be a pseudoforest.

Remark. *For $G \in \mathcal{G}_{n,e}$ and $\ell \in \{0, 1, \ldots, 4\}$,*

$$|\mathcal{P}_\ell(G)| = \binom{e}{\ell}.$$

Hence the first interesting case is $\ell = 5$. Since the diamond graph is the only graph with 5 edges that is neither a tree nor unicyclic, we have

$$|\mathcal{P}_5(G)| = \binom{e}{5} - |\{H \subseteq G \,:\, H \cong K_4 - e\}|$$

which leads us to the following question.

Question. *Which threshold graph(s) in $\mathcal{G}_{n,e}$ have the maximum number of diamond subgraphs?*

In Chapter 3 we investigate the minimum version of this question, seeking out which threshold graphs have the *maximum* value of $|\mathcal{P}_5|$. For larger values of ℓ, the proof of Lemma 2.1.1 can be reverse-engineered to show that $|\mathcal{P}_n(G)|$ counts the number of decompositions of G into vertex-disjoint unicyclic graphs. The question then of finding the graphs with the minimum value of $|\mathcal{P}_n|$, although very interesting, seems to be more difficult than the same question for smaller values of ℓ.

Chapter 3

Threshold Graphs with Few Diamond Subgraphs

Recall from Section 2.6, counting the number of size 5 pseudoforests in a graph from $\mathcal{G}_{n,e}$ can be done by counting the number of $K_4 - e$ subgraphs (depicted in Figure 3). The question of maximizing the number of K_4 subgraphs over all graphs in $\mathcal{G}_{n,e}$ has been answered by Cutler and Radcliffe in [7]—they proved that there is indeed a threshold graph which maximizes the number of cliques of any fixed size. Let $\mathcal{T}_{n,e}$ denote the set of threshold graphs with n vertices and e edges. Keough and Radcliffe later prove in [25] that amongst all graphs in $\mathcal{T}_{n,e}$, the colex graph minimizes the number of independent sets of a fixed size.

Definition 3.0.1. The *colexicographic order*, $<_C$, is defined by $A <_C B$ if $\max(A \Delta B) \in B$. The *colex graph* $\mathcal{C}(n, e)$ is the graph with vertex set $[n]$ and edge set consisting of the first e edges in colex order on $E(K_n)$.

This result implies that the complement of the colex graph minimizes the number of cliques of a fixed size. In particular, the complement of the colex graph has the fewest number of K_4 subgraphs among all threshold graphs with n vertices and e edges.

As noted in Section 2.6, there is a threshold graph which minimizes the number of size 5 pseudoforests and hence maximizes the number of $K_4 - e$ (diamond) subgraphs

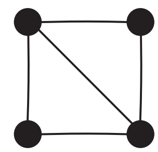


Figure 3.1: The diamond subgraph.

among all graphs in $\mathcal{G}_{n,e}$. Let $\mathcal{D}(\cdot)$ denote the number of diamond subgraphs. In this chapter we make progress in determining which graphs in $\mathcal{T}_{n,e}$ minimize $\mathcal{D}(\cdot)$.

3.1 Definitions and Results

Recall that one definition of a threshold graph is a graph that can be built sequentially from the empty graph by adding vertices one at a time, where each new vertex is either isolated or dominating. We can then code a threshold graph on n vertices by a binary code of length n. Letting 1 stand for a dominating vertex and 0 stand for an isolated vertex, construct the threshold graph by reading the binary string from right to left. The threshold graph with binary code σ will be denoted $T(\sigma)$ and we will refer to σ as the *code* of the graph $T(\sigma)$. Given a threshold graph code $\alpha_{n-1}\cdots\alpha_1\alpha_0$, let vertex v_i be the vertex added by the digit α_i. We will call a vertex v_i *dominating* if $\alpha_i = 1$ and *isolated* otherwise. Note that a dominating (resp. isolated) vertex is dominating (resp. not adjacent to) vertices preceding it in the build, but not necessarily *all* other vertices of the graph. In particular, the neighborhood of a vertex v_k of $T(\alpha_n \cdots \alpha_1 *)$ is given by

$$N(v_k) = \begin{cases} \{v_i \,:\, i > k \text{ and } \alpha_i = 1\} & \text{if } \alpha_k = 0, \\ \{v_i \,:\, i < k\} \cup \{v_i \,:\, i > k \text{ and } \alpha_i = 1\} & \text{if } \alpha_k = 1. \end{cases} \tag{3.1}$$

Example 3.1.1. *The following is the threshold graph with code 1001101.*

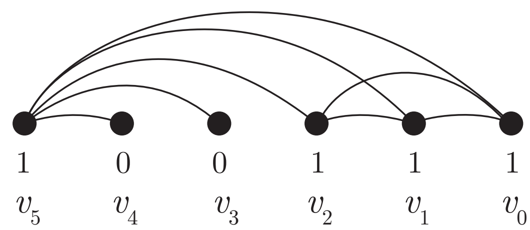


Figure 3.2: $T(1001101)$

One may have noticed that the same threshold graph is produced if the first vertex, v_0 is isolated or dominating, e.g., $T(1001101)$ is the same graph as $T(1001100)$. Because of this, we may instead think of a threshold graph as a binary sequence of length $n-1$ and put a $*$ at the far right end to represent the first vertex added when building the graph. With this, the code of the graph in Figure 3.1.1 can instead be written as $100110*$.

When considering a string of 0s or 1s, we will abbreviate such a string by a single digit with a superscript indicating the length of the string. For example, 000 will be replaced with 0^3. The main result of this chapter is the following.

Proposition 3.1.1. *If $G \in \mathcal{T}_{n,e}$ has the minimum number of diamond subgraphs, then one of the following holds*

(i) $e \leq n-1$,

(ii) $G = T(1^p0^q*)$,

(iii) $G = T(1^p0^q1*)$,

(iv) $G = T(1^p0^q10^{r_1}10^{r_2}\cdots10^{r_s}10^t*)$ *and* $t \geq 1$,

(v) $G = T(1^p0^q10^{r_1}10^{r_2}\cdots10^{r_s}10^t1*)$ *and* $t \geq 3$, *or*

(vi) $G = T(1^p*)$

where $p, q \geq 1$, *and* $r_i \geq 4$ *for all* $i \in [s]$.

We will postpone the proof of this proposition to Section 3.3.

Example 3.1.2. *The following are examples of threshold codes of the form described in Theorem 3.1.1.*

(i) $11010000010000* = 1^2 0^1 1 0^5 1 0^4 *$

(ii) $1001000010000000010001* = 1^1 0^2 1 0^4 1 0^7 1 0^3 1*$

(iii) $11101* = 1^3 0^1 1*$

It is important to note that this does not characterize the codes of threshold graphs with the minimum number of diamond subgraphs. Indeed, there are threshold graphs whose codes are of the form given above that do not have the minimum number of diamond subgraphs. It is also worth noting that given any $n \geq 0$, $e \leq \binom{n}{2}$, one can construct a threshold graph that is of one of the forms given in Proposition 3.1.1.

Lemma 3.1.1. *Given integers* n *and* e *such that* $n \geq 0$ *and* $0 \leq e \leq \binom{n}{2}$, *there exists a graph in* $\mathcal{T}_{n,e}$ *of one of the forms given in Proposition 3.1.1.*

Proof. Suppose $e \geq n$ and let $s(k) = kn - \frac{1}{2}k(k+1)$. If $e = s(n-1)$, then $T(1^{n-1}*)$ is a threshold graph of order n and size $s(n-1) = e$ of the form (vi) in Proposition 3.1.1. Otherwise, let $1 \leq k \leq n-2$ be the maximum integer such that $s(k) \leq e$.

$$\begin{aligned} e - s(k) &< s(k+1) - s(k) \\ &= (k+1)n - \frac{1}{2}(k+1)(k+2) - \left(kn - \frac{1}{2}k(k+1)\right) \\ &= n - k - 1. \end{aligned}$$

Therefore since $n-k-(e-s(k)+1) \geq 1$, $T(1^k 0^{n-k-(e-s(k)+1)} 1 0^{e-s(k)})$ is indeed a threshold graph of order n and size $s(k)+(e-s(k)) = e$. Note that this graph is either of the form *(ii)*, *(iii)*, or *(iv)*. $\square$

3.2 Local Moves Acting on Diamond Subgraphs

Kloks, Kratsch, and Müller showed in [26] that the number of diamond subgraphs can be computed straightforwardly using the graph's adjacency matrix A.

Lemma 3.2.1 ([26])**.**

$$\mathcal{D}(G) = \sum_{\{x,y\} \in E(G)} \binom{(A^2)_{x,y}}{2} = \sum_{\{x,y\} \in E(G)} \binom{|N(x) \cap N(y)|}{2}$$

Here, we will instead use the code of a threshold graph to determine the number of diamond subgraphs it contains. Since the diamond graph has order 4, we will count copies of this graph inside induced subgraphs of order 4. The following well-known property of threshold graphs states that induced subgraphs of threshold graphs are also threshold, and their codes can easily be obtained from the host graph's code.

Lemma 3.2.2. *Let $G = T(\alpha_{n-1} \cdots \alpha_1 *)$. For $i_k > \cdots > i_1$*

$$G[\{v_{i_k}, \ldots, v_{i_1}\}] = T(\alpha_{i_k} \cdots \alpha_{i_1}).$$

Proof. Let $S = \{v_{i_k}, \ldots, v_{i_1}\}$ and $G' = G[S]$, we will first show G' is threshold using Part of Theorem 1.1.1. Since G is threshold, given $v, v' \in V(G') \subseteq V(G)$ either $N_G(v) \subseteq N_G[v']$ or $N_G(v') \subseteq N_G[v]$. Without loss of generality, assume $N_G(v) \subseteq N_G[v']$. Note $N_{G'}(v) = N_G(v) \cap S$ and $N_{G'}(v') \cap S$, thus $N_{G'}(v) \subseteq N_{G'}(v)$ which proves G' is threshold.

We will now show that if $G' = T(\alpha'_{i_k} \cdots \alpha'_{i_2}*)$, then $\alpha'_{i_j} = \alpha_{i_j}$ for all $j \in \{2, \ldots, k\}$. We proceed by induction on $k - j$. For $j = k$, if $\alpha_{i_k} = 1$, then $d_{G'}(v_{i_k}) = k - 1$. Since the threshold graph G' has order k, it has a vertex of degree $k - 1$ only if $\alpha'_{i_k} = 1$. Otherwise $\alpha_{i_k} = 0$ which implies $d_{G'}(v_{i_k}) = 0$ and hence $\alpha'_{i_k} = 0$. Now suppose $0 < j < k$ and $\alpha'_{i_\ell} = \alpha_{i_\ell}$ for all $\ell < j$. Using Equation 3.1 for $N_G(v_j)$ we have that the neighborhood of v_{i_j} in G' is given by

$$\begin{aligned}
N_{G'}(v_{i_j}) &= N_G(v_{i_j}) \cap S \\
&= \begin{cases} \{v_i \,:\, i > i_j \text{ and } \alpha_i = 1\} \cap S & \text{if } \alpha_{i_j} = 0, \\ (\{v_i \,:\, i < i_j\} \cup \{v_i \,:\, i > i_j \text{ and } \alpha_i = 1\}) \cap S & \text{if } \alpha_{i_j} = 1 \end{cases} \\
&= \begin{cases} \{v_{i_\ell} \,:\, \ell > j \text{ and } \alpha_{i_\ell} = 1\} & \text{if } \alpha_{i_j} = 0, \\ (\{v_{i_\ell} \,:\, \ell < j\} \cup \{v_{i_\ell} \,:\, \ell > j \text{ and } \alpha_{i_\ell} = 1\}) & \text{if } \alpha_{i_j} = 1. \end{cases} \\
&= \begin{cases} \{v_{i_\ell} \,:\, \ell > j \text{ and } \alpha'_{i_\ell} = 1\} & \text{if } \alpha_{i_j} = 0, \\ (\{v_{i_\ell} \,:\, \ell < j\} \cup \{v_{i_\ell} \,:\, \ell > j \text{ and } \alpha'_{i_\ell} = 1\}) & \text{if } \alpha_{i_j} = 1. \end{cases}
\end{aligned}$$

Applying Equation 3.1 for $N_{G'}(v_{i_j})$, this implies $\alpha_{i_j} = \alpha'_{i_j}$. Therefore $G' = T(\alpha_{i_k} \cdots \alpha_{i_2}*)$.

□

The lemma below states that one can easily identify a diamond subgraph from the code of a threshold graph.

Lemma 3.2.3. *Let $\sigma* = \alpha_{n-1} \cdots \alpha_1 *$ be the code of some n-vertex threshold graph*

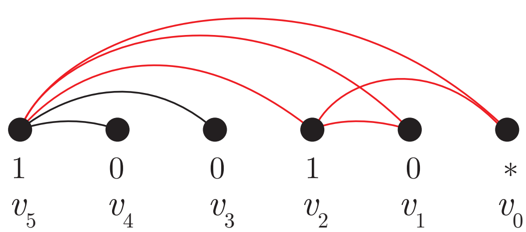
G. Then, given $i_4 > i_3 > i_2 > i_1$,

$$\mathcal{D}(G[\{v_{i_4}, v_{i_3}, v_{i_2}, v_{i_1}\}]) = \begin{cases} 1, & \textit{if } \alpha_{i_4}\alpha_{i_3}\alpha_{i_2}\alpha_{i_1} = 110* \\ 6, & \textit{if } \alpha_{i_4}\alpha_{i_3}\alpha_{i_2}\alpha_{i_1} = 111* \\ 0, & \textit{otherwise.} \end{cases}$$

where $*$ is either 0 or 1.

Proof. Suppose $G[\{v_{i_4}, v_{i_3}, v_{i_2}, v_{i_1}\}]$ contains a copy of the diamond graph and notice by Lemma 3.2.2, $G[\{v_{i_4}, v_{i_3}.v_{i_2}, v_{i_1}\}] = T(\alpha_{i_4}\alpha_{i_3}\alpha_{i_2}\alpha_{i_1})$. Vertices v_{i_4} and v_{i_3} have degree at least 2 in $T[v_{i_4}, v_{i_3}, v_{i_2}, v_{i_1}]$ only if $\alpha_{i_4} = \alpha_{i_3} = 1$. Since every vertex in the diamond graph has degree at least 2, we must have $\alpha_{i_4} = \alpha_{i_3} = 1$. If $\alpha_{i_3} = 0$, then $T(\alpha_{i_4}\alpha_{i_3}\alpha_{i_2}\alpha_{i_1}) \cong K_4 - e$. Otherwise, $\alpha_{i_3} = 1$ and so $T(\alpha_{i_4}\alpha_{i_3}\alpha_{i_2}\alpha_{i_1}) \cong K_4$. In this case, removing any of the 6 edges of K_4 gives a copy of the diamond graph. □

Example 3.2.1. *The set of vertices $\{v_5, v_2, v_1, v_0\}$ in the graph below form a $110*$ code and hence induce a $K_4 - e$.*



Corollary 3.2.1. *If $G \in \mathcal{T}_{n,e}$ contains no diamond subgraphs, then $e \le n$.*

Proof. Suppose $G \in \mathcal{T}_{n,e}$ does not contain a copy of $K_4 - e$. If there only exists one i such that $\alpha_i = 1$ in the code of G, then $e = i \le n - 1$. Otherwise there exists $\alpha_i = \alpha_j = 1$ in the code of G such that $i < j$ and $j \neq 0$. We must have that $j = 1$, else

$\alpha_i\alpha_j\alpha_1\alpha_0 = 11**$ and so by Lemma 3.2.3, G contains a copy of $K_4 - e$. Thus there are exactly two 1s in the code of G: α_i and α_1. Since $i \leq n-1$, $e = i+1 \leq n$. □

In what follows we will determine how local changes in a threshold graph's code will affect the number of diamond subgraphs it contains. Notice that changing an 01 to a 10 in a threshold graph's code corresponds to adding the edge between the corresponding vertices. The lemma below calculates exactly how many diamond subgraphs are gained by changing an 01 to a 10 in the code. Let $\ell(\cdot)$ denote the length of a binary string and $w(\cdot)$ denote the number of 1s in a binary string.

Lemma 3.2.4. *Let σ and ρ be (possibly empty) binary strings, then*

$$\mathcal{D}(T(\sigma 10\rho)) = \mathcal{D}(T(\sigma 01\rho)) + w(\sigma)\ell(\rho) + 5\binom{w(\sigma)}{2}.$$

Proof. Let u and v be the vertices added by the string 01 where u is isolated and v is dominating in $T(\sigma 01\rho)$. Note that $T(\sigma 10\rho) = T(\sigma 01\rho) + uv$, hence

$$\mathcal{D}(T(\sigma 10\rho)) = \mathcal{D}(T(\sigma 01\rho)) + |\{H \subseteq T(\sigma 10\rho) \,:\, H \cong K_4 - e,\, uv \in E(H)\}|.$$

Let x and y be vertices such that $T(\sigma 10\rho)$ induced on $\{x, y, u, v\}$ contains a copy of $K_4 - e$. Since u has no neighbors in $V(T(\rho))$, Lemma 3.2.3 implies that one of either x or y is not in $V(T(\rho))$. Lemma 3.2.3 also gives that $x \in V(T(\sigma))$ and $y \in V(T(\rho))$ if and only if x is dominating. There are $w(\sigma)\ell(\rho)$ such pairs of $\{x, y\}$ and each pair gives exactly one copy of $K_4 - e$ using the edge uv. Otherwise $x, y \in V(T(\sigma))$ which holds if and only if x and y are both dominating. There are $\binom{w(\sigma)}{2}$ pairs of dominating vertices x, y and since $\{u, v, x, y\}$ induces a K_4 in $T(\sigma 10\rho)$, there are exactly 5 of copies of $K_4 - e$ containing the edge uv. Therefore

$$|\{H \subseteq T(\sigma 10\rho) \,:\, H \cong K_4 - e,\, uv \in E(H)\}| = w(\sigma)\ell(\rho) + 5\binom{w(\sigma)}{2}$$

which concludes the proof. □

By changing an 01 to a 10 in the code of a threshold graph in $\mathcal{G}_{n,e}$, the outcome is a threshold graph in $\mathcal{G}_{n,e+1}$. Thus if a code of a threshold graph in $\mathcal{G}_{n,e}$ has a 01 and a 10 which don't overlap, changing them to a 10 and an 01 respectively, the resulting graph will remain in $\mathcal{G}_{n,e}$. By making these two switches, the following corollary describes when the resulting graph will have fewer diamond subgraphs.

Corollary 3.2.2. *Let σ, τ, ρ be (possibly empty) binary strings, then if $w(\sigma) = 0$ or $w(\tau) \geq \frac{\ell(\tau)-3}{5}$*

$$\mathcal{D}(T(\sigma 01\tau 10\rho)) \geq \mathcal{D}(T(\sigma 10\tau 01\rho)).$$

If $w(\sigma) = 0$, then equality only occurs if $w(\tau) = \ell(\rho) = 0$.

Proof. Letting $\rho_1 = \tau 10\rho$, then $T(\sigma 01\tau 10\rho) = T(\sigma 01\rho_1)$ and from Lemma 3.2.2 we have

$$\mathcal{D}(T(\sigma 10\rho_1)) = \mathcal{D}(T(\sigma 01\rho_1)) + w(\sigma)\ell(\rho_1) + 5\binom{w(\sigma)}{2}.$$

Similarly, letting $\sigma_2 = \sigma 10\tau$ then $\mathcal{D}(T(\sigma 10\rho_1)) = \mathcal{D}(T(\sigma_2 10\rho))$ and by Lemma 3.2.2, we have

$$\mathcal{D}(T(\sigma_2 10\rho)) = \mathcal{D}(T(\sigma_2 01\rho)) + w(\sigma_2)\ell(\rho) + 5\binom{w(\sigma_2)}{2}.$$

Hence since $\rho_1 = \tau 10 \rho$ and $\sigma_2 = \sigma 10 \tau$ we have

$$
\begin{aligned}
&\mathcal{D}(T(\sigma 01 \tau 10 \rho)) \\
&\quad = \mathcal{D}(T(\sigma 01 \rho_1)) \\
&\quad = \mathcal{D}(T(\sigma 10 \rho_1)) - \left(w(\sigma)\ell(\rho_1) + 5\binom{w(\sigma)}{2} \right) \\
&\quad = \mathcal{D}(T(\sigma 10 \rho_1)) - \left(w(\sigma)(\ell(\tau) + \ell(\rho) + 2) + 5\binom{w(\sigma)}{2} \right) \\
&\quad = \mathcal{D}(T(\sigma_2 10 \rho)) - \left(w(\sigma)(\ell(\tau) + \ell(\rho) + 2) + 5\binom{w(\sigma)}{2} \right) \\
&\quad = \mathcal{D}(T(\sigma_2 01 \rho)) + w(\sigma_2)\ell(\rho) + 5\binom{w(\sigma_2)}{2} - \left(w(\sigma)(\ell(\tau) + \ell(\rho) + 2) + 5\binom{w(\sigma)}{2} \right) \\
&\quad = \mathcal{D}(T(\sigma 10 \tau 01 \rho)) + w(\sigma_2)\ell(\rho) + 5\binom{w(\sigma_2)}{2} - \left(w(\sigma)(\ell(\tau) + \ell(\rho) + 2) + 5\binom{w(\sigma)}{2} \right).
\end{aligned}
$$

The difference

$$\mathcal{D}(T(\sigma 01 \tau 10 \rho)) - \mathcal{D}(T(\sigma 10 \tau 01 \rho))$$

is given by

$$
\begin{aligned}
&\quad w(\sigma_2)\ell(\rho) + 5\binom{w(\sigma_2)}{2} - \left(w(\sigma)(\ell(\tau) + \ell(\rho) + 2) + 5\binom{w(\sigma)}{2} \right) \\
&= (w(\sigma) + w(\tau) + 1)\ell(\rho) + 5\binom{w(\sigma) + w(\tau) + 1}{2} - \left(w(\sigma)(\ell(\tau) + \ell(\rho) + 2) + 5\binom{w(\sigma)}{2} \right) \\
&= w(\tau)\ell(\rho) + \ell(\rho) - w(\sigma)\ell(\tau) - 2w(\sigma) + 5\left(\binom{w(\sigma) + w(\tau) + 1}{2} - \binom{w(\sigma)}{2} \right) \\
&= w(\tau)\ell(\rho) + \ell(\rho) - w(\sigma)\ell(\tau) - 2w(\sigma) + 5\left(w(\sigma)w(\tau) + w(\sigma) + \frac{w(\tau)^2 + w(\tau)}{2} \right) \\
&= \frac{5}{2}w(\tau)\left(w(\tau) + \frac{2}{5}\ell(\rho) + 1 \right) + \ell(\rho) - w(\sigma)(\ell(\tau) - 5w(\tau) - 3). \qquad (3.2)
\end{aligned}
$$

Therefore

$$\mathcal{D}(T(\sigma 01 \tau 10 \rho)) - \mathcal{D}(T(\sigma 10 \tau 01 \rho)) \geq 0$$

if $w(\sigma) = 0$ or $w(\tau) \geq \frac{\ell(\tau)-3}{5}$. $\square$

3.3 Proof of Proposition 3.1.1

We will now use Corollary 3.2.2 to prove the main result of this chapter which narrows down the types of codes for threshold graphs which have the minimum number of diamond subgraphs.

Proof of Proposition 3.1.1. Let $G = T(\alpha_{n-1} \cdots \alpha_1 *) \in \mathcal{G}_{n,e}$ such that G minimizes the number of diamond subgraphs. Further suppose $e \geq n$.

We will first show α_{n-1}. Let k be maximum such that $\alpha_k = 1$ and assume by way of contradiction that $k < n-1$. Since $e \geq n$ and $\alpha_{k-1} \neq 1$, there exists some minimum j such that $0 < j < k$ and $\alpha_j = 1$. We may apply Corollary 3.2.2 to G with

$$\begin{aligned}
\sigma_1 &= \alpha_{n-1} \cdots \alpha_{k+2} = 0^{n-2-k} \\
\alpha_{k+1}\alpha_k &= 01 \\
\tau_1 &= \alpha_{k-1} \cdots \alpha_{j+1} \\
\alpha_j \alpha_{j+1} &= 10 \\
\rho_1 &= \alpha_{j+1} \cdots \alpha_0 * = 0^{j-2} *
\end{aligned}$$

since $w(\sigma_1) = 0$, which implies

$$\mathcal{D}(G) = \mathcal{D}(T(\sigma_1 01 \tau_1 10 \rho_1)) \leq \mathcal{D}(T(\sigma_1 10 \tau_1 01 \rho_1)).$$

Also note this is an equality only if $w(\tau_1) = 0$ and $\ell(\rho_1) = 0$, in which case $G = T(0^j 10^k 1*)$ which implies $e \leq n-1$. Hence $T(\sigma_1 10 \tau_1 01 \rho_1) \in \mathcal{G}_{n,e}$ has strictly fewer diamond subgraphs than G which yields a contradiction. Therefore $\alpha_{n-1} = 1$.

We will now show that strings of 1s of length at least two will only appear at the far left of the code. Again we suppose to the contrary that we can write

$$\alpha_{n-1}\cdots\alpha_1* = \sigma_2 01^k 0\rho_2$$

such that $k \geq 2$. Then applying Corollary 3.2.2 to G with

$$\tau_2 = 1^{k-2}$$

we have $w(\tau_2) = \ell(\tau) \geq \frac{\ell(\tau)-3}{5}$. From our previous argument, since $\alpha_{n-1} = 1$, then $w(\sigma) \geq 1$ and so Equation 3.2 implies

$$\mathcal{D}(G) = \mathcal{D}(T(\sigma_2 01\tau_2 10\rho_2)) < \mathcal{D}(T(\sigma_2 10\tau_2 01\rho_2)).$$

Therefore the code of G only contains a string of more than one 1 at the leftmost part of the code.

Finally, we will show that besides at the two far ends of the code, 1s are separated by a long string of 0s. In particular, suppose that

$$\alpha_{n-1}\cdots\alpha_1* = \sigma_3 010^j 10\rho_3$$

with $j \leq 3$. We may apply Corollary 3.2.2 to G with $\tau_3 = 0^j$ since $w(\tau_3) = 0 \geq \frac{\ell(\tau_3)-3}{5}$, this gives

$$\mathcal{D}(G) = \mathcal{D}(T(\sigma_3 01\tau_3 10\rho_3)) \leq \mathcal{D}(T(\sigma_3 10\tau_3 01\rho_3)).$$

Again note that $\alpha_{n-1} = 1$ so that $w(\sigma_3) \geq 1$, hence using Equation 3.2 we can deduce that this inequality is strict unless $j = 3$ and ρ_3 is empty. Therefore strings of 0s of length three or fewer only appear in one of the following cases:

(i) Following the first string of 1s on the leftmost side of the code.

(ii) Preceding $\alpha_0 = *$.

(iii) String of length exactly 3 preceding $\alpha_1 = 1$.

□

3.4 Future Directions

There are two clear open questions that result from this work. The first is to find an exact characterization of graphs in $\mathcal{T}_{n,e}$ which have the minimum number of diamond subgraphs. We believe that this would likely require a very different approach from the one given here. Evidence from examples generated by a program in Sage suggest a correlation between codes of graphs in $\mathcal{T}_{n,e}$ and $\mathcal{T}_{n,e+1}$ which have the minimum number of diamond subgraphs. It would be interesting to determine if such a diamond minimizer in $\mathcal{T}_{n,e+1}$ can be built from an existing diamond minimizer in $\mathcal{T}_{n,e}$.

Referring back to the original question in Section 2.6, we would also like to know which graphs in $\mathcal{T}_{n,e}$ have the *largest* number of diamond subgraphs. For $n \leq 13$, the maximizer is the colex graph in $\mathcal{G}_{n,e}$. If we were to apply the common approach to showing that the extremal graph is colex to this situation, we would need to show

(i) for all $j \geq 1$, $\mathcal{D}(T(\sigma 101^j 01\rho)) \leq \mathcal{D}(T(\sigma 01^{j+2} 0\rho))$, and

(ii) for all $j \geq 2$, $\mathcal{D}(T(\sigma 10^j 1\rho)) \leq \mathcal{D}(T(\sigma 010^{j-2} 10\rho))$.

Corollary 3.2.2 implies that the former of these two statements is true, while Equation 3.2 can be used to cook up an example in which the latter statement is false. This suggests that another approach must be used, or that for larger values of n the extremal graphs may *not* be colex. The latter speculation is perhaps strengthened by the fact that colex graphs often do not maximize the sum of the number of induced K_4s and the number of induced $(K_4 - e)$s. Thus, it seems unnatural to think that weighting an induced copy of K_4 by 6, which is how one counts diamond subgraphs, would affect the extremal graphs so drastically.

Chapter 4

Minimizing Rooted Spanning Forests

Recall the result of Satyanarayana, Schoppmann, and Suffel in Theorem 2.1.2 which states that compression decreases the number of spanning trees in a graph. Since compression does not change the order of the graph, this implies that compression also decreases the number of *rooted* spanning trees. A rooted spanning tree is a spanning tree with exactly one vertex labeled, hence the number of rooted spanning trees in a graph G of order n is given by $n\mathcal{T}(G)$. In this chapter we generalize this result by showing that for all k, compression decreases the number of rooted spanning forests with exactly k components. This then proves that there exists a threshold graph in $\mathcal{G}_{n,e}$ which minimizes this set of parameters simultaneously.

This result is seen in a paper by Csikvári published in 2011, but Section 4.2 will illustrate a flaw in the proof given there. We then proved the result with a different approach than that of the original paper, but later communication with Csikvári revealed that the result had been proven even prior to his paper in a series of lectures given by Kelmans between the years 1992 and 2009 [23]. Although our proof we give in the following was proven independently, it aligns with the proof of Kelmans highlighting that this is a natural argument for this result.

4.1 Definitions and Results

Let $\mathcal{F}_k(G)$ be the set of spanning forests of G with exactly k components. When considering only the special case of $k = 1$, we may denote the set of spanning trees instead by $\mathcal{T}(G)$. Given $F \in \mathcal{F}_k(G)$ with components $T_1, \ldots, T_k$, let $\gamma(F) = \prod_{i=1}^{k} n_i$ where n_i is the number of vertices in the tree T_i.

Definition 4.1.1. A *rooted spanning forest* is a spanning forest in which exactly one vertex from each component is labeled.

Hence the number of rooted spanning forests of G with exactly k components is given by

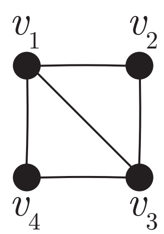
$$\sum_{F \in \mathcal{F}_k(G)} \gamma(F).$$

Rooted spanning forests appeared in a generalization of the Matrix Tree Theorem by Chelnokov and Kelmans in 1974. This generalization determined the coefficients of the characteristic polynomial of the Laplacian in terms of the number of rooted spanning forests the graph contains.

Definition 4.1.2. The *Laplacian matrix* of a graph, denoted $L(G)$, is an integer-valued matrix whose rows and columns are indexed by the vertices and whose entries are given by

$$L(G)_{ij} = \begin{cases} d(v_i) & \text{if } i = j \\ -1 & \text{if } v_i \sim v_j \\ 0 & \text{otherwise.} \end{cases}$$

Example 4.1.1. *For the diamond graph, shown on the left, its Laplacian matrix is given on the right.*



$$L(K_4 - e) = \begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{bmatrix}$$

Theorem 4.1.1 ([4])**.** *Denote the characteristic polynomial of the Laplacian matrix of G by $\Phi(G, x) = \sum_{k=0}^{n}(-1)^{n-k}c_k(G)x^k$. Then*

$$c_k(G) = \sum_{F \in \mathcal{F}_k(G)} \gamma(F),$$

where if F has components $T_1, \ldots, T_k$ with $n(T_i) = n_i$, $\gamma(F) := \prod_{i=1}^{k} n_i$.

We will refer to $c_k(G)$ as the kth Laplacian coefficient, although the coefficient actually is $(-1)^{n-k}c_k(G)$. The problem of which graphs have minimal Laplacian coefficients for all k has been solved for trees, unicyclic, bicyclic, and tricyclic graphs ([14],[29],[18],[19],[17],[13]). Often the approach is to define a graph operation which acts monotonically on all Laplacian coefficients—this is also the approach we will take here.

As in the proof of Theorem 2.1.1, we will first show that compression decreases the number of k-component rooted spanning forests. In Section 4.5, we will then prove the main result given below.

Theorem 4.1.2. *For each $k \in [n]$, if $H \in \mathcal{G}_{n,e}$ satisfies*

$$\sum_{F \in \mathcal{F}_k(H)} \gamma(F) = \min_{G \in \mathcal{G}_{n,e}} \sum_{F \in \mathcal{F}_k(G)} \gamma(F),$$

then H is threshold.

4.2 Proof of Csikvári

We remark that the following result appeared in [5], but we will show that although this fact remains true, the original author's proof was incorrect. Below we follow the notation outlined in Section 2.2.

Lemma 4.2.1. *For each $k \in [n]$,*

$$\sum_{F \in \mathcal{F}_k(c(G))} \gamma(F) \leq \sum_{F \in \mathcal{F}_k(G)} \gamma(F).$$

We will postpone the proof of Lemma 4.2.1 to Section 4.4. Given $x, y \in V(G)$, we will consider compression on G from x to y. Let R be a subset of the edge set $\{yw \,|\, w \in A_{xy}\}$ and let

$$\begin{aligned} S(G)_R = \{(T_1, T_2) \,|\, T_1, T_2 \text{ trees}, x \in V(T_1), y \in V(T_2), \\ V(T_1) \cap V(T_2) = \emptyset, V(T_1) \cup V(T_2) = V(G), R \subseteq E(T_1)\} \\ s(G, R, x, y) = \sum_{(T_1, T_2) \in S(G)_R} |V(T_1)||V(T_2)|. \end{aligned}$$

Suppose $h = yw$ is an edge not in R such that $w \in A_{xy}$. In order to use an induction hypothesis, it was stated that $G_{x \to y} - h$ can be obtained from $G - h$ via a compression from x to y. This is incorrect, since w is still a neighbor of x but is no longer a neighbor of y in $G - h$. Thus $(G - h)_{x \to y}$ contains the edge h and $c(G) - h$ does not. Unless $A_{x\overline{y}} \cup A_{\overline{x}y} = \emptyset$, we have $G_{x \to y} - h \neq (G - h)_{x \to y}$. This claim allowed the author to

Figure 4.1: $s(G-h, \emptyset, x, y) = 4 < 6 = s(G_{x\to y} - h, \emptyset, x, y)$

conclude (by induction)

$$s(G-h, R, x, y) \geq s(G_{x\to y} - h, R, x, y).$$

This inequality also does not hold in general—Figure 4.1 serves as a counterexample.

4.3 Compression Acting on Spanning Forests

Threshold graphs also contain the minimizer of the number of k-component spanning forests, for there are fewer spanning forests in a graph's compression than the graph itself. Although not found in the literature, the proof is straightforward but its general structure will be used to prove Lemma 4.2.1 in the following section. Let $\mathcal{F}(G)$ be the set of spanning forests of G.

Lemma 4.3.1. *There exists some threshold graph $H \in \mathcal{G}_{n,e}$ such that*

$$|\mathcal{F}(H)| = \min_{H' \in \mathcal{G}_{n,e}} |\mathcal{F}(H')|.$$

Proof. We will first show $|\mathcal{F}(c(G))| \leq |\mathcal{F}(G)|$. For $i = 1, 2$, let $\mathcal{F}^{(i)}(G)$ and $\mathcal{F}^{(i)}(c(G))$ denote the spanning forests of G and $c(G)$ respectively such that x and y are in exactly

i components. Notice that

$$|\mathcal{F}^{(1)}(c(G))| = \sum_{\{x,y\}\subseteq S\subset V(G)} |\mathcal{T}(c(G)[S])| \cdot |\mathcal{F}(c(G)\backslash S)|,$$

and so Lemma 2.2.2 and Theorem 2.1.2 give

$$|\mathcal{F}^{(1)}(c(G))| \leq \sum_{\{x,y\}\subseteq S\subseteq V(G)} |\mathcal{T}(G[S])| \cdot |\mathcal{F}(G\backslash S)| = |\mathcal{F}^{(1)}(G)|.$$

For $F \in \mathcal{F}^{(2)}(c(G))$, since d preserves the number of vertices and edges, Lemma 2.2.3 implies $d(F) \in \mathcal{F}^{(2)}(G)$. Therefore by Corollary 2.2.1, $|\mathcal{F}^{(2)}(c(G))| \leq |\mathcal{F}^{(2)}(G)|$ and hence

$$|\mathcal{F}(c(G))| = |\mathcal{F}^{(1)}(c(G))| + |\mathcal{F}^{(2)}(c(G))| \leq |\mathcal{F}^{(1)}(G)| + |\mathcal{F}^{(2)}(G)| = |\mathcal{F}(G)|.$$

Select a graph H from $\mathcal{G}_{n,e}$ with $|\mathcal{F}(H)|$ minimal. Since compression does not increase the number of spanning forests, any compression of H is also minimal. Among these, pick H with $d_2(H)$ maximal and by Corollary 2.2.2, H is threshold. □

4.4 Compression Acting on Rooted Spanning Forests

We will now prove that, for all k, compression decreases the number of k-component rooted spanning forests. This shows that the result of Lemma 4.2.1 is correct, but the proof uses techniques unrelated to that of the original (incorrect) proof. Note that we follow the notation outlined in Section 2.2.

Proof. For $i = 1, 2$ let $\mathcal{F}_k^{(i)}(G)$ and $\mathcal{F}_k^{(i)}(c(G))$ denote the number of spanning forests in $\mathcal{F}_k(G)$ and $\mathcal{F}_k(c(G))$ respectively where x and y are in exactly i components. As in the proof of Lemma 4.3.1, we will show for $i = 1, 2$

$$\sum_{F \in \mathcal{F}_k^{(i)}(c(G))} \gamma(F) \leq \sum_{F \in \mathcal{F}_k^{(i)}(G)} \gamma(F) \tag{4.1}$$

which implies the result.

First consider the case where x and y are in exactly one component, i.e., $i = 1$. Notice that we can rewrite

$$\sum_{F \in \mathcal{F}^{(1)}(c(G))} \gamma(F) = \sum_{\{x,y\} \subseteq S \subseteq V(G)} |\mathcal{T}(c(G)[S])| \cdot |S| \cdot \left(\sum_{F \in \mathcal{F}_{k-1}(c(G) \setminus S)} \gamma(F) \right)$$

and similarly for G. Lemma 2.2.2 and Theorem 2.1.2 give

$$\sum_{\{x,y\} \subseteq S \subseteq V(G)} |\mathcal{T}(c(G)[S])| \cdot |S| \cdot \left(\sum_{F \in \mathcal{F}_{k-1}(c(G) \setminus S)} \gamma(F) \right) \leq \sum_{\{x,y\} \subseteq S \subseteq V(G)} |\mathcal{T}(G[S])| \cdot |S| \cdot \left(\sum_{F \in \mathcal{F}_{k-1}(G \setminus S)} \gamma(F) \right)$$

which implies the inequality in (4.1) for $i = 1$.

Now suppose that x and y are in exactly two components, i.e., $i = 2$. Let $r : E(c(G)) \to E(c(G))$ be the edge replacement function given by

$$r(e) = \begin{cases} e \Delta \{x, y\} & \text{if } e \cap \{x, y\} \neq \emptyset \text{ and } e \cap A_{xy} \neq \emptyset \\ e & \text{otherwise.} \end{cases}$$

Let $F \in \mathcal{F}^{(2)}(c(G))$, then $r(F)$ is defined so that

$$N_{r(F)}(x) \cap A_{xy} = N_F(y) \cap A_{xy} \qquad \text{and} \qquad N_{r(F)}(y) \cap A_{xy} = N_F(x) \cap A_{xy}.$$

Note that since $r(r(F)) = F$, the set of pairs $\{F, r(F)\}$ partitions the set $\mathcal{F}^{(2)}(c(G))$.

Thus for each $\{F, r(F)\}$ we will show

$$\gamma(F) + \gamma(r(F)) \leq \gamma(d(F)) + \gamma(d(r(F))).$$

Let

$$\begin{aligned}
a &= \sum_{z \in N_F(y) \cap A_{x\overline{y}}} |V((F \backslash \{x, y\})_z)| \\
b &= \sum_{z \in N_F(y) \cap A_{\overline{x}y}} |V((F \backslash \{x, y\})_z)| \\
c_x &= \sum_{z \in N_F(x)} |V((F \backslash \{x, y\})_z)| \\
c_y &= \sum_{z \in N_F(y) \cap A_{xy}} |V((F \backslash \{x, y\})_z)|.
\end{aligned}$$

Then

$$\begin{aligned}
\gamma(F) + \gamma(r(F)) &= (c_x + 1)(c_y + a + b + 1) + (c_y + 1)(c_x + a + b + 1) \\
&= (c_x + a + 1)(c_y + b + 1) + (c_y + a + 1)(c_x + b + 1) - 2ab \\
&\leq (c_x + a + 1)(c_y + b + 1) + (c_y + a + 1)(c_x + b + 1) \\
&= \gamma(d(F)) + \gamma(d(r(F))).
\end{aligned}$$

Therefore

$$\sum_{F \in \mathcal{F}_k^{(2)}(c(G))} \gamma(F) \leq \sum_{F \in \mathcal{F}_k^{(2)}(G)} \gamma(F)$$

and so

$$\sum_{F \in \mathcal{F}_k(c(G))} \gamma(F) \leq \sum_{F \in \mathcal{F}_k(G)} \gamma(F).$$

$\square$

In particular, the inequality in Lemma 4.2.1 is strict unless one of $A_{x\overline{y}}$ or $A_{\overline{x}y}$ are empty, i.e., $G_{x\to y} \cong G$.

4.5 Proof of Theorem 4.1.2

Recall that Theorem 4.1.2, our main theorem in this chapter, is that there exists a threshold graph in $\mathcal{G}_{n,e}$ with the minimum number of k-component rooted spanning forests for all k.

Proof of Theorem 4.1.2. Select a graph H from $\mathcal{G}_{n,e}$ with $c_k(G)$ minimal for all k. By Lemma 4.2.1, compression strictly decreases the Laplacian coefficients of H unless H is threshold. $\square$

4.6 Future Directions

There are still many open problems related to this question. The clear open question is which graphs attain the minimum value of this set of parameters. We know that it is not true for all n and e that one graph will minimize c_k for all k. In some cases, the minimum value is either achieved by the lex or colex graph.

Definition 4.6.1. The *lexicographic order*, $<_L$, on finite subsets of $\mathbb{N}$ is defined by $A <_L B$ if $\min(A\Delta B) \in A$. The *lex graph* $\mathcal{L}(n,e)$ is the graph with vertex set $[n]$ and edge set consisting of the first e edges in lex order on $E(K_n)$.

For all graphs in $\mathcal{G}_{n,e}$, $c_n = 1$ and $c_{n-1} = 2e$. The coefficient c_{n-2} counts four times the number of 2-matchings and three times the number of paths of length 2. Thus

$$c_{n-2}(G) = 4\binom{e}{2} - \sum_{v \in V(G)} \binom{d(v)}{2} = 2e^2 - e - \frac{d_2(G)}{2},$$

and it was shown in [1] that d_2 is maximized by either the lex or colex graph. For $e \leq \binom{n-1}{2}$, $\mathcal{C}(n,e)$ contains no spanning trees and therefore is a minimizer of c_1. In [2], Bogdanowicz proved Boesch's Conjecture which stated $\mathcal{L}(n,e)$ has the minimum number of spanning trees over all connected graphs in $\mathcal{G}_{n,e}$ and thus is the minimizer of c_1. Therefore the first interesting cases to investigate would be c_2 and c_{n-3}. Furthermore, a few graphs on 11 vertices have coefficient $c_{n-3} = c_8$ smaller than that of both the lex and colex graph on the same number of vertices and edges. Given k, it seems that it is a non-trivial problem to determine the graph $G \in \mathcal{G}_{n,e}$ for which $c_k(G)$ is minimum.

Chapter 5

Hamilton Cycles

A decision problem is one that can be posed as a yes-no question of the input. In this final chapter we determine that certain decision problems regarding Hamiltonicity in hypergraphs are NP-complete. In particular we restrict the problem by imposing a lower bound on the minimum vertex degree of the hypergraphs considered. The decision problems in question are that of determining the existence of a Hamilton 1-cycle and a Hamilton 2-cycle.

5.1 Definitions and Results

Recall that a k-graph, C is an *ℓ-cycle* if its vertices can be cyclically ordered in such a way that each edge of C consists of k consecutive vertices and each pair of consecutive edges overlaps in exactly ℓ vertices. Then C is a Hamilton ℓ-cycle of a hypergraph H if C is a subhypergraph of H and $V(C) = V(H)$. Suppose we are seeking a Hamilton ℓ-cycle in a k-graph. If we start with an edge e, we know that the next edge must contain at least ℓ vertices from e. Thus, it would be nice to know that for every ℓ elements of the vertex set, there are many edges which contain all ℓ elements. In an n-vertex k-graph, there are at most $\binom{n-\ell}{k-\ell}$ edges which contain those ℓ elements. Motivated by this observation, we generalize the decision problem in Theorem 1.2.3

to a problem which imposes a minimum ℓ-degree condition on the input graphs.

Definition 5.1.1. Let $\mathrm{HAM}_\ell(k,c)$ be the problem of deciding the existence of a Hamilton ℓ-cycle in a k-uniform hypergraph $\mathcal{H}$ with

$$\delta_\ell(\mathcal{H}) \geq c\binom{|V(\mathcal{H})|-\ell}{k-\ell}.$$

The case $\ell = k-1$ is equivalent to $\mathrm{HAM}(k,c)$ studied in [21]. Here we consider the special case of $\ell = 1$ and hence we are restricting the minimum vertex degree of the input graphs. We prove the following theorem whose proof can be found in Section 5.2.

Theorem 5.1.1. *For all $k \geq 2$ and $c < \frac{(k-2)!}{2(k-1)^{k-1}}$, $\mathrm{HAM}_1(k,c)$ is NP-complete.*

We also investigate Hamilton 2-cycles in k-graphs, but were challenged by the condition on δ_2. Instead, we have the following result which restricts the vertex degree as opposed to δ_2.

Proposition 5.1.1. *For all $c < \frac{3(k-3)!}{(k-1)(k-2)^{k-2}}$, the problem of deciding the existence of a Hamilton 2-cycle in a k-graph $\mathcal{H}$ with*

$$\delta_1(\mathcal{H}) \geq c\binom{|V(\mathcal{H})|-1}{k-2}$$

is NP-complete.

We postpone the proof of Proposition 5.1.1 to Section proofham2.

Before we move on, we first we remark that an *instance* in a decision problem is an input which satisfies the hypothesis. For the problem $\mathrm{HAM}_\ell(k,C)$, an instance would be a k-graph $\mathcal{H}$ with

$$\delta_\ell(\mathcal{H}) \geq c\binom{|V(\mathcal{H})| - \ell}{k-\ell}.$$

A YES-instance in a decision problem is an instance which outputs 'YES', and similarly for a NO-instance.

There are two statements to prove in a general approach to proving that a decision problem, call it X, is NP-complete. The first of which is to prove that the decision problem itself can be verified in polynomial time. In our case, we need to show that given a k-graph $\mathcal{H}$ and a list of vertices, checking that the list is a Hamilton ℓ-cycle of $\mathcal{H}$ can be done in polynomial time. Indeed, first check that the list of vertices is a permutation of the vertex set of $\mathcal{H}$. Then check that every $\frac{n}{k-\ell}$ cyclic shifts by ℓ of k consecutive vertices are indeed edges of $\mathcal{H}$.

We must also prove that problem X is just as hard as some other NP-complete problem, call it Y. Phrased another way, show that if there is a polynomial-time algorithm for problem X, there is also a polynomial-time algorithm for problem Y. We do this by taking an instance of Y and transforming it by a polynomial-time algorithm, to an instance of X. Finally, one must prove that an instance of Y is a YES-instance if and only if it is transformed by this algorithm to a YES-instance of X. In the following proofs, it is often the case that the NP-complete problem we use for Y is HAM$(2, \frac{1}{2} - \epsilon)$ for some $\epsilon > 0$.

5.2 Loose Hamilton Cycles

Recall that Theorem 5.1.1 states that the problem of determining whether or not a hypergraph with certain minimum vertex degree conditions contains a Hamilton $1-$cycle is NP-complete.

Proof of Theorem 5.1.1. We induct on k. Theorem 1.2.1 proves that the result is true for $k = 2$.

Now suppose $k \geq 3$, $\epsilon > 0$ and set $\epsilon' = \epsilon \left(\frac{k-2}{k-1}\right)^{k-1}$. Let $\mathcal{H}_{k-1}$ be an instance of $\mathrm{HAM}_1(k-1, \frac{\prod_{i=1}^{k-3} i}{2(k-2)^{k-2}} - \epsilon')$ where $V(\mathcal{H}_{k-1}) =: A$ and $E(\mathcal{H}_{k-1}) =: E_{k-1}$. Let B be a set of size $\frac{n}{k-2}$ disjoint from A and define $\mathcal{H}_k$ to be the k-uniform hypergraph on vertex set $A \sqcup B$ and edge set

$$E_k := \{b \cup e \,:\, e \in E_{k-1},\, b \in B\}.$$

For $x \in A$ we have

$$d_1(x) \geq \left(\frac{\prod_{i=1}^{k-3} i}{2(k-2)^{k-2}} - \epsilon'\right) \binom{n-1}{k-2} \frac{n}{k-2}$$

and for $y \in B$ we have

$$d_1(y) \geq \frac{n}{k-1} \left(\frac{\prod_{i=1}^{k-3} i}{2(k-2)^{k-2}} - \epsilon'\right) \binom{n-1}{k-2}.$$

Since $d_1(x) \geq d_1(y)$ we need only check that

$$d_1(y) \geq \left(\frac{\prod_{i=1}^{k-2} i}{2(k-1)^{k-1}} - \epsilon\right) \binom{\frac{k-1}{k-2}n - 1}{k-1}.$$

Note that

$$
\begin{aligned}
d_1(y) &\geq \frac{n}{k-1}\left(\frac{(k-3)!}{2(k-2)^{k-2}}-\epsilon'\right)\binom{n-1}{k-2}\\
&= \frac{n_{(k-1)}}{(k-1)!}\left(\frac{(k-3)!}{2(k-2)^{k-2}}-\epsilon'\right)\\
&= \frac{n_{(k-1)}}{\left(\frac{k-1}{k-2}n-1\right)_{(k-1)}}\left(\frac{(k-3)!}{2(k-2)^{k-2}}-\epsilon'\right)\binom{\frac{k-1}{k-2}n-1}{k-1}\\
&= \frac{n(n-1)\cdots(n-k+2)}{\left(\frac{k-1}{k-2}\right)^{k-1}(n-\frac{k-2}{k-1})\cdots(n-k\frac{k-2}{k-1}+\frac{k-2}{k-1})}\left(\frac{(k-3)!}{2(k-2)^{k-2}}-\epsilon'\right)\binom{\frac{k-1}{k-2}n-1}{k-1}\\
&= \frac{\prod_{j=0}^{k-2}(n-j)}{\prod_{j=0}^{k-2}\left(n-(j+1)\frac{k-2}{k-1}\right)}\left(\frac{(k-2)!}{2(k-1)^{k-1}}-\epsilon\right)\binom{\frac{k-1}{k-2}n-1}{k-1}.
\end{aligned}
$$

Claim.

$$
\prod_{j=0}^{k-2}(n-j) \geq \prod_{j=0}^{k-2}\left(n-(j+1)\frac{k-2}{k-1}\right)
$$

Proof of Claim. For $0 \leq j \leq k-2$,

$$
\begin{aligned}
(k-1-(k-2))j &\leq k-2\\
\left(1-\frac{k-2}{k-1}\right)j &\leq \frac{k-2}{k-1}\\
n-j\left(\frac{k-2}{k-1}\right)-\frac{k-2}{k-1} &\leq n-j\\
n-(j+1)\left(\frac{k-2}{k-1}\right) &\leq n-j.
\end{aligned}
$$

□

Therefore

$$
\begin{aligned}
d_1(y) &= \frac{\prod_{j=0}^{k-2}(n-j)}{\prod_{j=0}^{k-2}\left(n-(j+1)\frac{k-2}{k-1}\right)}\left(\frac{(k-2)!}{2(k-1)^{k-1}}-\epsilon'\right)\binom{\frac{k-1}{k-2}n-1}{k-1}\\
&\geq \left(\frac{(k-2)!}{2(k-1)^{k-1}}-\epsilon'\right)\binom{\frac{k-1}{k-2}n-1}{k-1},
\end{aligned}
$$

hence $\mathcal{H}_k$ is an instance of $\mathrm{HAM}_1(k, \frac{\prod_{i=1}^{k-2} i}{2(k-1)^{k-1}} - \epsilon)$.

Now we will show $\mathcal{H}_{k-1}$ is a YES-instance of $\mathrm{HAM}_1(k-1, \frac{(k-3)!}{2(k-2)^{k-2}} - \epsilon')$ if and only if $\mathcal{H}_k$ is a YES-instance of $\mathrm{HAM}_1(k, \frac{(k-2)!}{2(k-1)^{k-1}} - \epsilon)$. Suppose $v_1 \cdots v_n$ is a loose Hamilton cycle in $\mathcal{H}_{k-1}$, i.e., for $i \in \left[\frac{n}{k-2}\right]$, $\{v_{(i-1)(k-2)+1}, \ldots, v_{(i-1)(k-2)+k-1}\} \in E_{k-1}$. Then consider

$$v_1 b_1 \cdots v_{k-1} b_2 \cdots v_{n-k+3} b_{\frac{n}{k-2}} \cdots v_n$$

and relabel the vertices $u_1 \cdots u_{\frac{k-1}{k-2}n}$ where for $i \in \left[\frac{n}{k-2}\right]$,

$$u_{(i-1)(k-1)+1} = v_{(i-1)(k-2)+1} \text{ and } u_{(i-1)(k-1)+2} = b_i,$$

and for $j \in \{3, \ldots, k-1\}$

$$u_{(i-1)(k-1)+j} = v_{(i-1)(k-2)+j-1}.$$

Note for $i \in \left[\frac{n}{k-2}\right]$,

$$\{u_{(i-1)(k-1)+1}, \ldots, u_{(i-1)(k-1)+k}\} = \{v_{(i-1)(k-2)+1}, b_i, \ldots, v_{(i-1)(k-2)+k-1}\} \in E_k$$

since

$$\{v_{(i-1)(k-2)+1}, \ldots, v_{(i-1)(k-2)+k-1}\} \in E_{k-1}.$$

Therefore $u_1 \cdots u_{\frac{k-1}{k-2}n}$ is a loose Hamilton cycle in $\mathcal{H}_k$.

Now suppose $u_1 \cdots u_{\frac{k-1}{k-2}n}$ is a loose Hamilton cycle in $\mathcal{H}_k$. In order to show this implies existence of a loose Hamilton cycle in $\mathcal{H}_{k-1}$, it suffices to show no element from B is double covered (i.e., no element from B lies in 2 edges of the loose Hamilton cycle). But since each edge only contains one element of B and there are exactly $|B|$ edges in a loose Hamilton cycle for $\mathcal{H}_k$, this implies no element of B can be double covered. Therefore $\mathcal{H}_{k-1}$ has a loose Hamilton cycle given by $(u_i : u_i \in A)$. $\square$

5.3 2-Hamilton Cycles

In what follows we prove Proposition 5.1.1 which states that the problem of determining whether or not a hypergraph with certain minimum vertex degree conditions contains a Hamilton 2−cycle is NP-complete. We will approach this by first establishing a base case of $k = 4$ and then proceed by induction on k.

Lemma 5.3.1. *For all $c < \frac{1}{4}$, the problem of deciding the existence of a Hamilton 2-cycle in a 4-graph $\mathcal{H}$ with*

$$\delta_1(\mathcal{H}) \geq c\binom{|V(\mathcal{H})| - 1}{2}$$

is NP-complete.

Proof. Let $\epsilon > 0$ and G be an arbitrary instance of $\mathrm{HAM}(2, \frac{1}{2} - \epsilon')$ with $\epsilon' \geq 2\epsilon$. Call $A := V(G)$ and $E_2 := E(G)$. Label the vertices of G as $\{v_1, \ldots, v_n\}$ and let $B = \{b_1, \ldots, b_n\}$ be a set of size n disjoint from A. Define the 4-uniform hypergraph $\mathcal{H}_4$ on the vertex set $A \sqcup B$ and edge set

$$E_4 := \{\{v_i, v_j, b_i, b_k\} : \{v_i, v_j\} \in E_2,\, j \neq 1,\, k \in [n]\backslash\{i\},\, b_i, b_k \in B\}.$$

Note that $d(v_1) \geq \left(\frac{1}{2} - \epsilon'\right) n(n-1)$, and for $j \geq 2$ (and n large enough)

$$d(v_j) \geq \left(\left(\frac{1}{2} - \epsilon'\right) n - 1\right)(n-1) + \left(\frac{1}{2} - \epsilon'\right) n(n-2) \geq d(v_1).$$

Also,

$$d(b_1) \geq \left(\frac{1}{2} - \epsilon'\right) n(n-1)$$

and for $j \geq 2$

$$d(b_j) \geq \left(\left(\frac{1}{2} - \epsilon'\right) n - 1\right)(n-1) + (n-2)\left(\left(\frac{1}{2} - \epsilon'\right) n - 1\right) \geq d(v_1).$$

Then

$$\begin{aligned}
d(v_1) &\geq \left(\frac{1}{2} - \epsilon'\right) n(n-1) \\
&= \frac{2n(n-1)}{(2n-1)(2n-2)} \left(\frac{1}{2} - \epsilon'\right) \binom{2n-1}{2} \\
&= \frac{n(n-1)}{2(n-\frac{1}{2})(n-1)} \left(\frac{1}{2} - \epsilon'\right) \binom{2n-1}{2} \\
&\geq \frac{1}{2} \left(\frac{1}{2} - \epsilon'\right) \binom{2n-1}{2} \\
&= \left(\frac{1}{4} - \epsilon\right) \binom{2n-1}{2}
\end{aligned}$$

and hence $\mathcal{H}_4$ is an instance of of $\mathrm{LHC}_2(4, \frac{1}{4})$.

Now we will show G is a YES-instance of $\mathrm{HAM}(2, \frac{1}{2} - \epsilon')$ if and only if $\mathcal{H}_4$ is a YES-instance of $\mathrm{LHC}_2(4, \frac{1}{4} - \epsilon)$. Suppose $u_1 \cdots u_n$ is a Hamilton cycle in G, then since $\{v_i, v_i, v_j, b_j\} \in E_4$ for all $i \neq j$,,

$$u_1 b_1 u_2 b_2 \cdots u_n b_n$$

is a Hamilton 2-cycle in $\mathcal{H}_4$.

Now assume $u_1 \cdots u_{2n}$ is a Hamilton 2-cycle in $\mathcal{H}_4$. We will show that given any integer $1 \leq \ell \leq n$, we have $u_{2\ell-1} \in A$ and $u_{2\ell} \in B$. Note that since each edge of $\mathcal{H}_4$ contains exactly two elements from each of A and B, there exists some ℓ such that

$u_{2\ell-1} = b_i$ and $u_{2\ell} = b_j$ if and only if there exists some ℓ' such that $u_{2\ell'-1} \in A$ and $u_{2\ell'} \in A$. Thus it suffices to show there does not exist an ℓ such that $u_{2\ell-1} = b_i$ and $u_{2\ell} = b_j$. Suppose otherwise, then

$$\{u_{2\ell-3}, u_{2\ell-2}, u_{2\ell-1}, u_{2\ell}\}, \{u_{2\ell-1}, u_{2\ell}, u_{2\ell+1}, u_{2\ell+2}\} \in E_4$$

implies

$$u_{2\ell-3}, u_{2\ell-2}, u_{2\ell+1}, u_{2\ell+2} \in A.$$

Furthermore by definition of E_4, since

$$\{u_{2\ell-3}, u_{2\ell-2}, u_{2\ell-1}, u_{2\ell}\}, \{u_{2\ell-1}, u_{2\ell}, u_{2\ell+1}, u_{2\ell+2}\} \in E_4$$

then $u_{2(\ell-1)-1}, u_{2(\ell-1)}, u_{2(\ell+1)+1}, u_{2(\ell+1)+1} \in A$ and by induction we may assume for all integers $0 \leq k \leq \frac{n}{2} - 2$,

$$u_{4k+1}, u_{4k+2} \in A \text{ and } u_{4k+3}, u_{4k+4} \in B.$$

Now there exists some k such that $v_1 = u_{4k+1}$. Then v_1 is contained in the edge $\{u_{4k+1}, u_{4k+2}, u_{4k+3}, u_{4k+4}\}$ and $\{u_{4k}, u_{4k-1}, u_{4k+1}, u_{4k+2}\}$. By definition of E_4, every edge which contains v_1 must also contain b_1. But u_{4k+2} is the only vertex in both edges along with x and since $u_{4k+2} \in A$, this yields a contradiction. Therefore every vertex $v \in A$ is contained in exactly two edges, which implies $(u_i : i \in A)$ is a Hamilton 2-cycle in G. $\square$

Proof of Proposition 5.1.1. This proof follows that of Proposition 5.1.1. We induct on k. Lemma 5.3.1 proves that the result is true for $k = 4$. Assume $k \geq 5$ and that the statement holds for $k-1$. Let $\epsilon > 0$ and set $\epsilon' = \frac{(k-2)^{k-3}}{(k-3)^{k-2}}\epsilon$. Let $\mathcal{H}_{k-1}$ be an instance of $\mathrm{LHC}_2(k-1, \frac{3(k-4)!}{(k-2)(k-3)^{k-3}} - \epsilon')$ where $V(\mathcal{H}_{k-1}) =: A$ and $E(\mathcal{H}_{k-1}) =: E_{k-1}$. Let B be a set of size $\frac{n}{k-3}$ disjoint from A and define $\mathcal{H}_k$ to be the k-graph on vertex set $A \sqcup B$ and edge set

$$E_k := \{b \cup e \,:\, e \in E_{k-1},\, b \in B\}.$$

For $x \in A$ we have

$$d_1(x) \geq \left(\frac{3(k-4)!}{(k-2)(k-3)^{k-3}} - \epsilon'\right)\binom{n-1}{k-3}\frac{n}{k-3}$$

and for $y \in B$ we have

$$d_1(y) \geq \frac{n}{k-1}\left(\frac{3(k-4)!}{(k-2)(k-3)^{k-3}} - \epsilon'\right)\binom{n-1}{k-3}.$$

Since $d_1(x) \geq d_1(y)$ we need only check that

$$d_1(y) \geq \left(\frac{3(k-3)!}{(k-1)(k-2)^{k-2}} - \epsilon\right)\binom{n-1}{k-2}.$$

Note that

$$\begin{aligned}
d_1(y) &\geq \frac{n}{k-1}\left(\frac{3(k-4)!}{(k-2)(k-3)^{k-3}} - \epsilon'\right)\binom{n-1}{k-3} \\
&= \frac{n_{(k-2)}}{(k-2)!}\left(\frac{3(k-4)!}{(k-1)(k-3)^{k-3}} - (k-2)\epsilon'\right) \\
&= \frac{n_{(k-2)}}{\left(\frac{k-2}{k-3}n-1\right)_{k-2}}\left(\frac{3(k-4)!}{(k-1)(k-3)^{k-3}} - (k-2)\epsilon'\right)\binom{\frac{k-2}{k-3}n-1}{k-2} \\
&= \frac{\prod_{j=0}^{k-2}(n-j)}{\prod_{j=0}^{k-2}\left(n-(j+1)\frac{k-3}{k-2}\right)}\left(\frac{3(k-3)!}{(k-1)(k-2)^{k-2}} - \frac{(k-3)^{k-2}}{(k-2)^{k-3}}\epsilon'\right)\binom{\frac{k-2}{k-3}n-1}{k-2}
\end{aligned}$$

Claim.

$$\prod_{j=0}^{k-2}(n-j) \geq \prod_{j=0}^{k-2}\left(n-(j+1)\frac{k-3}{k-2}\right)$$

Proof of Claim. For $0 \le j \le k-3$, First note

$$\begin{aligned}(k-2-(k-3))j &\le k-3 \\ \left(1-\frac{k-3}{k-2}\right) j &\le \frac{k-3}{k-2} \\ n-j\left(\frac{k-3}{k-2}\right)-\frac{k-3}{k-2} &\le n-j \\ n-(j+1)\left(\frac{k-3}{k-2}\right) &\le n-j.\end{aligned}$$

Then for $k \ge 5$ and n large enough,

$$n(n-k+2) \ge \left(n-\frac{k-3}{k-2}\right)\left(n-\frac{(k-1)(k-3)}{k-2}\right).$$

$\square$

Therefore

$$\begin{aligned}d_1(y) &= \frac{\prod_{j=0}^{k-2}(n-j)}{\prod_{j=0}^{k-2}\left(n-(j+1)\frac{k-3}{k-2}\right)}\left(\frac{3(k-3)!}{(k-1)(k-2)^{k-2}}-\frac{(k-3)^{k-2}}{(k-2)^{k-3}}\epsilon'\right)\binom{\frac{k-2}{k-3}n-1}{k-2} \\ &\ge \left(\frac{3(k-3)!}{(k-1)(k-2)^{k-2}}-\epsilon\right)\binom{\frac{k-3}{k-2}n-1}{k-2},\end{aligned}$$

hence $\mathcal{H}_k$ is an instance of $\mathrm{LHC}_2(k, \frac{3(k-3)!}{(k-1)(k-2)^{k-2}}-\epsilon)$.

Now we will show $\mathcal{H}_{k-1}$ is a YES-instance of $\mathrm{LHC}_2(k-1, \frac{3(k-3)!}{(k-3)(k-3)^{k-3}}-\epsilon')$ if and only if $\mathcal{H}_k$ is a YES-instance of $\mathrm{LHC}_2(k, \frac{3(k-3)!}{(k-1)(k-2)^{k-2}}-\epsilon)$. Suppose $v_1\cdots v_n$ is a Hamilton 2-cycle in $\mathcal{H}_{k-1}$, i.e., for $i \in \left[\frac{n}{k-3}\right]$, $\{v_{(i-1)(k-3)+1},\ldots,v_{(i-1)(k-3)+k-1}\} \in E_{k-1}$. Then consider

$$v_1v_2b_1\cdots v_{k-1}b_2\cdots v_{n-k+4}b_{\frac{n}{k-3}}\cdots v_n$$

and relabel the vertices $u_1 \cdots u_{\frac{k-2}{k-3}n}$ where for $i \in \left[\frac{n}{k-3}\right]$,

$$u_{(i-1)(k-2)+1} = v_{(i-1)(k-3)+1}$$

$$u_{(i-1)(k-2)+2} = v_{(i-1)(k-3)+2}$$

$$u_{(i-1)(k-2)+3} = b_i$$

and for $j \in \{4, \ldots, k-2\}$

$$u_{(i-1)(k-2)+j} = v_{(i-1)(k-3)+j-1}.$$

Note for $i \in \left[\frac{n}{k-3}\right]$,

$$\left\{u_{(i-1)(k-2)+1}, \ldots, u_{(i-1)(k-2)+k}\right\} = \left\{v_{(i-1)(k-3)+1}, v_{(i-1)(k-3)+2}, b_i, \ldots, v_{(i-1)(k-3)+k-1}\right\} \in E_k$$

since

$$\left\{v_{(i-1)(k-3)+1}, \ldots, v_{(i-1)(k-3)+k-1}\right\} \in E_{k-1}.$$

Therefore $u_1 \cdots u_{\frac{k-2}{k-3}n}$ is a Hamilton 2-cycle in $\mathcal{H}_k$.

Now suppose $u_1 \cdots u_{\frac{k-2}{k-3}n}$ is a Hamilton 2-cycle in $\mathcal{H}_k$. In order to show this implies existence of a Hamilton 2-cycle in $\mathcal{H}_{k-1}$, it suffices to show no element from B is double covered (i.e., no element from B lies in 2 edges of the Hamilton 2-cycle). But since each edge only contains one element of B and there are exactly $|B|$ edges in a Hamilton 2-cycle for $\mathcal{H}_k$, this implies no element of B can be double covered. Therefore $\mathcal{H}_{k-1}$ has a Hamilton 2-cycle given by $(u_i : u_i \in A)$. $\square$

5.4 Future Directions

Certainly, there is much more to do regarding the decision problem $\mathrm{HAM}_\ell(k,c)$. We also believe that one should be able to improve the minimum vertex degree condition in Proposition 5.1.1 to $c\binom{|V(\mathcal{H})|-1}{k-1}$ for some constant c. There is also still the question of generalizing the result of Theorem 1.2.3 to still restricting codegree, but determining the existence of Hamilton ℓ-cycles for other ℓ.

What I am particularly interested in is a result in [21] which proves that for graphs which do satisfy the codegree conditions of Theorem 1.2.2 with $\ell = k-1$, the Hamilton $(k-1)$-cycle can be found in polynomial time. This typically involves analyzing the proof of the existence of the Hamilton cycle and using derandomization algorithms to make a fully constructive proof. At the time, the proof of the existence of the cycle used Szemeredi's Regularity Lemma, which guarantees very nice structure in the graph but at the cost of an astronomically large vertex set. What was most interesting about the polynomial-time result in [21] was that they were able to develop an algorithm which finds the cycle without applying the Regularity Lemma. The proof of Theorem 1.2.2 in its full general form uses the Hypergraph Regularity Lemma, which is a more general version of Szemeredi's Regularity Lemma. It was proven in [30], that there is a polynomial time algorithm for the Hypergraph Regularity Lemma but one should note that this still requires an extremely large vertex set. Using the Hypergraph Regularity Lemma algorithm, we can simply employ derandomization techniques in the proof of Theorem 1.2.2 in order to show that one can find a Hamilton ℓ-cycle in polynomial time. But the value in the solution to this problem would really come from a proof that avoids the use of the Hypergraph Regularity Lemma, hence making the "finitely" many cases a computer would need to check by brute-force a reasonable amount.

Chapter 6

Glossary

- **ℓ-cycle**: C is an ℓ*-cycle* if its vertices can be cyclically ordered in such a way that each edge of C consists of k consecutive vertices and each pair of consecutive edges overlaps in exactly ℓ vertices.

- **k-graph**: a hypergraph is a k-graph, or a k-uniform hypergraph, if its edges are all of size k.

- **Acyclic graph**: an acyclic graph is a graph containing no cycles.

- **Adjacency matrix of a graph**: for a graph with vertex set V, its adjacency matrix is a $|V| \times |V|$ matrix A such that the element $A_{ij} = 1$ if vertices v_i and v_j are adjacent, and is 0 otherwise.

- **Adjacent vertices**: two vertices, u and v, of a graph are adjacent if and only if they form an edge in that graph and we denote this by $u \sim v$.

- **Bicyclic graph**: a bicyclic graph is a graph with exactly two cycles.

- **Complete graph**: the complete graph on n vertices, denoted K_n, is a graph wherein every pair of vertices are form an edge.

- **(Connected) Component of a graph**: a connected component of a graph G is a subgraph in which every pair of vertices are connected by a path and which is connected to no additional vertices of G

- **Connected graph**: a graph is connected if every pair of vertices can be connected by a path in the graph.

- **Cycle graph**: a graph is a cycle if there exists an ordering of the vertices $v_1 \cdots v_n$ such that its edges are $v_i v_{i+1}$ for $i = 1, \ldots, n-1$ and $v_1 v_n$.

- **Degree of a vertex**: the degree of a vertex v in a graph, denoted $d(v)$, is the number of edges which contain it.

- **Forest**: a forest is a disjoint union of trees.

- **Hypergraph**: a hypergraph is a pair of sets $H = (V, E)$ where E is some subset of the power set of V.

- **Incident edges**: two edges e_1 and e_2 of a graph are incident if $e_1 \cap e_2 \neq \emptyset$.

- **Induced subgraph**: a subgraph H of G is an induced subgraph if $uv \in E(G) \backslash E(H)$ implies $u \notin V(H)$ or $v \notin V(H)$.

- **Graph**: a graph is a pair of sets $G = (V, E)$ where $E \subseteq \{\{u, v\} : u, v \in V, \text{ and } u \neq v\}$.

- **Hamilton cycle**: a spanning subgraph which is a cycle.

- **Hypergraph**: a hypergraph H is a pair (V, E) such that E is any collection of subsets of V.

- **Neighborhood of a vertex**: the neighborhood of a vertex in a graph, denoted $N(v)$, is the set of vertices which are adjacent to it.

- **Order of a graph**: the order of a graph is the number of edges it has.
- **Path graph**: a graph is a path if there exists an ordering of the vertices $v_1 \cdots v_n$ such that its edges are $v_i v_{i+1}$ for $i = 1, \dots, n-1$.
- **Size of a graph**: the size of a graph is the number of vertices it has.
- **Spanning subgraph**: a subgraph H of G is spanning if $V(H) = V(G)$.
- **Subgraph**: H is a subgraph of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.
- **Tree**: a tree is a connected acyclic graph.
- **Tricyclic graph**: a tricyclic graph is a graph with exactly 3 cycles.
- **Unicyclic graph**: a unicyclic graph is a graph with exactly 1 cycle.

Bibliography

[1] R. Ahlswede and G.O.H. Katona. Graphs with maximal number of adjacent pairs of edges. *Acta Mathematica Academiae Scientiarum Hungarica*, 32(1):97–120, Mar 1978.

[2] Z. Bogdanowicz. Undirected simple connected graphs with minimum number of spanning trees. *Discrete Mathematics*, 309(10):3074 – 3082, 2009.

[3] B. Bollobás. *Modern Graph Theory*. Graduate Texts in Mathematics. Springer New York, 1998.

[4] V. M. Chelnokov and A. Kelmans. A certain polynomial of a graph and graphs with an extremal number of trees. *Journal of Combinatorial Theory, Series B*, 16(3):197 – 214, 1974.

[5] P. Csikvári. Applications of the kelmans transformation: extremality of the threshold graphs. *Electr. J. Comb.*, 18, 09 2011.

[6] J. Cutler and A.J. Radcliffe. Extremal graphs for homomorphisms. *Journal of Graph Theory*, 67:261–284, 2011.

[7] J. Cutler and A.J. Radcliffe. Extremal problems for independent set enumeration. *Electr. J. Comb.*, 18(1), 2011.

[8] E. Dahlhaus, P. Hajnal, and M. Karpiński. On the parallel complexity of hamiltonian cycle and matching problem on dense graphs. *Journal of Algorithms*, 15(3):367–384, Nov 1993.

[9] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

[10] K. Ecker and S. Zaks. On a graph labelling problem. *Gesellschraft für Mathematik und Datenverarbeitung MBH, Bonn*, Bericht 99, 1977.

[11] F. Garbe and R. Mycroft. The complexity of the hamilton cycle problem in hypergraphs of high minimum codegree. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, volume 47 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1–13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[12] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York, 1980.

[13] J. Guo, S. Liu, X. Pai, and S. Liu. On the laplacian coefficients of tricyclic graphs. *Journal of Mathematical Analysis and Applications*, 405(1):200 – 208, 2013.

[14] I. Gutman and L. Pavlović. On the coefficients of the laplacian characteristic polynomial of trees. *Bull. Acad. Serbe Sci. Arts (Cl. Math. Natur.)*, (28):31–40, 2003.

[15] H. Hàn and M. Schacht. Dirac-type results for loose hamilton cycles in uniform hypergraphs. *Journal of Combinatorial Theory, Series B*, 100(3):332 – 346, 2010.

[16] J. Han and Y. Zhao. Forbidding hamilton cycles in uniform hypergraphs. *Journal of Combinatorial Theory, Series A*, 143:107 – 115, 2016.

[17] C. He and H. Shan. On the laplacian coefficients of bicyclic graphs. *Discrete Mathematics*, 310(23):3404 – 3412, 2010.

[18] A. Ilić. Trees with minimal laplacian coefficients. *Computers & Mathematics with Applications*, 59(8):2776 – 2783, 2010.

[19] A. Ilić and D. Stevanović. On the laplacian coefficients of unicyclic graphs. *Linear Algebra and its Applications*, 430(8):2290 – 2300, 2009.

[20] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[21] M. Karpiński, A. Ruciński, and E. Szymańska. Computational complexity of the hamiltonian cycle problem in dense hypergraphs. *LATIN 2010: Theoretical Informatics, 9th Latin American Symposium, Oaxaca, Mexico*, Proceedings:662–673, April 19-23, 2010 2010.

[22] P. Keevash, D. Kühn, R. Mycroft, and D. Osthus. Loose hamilton cycles in hypergraphs. *Discrete Mathematics*, 311(7):544 – 559, 2011.

[23] A. Kelmans. Operations on graphs increasing some graph parameters. *arXiv:1109.4622v3*, 2013.

[24] L. Keough and A.J. Radcliffe. Graphs with the fewest matchings. *Combinatorica*, 36(6):703–723, Dec 2016.

[25] L. Keough and A.J. Radcliffe. Extremal threshold graphs for matchings and independent sets. *arXiv*, 2017.

[26] T. Kloks, D. Kratsch, and H. Müller. Finding and counting small induced subgraphs efficiently. *Information Processing Letters*, 74(3-4):115–121, 2000.

[27] D. Kühn, R. Mycroft, and D. Osthus. Hamilton ℓ-cycles in uniform hypergraphs. *Journal of Combinatorial Theory, Series A*, 117(7):910 – 927, 2010.

[28] D. Kühn and D. Osthus. Loose hamilton cycles in 3-uniform hypergraphs of high minimum degree. *Journal of Combinatorial Theory, Series B*, 96(6):767 – 821, 2006.

[29] B. Mohar. On the laplacian coefficients of acyclic graphs. *Linear Algebra and its Applications*, 422(2):736 – 741, 2007.

[30] B. Nagle, V. Rödl, and M. Schacht. An algorithmic hypergraph regularity lemma. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 1765–1773, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

[31] Turán P. On an extremal problem in graph theory. *Matematikai és Fizikai Lapok (in Hungarian)*, 48:436 – 452, 1941.

[32] V. Rödl, A. Ruciński, and E. Szemerédi. A dirac-type theorem for 3-uniform hypergraphs. 15, 01 2006.

[33] V. Rödl, A. Ruciński, and E. Szemerédi. An approximate dirac-type theorem for. 28:229–260, 03 2008.

[34] A. Satyanarayana, L. Schoppmann, and C.L. Suffel. A reliability-improving transformation with applications to network reliability. *Networks*, 22:209 – 216, 03 1992.