

RICE UNIVERSITY

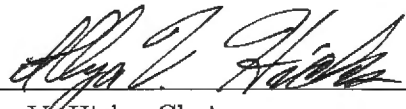
**Graph Coloring, Zero Forcing, and Related
Problems**

by

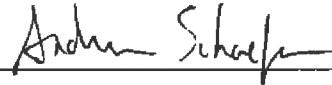
Boris Brimkov

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
Doctor of Philosophy

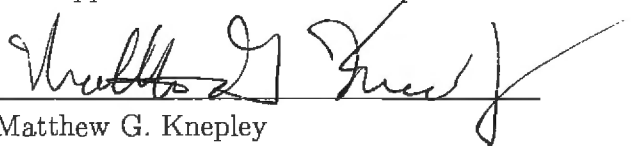
APPROVED, THESIS COMMITTEE:



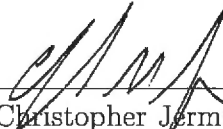
Illya V. Hicks, Chair
Professor of Computational and Applied
Mathematics



Andrew J. Schaefer
Noah Harding Professor of Computational
and Applied Mathematics



Matthew G. Knepley
Assistant Professor of Computational and
Applied Mathematics



Christopher Jermaine
Associate Professor of Computer Science

Houston, Texas

May, 2017

ABSTRACT

Graph Coloring, Zero Forcing, and Related Problems

by

Boris Brimkov

This thesis investigates several problems related to classical and dynamic coloring of graphs, and enumeration of graph attributes. In the first part of the thesis, I present new efficient methods to compute the chromatic and flow polynomials of specific families of graphs. The chromatic and flow polynomials count the number of ways to color and assign flow to the graph, and encode information relevant to various physical applications. The second part of the thesis focuses on zero forcing — a dynamic graph coloring process whereby at each discrete time step, a colored vertex with a single uncolored neighbor forces that neighbor to become colored. Zero forcing has applications in linear algebra, quantum control, and power network monitoring. A connected forcing set is a connected set of initially colored vertices which forces the entire graph to become colored; the connected forcing number is the cardinality of the smallest connected forcing set. I present a variety of structural results about connected forcing, such as the effects of vertex and edge operations on the connected forcing number, the relations between the connected forcing number and other graph parameters, and the computational complexity of connected forcing. I also give efficient algorithms for computing the connected forcing numbers of different families of graphs, and characterize the graphs with extremal connected forcing numbers. Finally, I investigate several enumeration problems associated with zero forcing, such

as the exponential growth of certain families of forcing sets, relations of families of forcing sets to matroids and greedoids, and polynomials which count the number of distinct forcing sets of a given size.

Acknowledgments

I would like to thank my parents for their love and encouragement; my advisor Dr. Illya Hicks for his wisdom, leadership, and enthusiasm which has inspired me to fulfill my research goals; the members of my committee Dr. Andrew Schaefer, Dr. Matthew Knepley, and Dr. Christopher Jermaine for their helpful remarks and for enhancing my background in computational and applied mathematics. I am grateful to all my colleagues in the CAAM Department for supporting me throughout the course of my graduate career and providing an enriching work environment.

Financial support for this research was provided by the National Science Foundation under Grant No. 1450681.

Contents

Abstract	ii
List of Illustrations	viii
1 Background and literature review	1
1.1 Graph polynomials	1
1.1.1 The chromatic polynomial	3
1.1.2 The flow polynomial	15
1.1.3 Other graph polynomials	28
1.2 Zero forcing and related problems	30
1.2.1 Zero forcing	31
1.2.2 Minimum rank	33
1.2.3 Power domination	36
1.2.4 Graph searching, quantum control, and zero forcing variants .	38
1.2.5 Connected zero forcing	41
2 Preliminaries	45
2.1 Graph definitions	45
2.2 Graph parameters and operations	47
2.3 Special types of graphs	49
3 Characterizations of chromatic and flow polynomials	51
3.1 Characterizations of chromatic polynomials	52
3.1.1 Generalized vertex join trees	53
3.1.2 Generalized vertex join cycles	61

3.1.3	Generalized vertex join cliques	64
3.2	Characterizations of flow polynomials	65
3.2.1	Outerplanar graphs	65
3.2.2	Generalized vertex join cycles	68
3.3	Computational results	69
3.3.1	Generalized vertex join trees	70
3.3.2	Generalized vertex join cycles	72
3.3.3	Generalized vertex join cliques	72
3.3.4	Discussion	74
4	Structural results and complexity of connected forcing	76
4.1	Basic properties and relation to other parameters	76
4.2	Vertex and edge operations	79
4.3	Bounds and mandatory vertices	85
4.4	Computational complexity	91
5	Characterizations of connected forcing numbers	94
5.1	Graphs with symmetry	94
5.2	Trees and unicycle graphs	100
5.3	Unicyclic graphs	104
5.4	Cactus and block graphs with no pendant paths	117
6	Graphs with extremal connected forcing numbers	121
6.1	Graphs with $Z_c(G) = 1$ and $Z_c(G) = n - 1$	122
6.2	Graphs with $Z_c(G) = 2$	123
6.3	Graphs with $Z_c(G) = n - 2$	129
7	Enumeration problems related to zero forcing	152
7.1	Exponential quantities associated with zero forcing	152

7.2	Connected forcing and matroids	156
7.3	Zero forcing polynomial	162
8	Conclusion	177
	Bibliography	181

Illustrations

1.1	A four-colored map	2
1.2	3-colorings of a graph	4
1.3	Chromatic polynomial of house graph	6
1.4	Appearance of chromatic polynomial	8
1.5	Nowhere-zero \mathbb{Z}_3 - and \mathbb{Z}_4 -flows of a graph	19
1.6	Flow polynomial of house graph	22
1.7	Zero forcing in a graph	31
2.1	Vertex join of a graph	49
2.2	Graph and its dual	50
3.1	Generalized vertex join of a graph	52
3.2	Forming a generalized vertex join tree	54
3.3	Removing bridges of a generalized vertex join tree	55
3.4	Forming levels in a generalized vertex join tree	56
3.5	Special subgraphs of a generalized vertex join tree	57
3.6	Generalized vertex join cycle C_S	62
3.7	$C_{S'}$, the underlying graph of C_S	62
3.8	Decomposing a generalized vertex join cycle	63
3.9	The dual of a generalized vertex join cycle	69
3.10	Computing $P(T_S)$ with Algorithm 1 and Mathematica	71
3.11	Computing $P(T_S)$ with Algorithm 1 for larger graphs	71

3.12	Computing $P(C_S)$ with Algorithm 3 and Mathematica	73
3.13	Computing $P(K_S)$ with Algorithm 4 and Mathematica	74
4.1	$Z_c(G)$ and $Z(G)$ can differ arbitrarily	77
4.2	Minimum connected forcing set containment	78
4.3	The connected forcing spread can be arbitrarily large	81
4.4	Contraction can change the connected forcing number arbitrarily . . .	82
4.5	Subdivision can make the connected forcing number grow arbitrarily .	84
4.6	Gadget for complexity of connected forcing	92
5.1	Sun graph and minimum connected forcing set	100
5.2	Uniclique graphs G for which $Z(G) = Z_c(G)$	104
5.3	A unicyclic graph and a minimum connected forcing set	115
5.4	Unicyclic graphs G for which $Z(G) = Z_c(G)$	117
6.1	Graphs G of two parallel paths with $Z_c(G) = 2$, case 1	125
6.2	Graphs G of two parallel paths with $Z_c(G) = 2$, case 2	126
6.3	Forbidden induced subgraphs for $Z(G) \geq n - 2$	129
6.4	Graphs G with $Z_c(G) = n - 2$, $\kappa(G) = 1$, case 1	132
6.5	Graphs G with $Z_c(G) = n - 2$, $\kappa(G) = 1$, case 2	134
6.6	Graphs G with $Z_c(G) = n - 2$, $\kappa(G) = 1$, case 3	136
6.7	Graphs G with $Z_c(G) = n - 2$, $\kappa(G) = 1$, case 4	139
6.8	Graphs G with $Z_c(G) = n - 2$, $\kappa(G) \geq 2$, case 1	141
6.9	Graphs G with $Z_c(G) = n - 2$, $\kappa(G) \geq 2$, case 2	149
7.1	A tree for which $z_c(T; 8)$ is computed	172
7.2	All connected forcing sets of size 8 of a tree	173
7.3	A graph with a unique minimum connected forcing set	174

Chapter 1

Background and literature review

This chapter surveys the literature and historical development of chromatic and flow polynomials, zero forcing, and related topics. See Chapter 2 for graph theoretic notions and operations used throughout this discussion.

1.1 Graph polynomials

Algebraic graph theory studies properties of graphs by algebraic means. This approach often leads to elegant proofs and at times reveals deep and unexpected connections between graph theory and algebra. In the last few decades, algebraic graph theory has developed rapidly, generating a substantial body of literature; the monographs of Biggs [1] and Godsil and Royle [2] are standard sources on the subject.

A central topic of algebraic graph theory is the study of polynomials associated with graphs. These polynomials contain important information about the structure and properties of graphs, and enable its extraction by algebraic methods. In particular, the values of graph polynomials at specific points, as well as their coefficients, roots, and derivatives, often have meaningful interpretations.

The study of graph polynomials was motivated by the Four Color Conjecture, which states that any map can be “face-colored” using four colors so that neighboring regions do not share the same color; see Figure 1.1 for an example. In 1912, Birkhoff [4] introduced a polynomial which counts the ways to face-color a planar

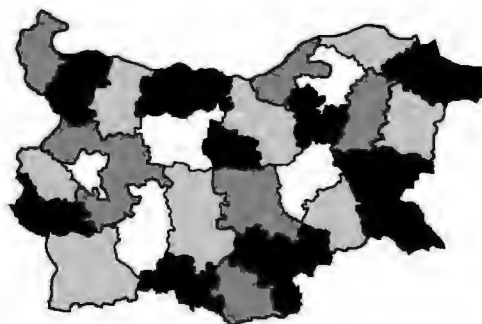


Figure 1.1 : A map colored with four colors (adapted from [3])

graph and attempted to prove the Four Color Conjecture by analyzing the roots of this polynomial. For planar graphs, the concept of face-coloring is equivalent to “vertex-coloring”, and in 1932, Whitney [5] generalized Birkhoff’s polynomial to count vertex-colorings of general graphs; this polynomial is known today as the chromatic polynomial. The Four Color Conjecture was proved true in 1976 by Appel and Haken [6] with the help of a computer, though an analytic proof in the vein of Birkhoff’s attempt is still being sought.

In 1954, Tutte [7] extended the idea of face-coloring to non-planar graphs by introducing group-valued flows and the associated flow polynomial; the chromatic and flow polynomials are closely related, and are essentially equivalent in planar graphs by graph duality. Tutte and Whitney further generalized the chromatic and flow polynomials into the two-variable Tutte polynomial [8], which includes as special cases several other graph polynomials such as the reliability and Jones polynomials. Since then, graph polynomials which are not special cases of the Tutte polynomial have also been introduced. For instance, Hoede and Li [9] introduced the clique and independence polynomials, and McClosky, Simms, and Hicks [10] generalized the

independence polynomial into the co- k -plex polynomial; see Section 1.1.3 for other examples. Studying these polynomials has been an active area of research: Chia's bibliography published in 1997 [11] counts 472 titles on chromatic polynomials alone.

The interest in graph polynomials is in the information they contain about the properties of graphs and networks, which can be easily obtained by algebraic techniques but is much harder to access through purely graph theoretic approaches. Graph polynomials also have connections to sciences such as statistical physics, knot theory, and theoretical computer science. The chromatic and flow polynomials remain two of the most well-studied single-variable graph polynomials, and are a subject of this thesis; several important results about them are discussed in the remainder of this section. Dong [12] and Zhang [13] are standard resources on chromatic and flow polynomials respectively, while Tutte [14] relates the two polynomials in a broader framework. These monographs provide a number of technical tools for the computation of chromatic and flow polynomials, some of which are included in this section and applied in Chapter 3.

1.1.1 The chromatic polynomial

A *vertex coloring* of G is an assignment of colors to the vertices of G so that no edge is incident to vertices of the same color. Many problems which involve vertex coloring (which shall be referred to simply as *coloring*) are concerned with the following two questions:

Q1. Can G be colored with t colors?

Q2. In how many ways can G be colored with t colors?

Clearly, the answer to $Q2$ contains the answer to $Q1$ and therefore $Q2$ is more general and typically more difficult to answer. In a sense, $Q1$ is equivalent to asking “What is the least number of colors needed to color G ?” because if G can be colored with t colors, it can be colored with $t + 1$ colors as well. Thus, if t^* is the least number of colors needed to color G , then $Q1$ can be answered in the affirmative for all $t \geq t^*$ and in the negative for all $t < t^*$. The parameter t^* (usually written $\chi(G)$) which answers $Q1$ is called the *chromatic number* of G . The rest of this section will analyze the *chromatic polynomial* of G , which counts the number of ways to color G with t colors and answers $Q2$.

Let $G = (V, E)$ be a graph with n vertices. Formally, a t -coloring of G is a function $c : V \rightarrow \{1, \dots, t\}$ such that $c(u) \neq c(v)$ for any $e = uv \in E$. Let $p(G; t)$ denote the number of t -colorings of G for each nonnegative integer t . Figure 1.2 shows all 18 ways to color the house graph H using 3 colors; thus $p(H; 3) = 18$.

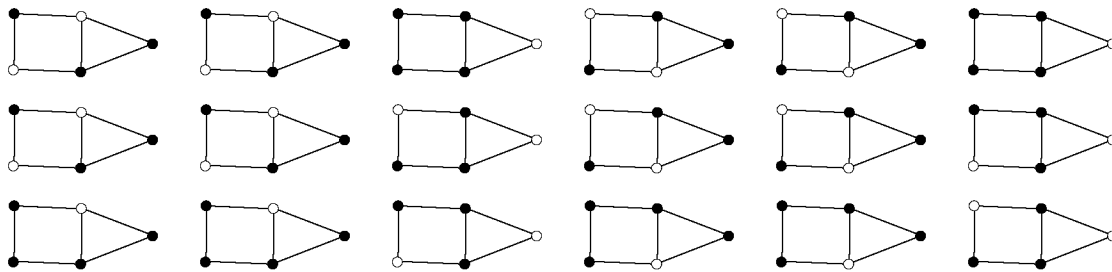


Figure 1.2 : All 3-colorings of the house graph H ; $p(H; 3) = 18$.

There are several equivalent definitions of the chromatic polynomial. Definition 1.1 gives immediate intuition into the nature and purpose of the chromatic polynomial; see (1.18) for an alternate definition.

Definition 1.1. The chromatic polynomial $P(G; t)$ is the unique interpolating polynomial of degree at most n of the integer points $\{(t, p(G; t))\}_{t=0}^n$, where $n = |V(G)|$.

Let us examine this definition closely. First, we can rightly claim that $P(G; t)$ is the unique interpolating polynomial of these $n + 1$ points due to the following well-known theorem (see [15] for a proof).

Theorem 1.1 (Unisolvence Theorem). *Let $(x_0, y_0), \dots, (x_n, y_n)$ be points in \mathbb{R}^2 such that $x_0 < \dots < x_n$. Then, there exists a unique polynomial P of degree at most n such that $P(x_i) = y_i$, $0 \leq i \leq n$.*

By definition, the chromatic polynomial counts the number of ways to color G with n or fewer colors. Figure 1.3 shows a graphical representation of the chromatic polynomial of the house graph H and the points it interpolates. Since H has 5 vertices, by definition $P(H; t)$ is guaranteed to interpolate $p(H; t)$ for $t = \{0, 1, 2, 3, 4, 5\}$; however, notice that $P(H; 6) = p(H; 6)$. In fact, at each nonnegative integer t , $P(H; t) = p(H; t)$, and this is true for the chromatic polynomial of any graph. This fact is stated in the following theorem, followed by a proof adapted from [16].

Theorem 1.2. *At each nonnegative integer t , $P(G; t) = p(G; t)$.*

Proof. If G is a graph with a loop, there can be no proper coloring of G with any number of colors since the loop will always be incident to vertices of the same color. Thus, $p(G; t) = 0$ for all integers $t \geq 0$, so $P(G; t) = 0$ and the claim is true.

Let G be a loopless graph and u, v be two vertices of G . The t -colorings of G can be split up into those in which u and v have different colors and those in which they have the same color. Adding the edge uv to G assures that u and v have different colors, and identifying u and v into a single vertex assures that they have the same color.

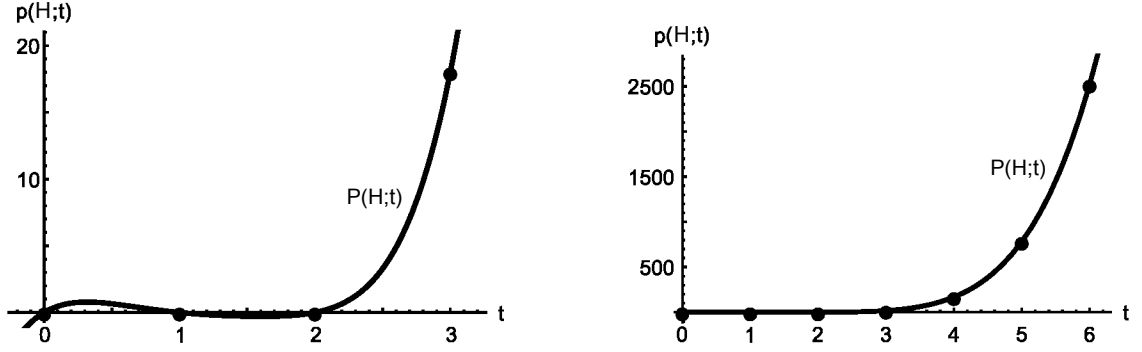


Figure 1.3 : The chromatic polynomial of the House Graph H and the points it interpolates evaluated for $0 \leq t \leq 3$ (left) and $0 \leq t \leq 6$ (right). Note that $P(H; 3) = 18$ as found in Figure 1.2.

Thus, $p(G; t) = p(G + e; t) + p(G/e; t)$; equivalently, $p(G; t) = p(G - e; t) - p(G/e; t)$ for each nonnegative integer t . Note also that if e is an edge of multiplicity greater than 1, $p(G; t) = p(G - e; t)$.

We will now show by induction on the number of edges that for any loopless graph G , there exists a degree n polynomial $P(G; t)$ such that $P(G; t) = p(G; t)$ at each nonnegative integer t . This polynomial must be the chromatic polynomial, since two polynomials of degree n which agree at $n + 1$ points must be identical by Theorem 1.1.

If G is a loopless graph with no edges, each vertex can be colored with any available color, so $p(G; t) = t^n$ and the degree n polynomial $P(G; t) = t^n$ satisfies the conditions of the theorem. Now, suppose G is a loopless graph with $m \geq 1$ edges and n vertices. If G has an edge e of multiplicity greater than 1, it is easy to see that $G - e$ is loopless and has n vertices and $m - 1$ edges. By induction, there exists a polynomial $P(G - e; t)$ of degree n equal to $p(G - e; t)$ for all nonnegative integers t . Thus, we define $P(G; t) = P(G - e; t)$, so that $P(G; t) = P(G - e; t) = p(G - e; t) = p(G; t)$, at

all nonnegative integers t , and $P(G; t)$ has degree n .

If every edge of G has multiplicity 1, let e be any edge of G . It is easy to see that $G - e$ is loopless and has n vertices and $m - 1$ edges, and that G/e is loopless and has $n - 1$ vertices and $m - 1$ edges. Hence, by induction, there exists a polynomial $P(G - e; t)$ of degree n equal to $p(G - e; t)$ for all nonnegative integers t , and a polynomial $P(G/e; t)$ of degree $n - 1$ equal to $p(G/e; t)$ for all nonnegative integers t . Thus, we define $P(G; t) = P(G - e; t) - P(G/e; t)$, so that $P(G; t) = P(G - e; t) - P(G/e; t) = p(G - e; t) - p(G/e; t) = p(G; t)$, at all nonnegative integers t , and $P(G; t)$ has degree n . This completes the induction. \square

The dependence of the chromatic polynomial on t is often implied in the context; if there is no scope for confusion, $P(G; t)$ can be abbreviated to $P(G)$. By convention, the graph with no vertices has chromatic polynomial equal to 1; this graph will be excluded from further considerations.

Properties

Trivially, G can be colored by assigning a different color to each vertex; thus, $\chi(G) \leq n$. Unless $t = \chi(G)$, it is possible to have unused colors in a t -coloring; if $t > n$, this becomes necessary. If $t_2 > t_1 \geq \chi(G)$, any t_1 -coloring is also a t_2 -coloring and $p(G; t_2) > p(G; t_1)$. Thus, the sequence $\{p(G; t)\}_{t=0}^{\infty}$ starts with $\chi(G)$ zeroes and then strictly increases. The behavior of the chromatic polynomial of a general graph is pictured in Figure 1.4.

The coefficients, derivatives, roots, and values of the chromatic polynomial at certain points contain various information about the graph and are widely studied. Below are several characteristics of the chromatic polynomial evaluated at specific points.

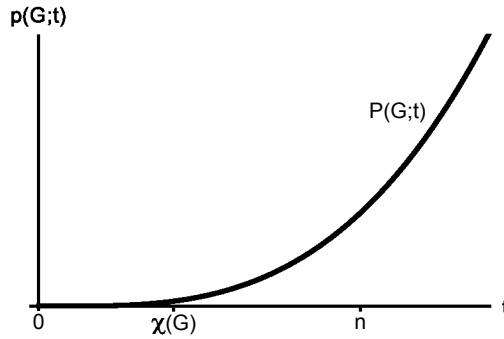


Figure 1.4 : General appearance of the chromatic polynomial in the first quadrant.

- $P(G; t)$ is the number of t -colorings of G for any nonnegative integer t .
- The chromatic number of G is the smallest positive integer t for which $P(G; t) > 0$. It can be determined by evaluating $P(G; t)$ at $t = 1, \dots, n$ (or faster, by a form of binary search).
- For any integers $t_2 > t_1 \geq \chi(G)$, $P(G; t_2) > P(G; t_1)$.
- Stanley [17] gives a combinatorial interpretation of the chromatic polynomial evaluated at negative integers in terms of orientations of G . In particular, $|P(G; -1)|$ is the number of acyclic orientations of G .

Let $P(G; t) = c_n t^n + c_{n-1} t^{n-1} + \dots + c_1 t + c_0$. The coefficients of the chromatic polynomial have the following properties:

- c_0, \dots, c_n are integers.
- $c_n = 1$.
- $c_{n-1} = -m$ where m is the number of edges of G .
- If $m \neq 0$, $\sum_{i=1}^n c_i = 0$; if $m = 0$, $P(G; t) = t^n$ and $\sum_{i=1}^n c_i = 1$.

- $c_0, \dots, c_{k-1} = 0$ and $c_k, \dots, c_n \neq 0$ where k is the number of components of G .
- $c_i \geq 0$ if $i = n \bmod 2$, $c_i \leq 0$ if $i \neq n \bmod 2$, i.e., the coefficients alternate signs.
- c_1, \dots, c_{n-1} are #P-hard to compute, even for bipartite and planar graphs [18].
- The sequence $\{|c_i|\}_{i=0}^n$ is log-concave, i.e., $|c_i|^2 \geq |c_{i-1}||c_{i+1}|$ for $0 < i < n$ [19].
This also implies it is unimodal, i.e., for some c_i , $|c_1| \leq \dots \leq |c_i| \geq \dots \geq |c_n|$.
The log-concavity of the chromatic polynomial was a long-standing conjecture of Read [20]; it was proven true for outerplanar graphs [21] and some other special families, before being settled recently for all graphs by Huh [19].

The derivative of the chromatic polynomial has interesting properties as well, in particular when evaluated at 1. The quantity $\theta(G) = (-1)^n \frac{d}{dt} P(G; t) \big|_{t=1}$ is called the *chromatic invariant* of G and has been widely studied. Below are two results due to Crapo [22] and Brylawski [23], respectively; in addition, see [24, 25] for combinatorial interpretations of $\theta(G)$.

- $\theta(G) \neq 0$ if and only if G is biconnected.
- $\theta(G) = 1$ if and only if G is series-parallel. Series-parallel (SP) graphs are used to model electrical circuits. It is useful to know that a graph is SP because many NP-hard problems can be solved in linear time over SP graphs (cf. [26]).

Finally, the roots of $P(G; t)$ — called the *chromatic roots* of G — contain significant information about the structure of G . The set of chromatic roots of all graphs (or of special families of graphs) is interesting in its own right; recall that Birkhoff's motivation for introducing the chromatic polynomial was to investigate gaps in the set of chromatic roots of planar graphs in order to prove the Four Color Theorem.

- The number of biconnected components of G is the multiplicity of the root ‘1’ of $P(G; t)$ [27].
- If $P(G; t)$ has a noninteger root less than or equal to $h \approx 1.29559$, then G has no Hamiltonian path [28]. Thus, if the chromatic polynomial of a graph can be found efficiently, its roots can be approximated to an appropriate precision to check whether the Traveling Salesman Problem has no solution on the graph.
- Let R be the set of all chromatic roots (of all graphs). R is dense in $[32/27, \infty)$ [29] and dense in \mathbb{C} [30]. Moreover, $R \cap (-\infty, 32/27) = \{0, 1\}$, i.e., there are no real chromatic roots less than $32/27$ other than 0 and 1 [31].
- Not every complex number and real number in $[32/27, \infty)$ is a chromatic root; for example, $\phi + 1 = \frac{\sqrt{5}+3}{2}$ is not a root of any chromatic polynomial [32].
- 5-Color Theorem [33]: Planar graphs have no real chromatic roots in $[5, \infty)$.
- 4-Color Theorem [6]: 4 is not a chromatic root of planar graphs. Appel and Haken proved this by eliminating a long list of minimum counterexamples using a computer; it is still an open problem to prove the 4-Color Theorem by analyzing chromatic roots.
 - Birkhoff-Lewis Conjecture [34]: Planar graphs have no real roots in $[4, \infty)$.

Naturally, the chromatic polynomial is often used in graph coloring problems, which are widely applicable in scheduling, resource allocation, and pattern matching. Read [20] gives two specific applications of the chromatic polynomial to the construction of timetables and the allocation of channels to television stations; see Kubale’s monograph [35] for more graph coloring problems.

The chromatic polynomial is also used in statistical physics to model the behavior of ferromagnets and crystals; in particular, it is the zero-temperature limit of the anti-ferromagnetic Potts model. The limit points of the roots of chromatic polynomials indicate where physical phase transitions may occur [36]. Finally, the chromatic polynomial is related to the Stirling numbers [37, 38], the Beraha numbers [39], and the golden ratio [40], and thus finds applications in a variety of analytic and combinatorics problems.

Computation

Computing the chromatic polynomial of a graph from its definition is highly impractical. Fortunately, there are several well-known formulas which aid in this computation by reducing the chromatic polynomial of a graph into that of smaller graphs. The most notable of these is the deletion-contraction formula, which is discussed next.

Let $G = (V, E)$ be a graph and u and v be two vertices of G . The t -colorings of G can be split up into those in which u and v have different colors and those in which they have the same color. Adding the edge uv to G assures that u and v have different colors, and identifying u and v into a single vertex assures that they have the same color. Thus, for every coloring of G in which u and v have different colors, there is exactly one coloring of $G + uv$ and for every coloring of G in which u and v have the same color, there is exactly one coloring of G/uv . This observation yields the *addition-contraction* formula:

$$P(G) = P(G + e) + P(G/e) \text{ for any } e = uv, \text{ where } u, v \in V. \quad (1.1)$$

Note that this formula is only valid when the edge e does not already exist in the graph. Alternately, we can set $H = G - e$ for some edge e of G ; then, $P(G - e) =$

$P(H) = P(H + e) + P(H/e) = P(G) + P((G - e)/e)$, but $(G - e)/e = G/e$, so $P(G) = P(G - e) - P(G/e)$. This yields the *deletion-contraction* formula:

$$P(G) = P(G - e) - P(G/e) \text{ for any } e = uv, \text{ where } u, v \in V. \quad (1.2)$$

Equations 1.1 and 1.2 can be used recursively to compute the chromatic polynomial of any graph G . In particular, each application of (1.2) reduces G into two graphs each with one fewer edge and, after enough iterations, into graphs with no edges; thus, $P(G)$ can be expressed as a linear combination of the chromatic polynomials of empty graphs. Similarly, (1.1) can be used to express $P(G)$ as a linear combination of the chromatic polynomials of complete graphs. The chromatic polynomials of empty graphs and complete graphs can be computed directly using combinatorial arguments as shown in Examples 1.1 and 1.2.

The *Combinatorica* package of the computer algebra system Mathematica has a `ChromaticPolynomial` function which uses the addition-contraction formula to compute the chromatic polynomials of dense graphs, and the deletion-contraction formula for sparse graphs [41]. Version 10 of Mathematica has a more efficient implementation of the deletion-contraction algorithm built into the Wolfram System.

It can be shown (cf. [42]) that an algorithm based on the recurrences (1.1) and (1.2) has a worst-case run time of $O(\phi^{n+m})$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. Some improvements on this algorithm have been made by Bjorklund [43]. With *a priori* information about the structure of the graph, the run time of this algorithm can be improved significantly; in particular, the order of the edges being contracted and deleted can be chosen strategically to obtain large subgraphs whose chromatic polynomials are known. Moreover, there are polynomial time algorithms for computing the chromatic polynomials of graphs with bounded clique-width and tree-width.

In particular, Makowsky et al. [44] show that $P(G)$ can be computed in $O(n^{f(w)})$ time, where w is the clique-width of G and $f(w) \leq 3 \cdot 2^{w+2}$. However, even for $w = 2$, this yields a worst-case run time of $O(n^{48})$. Similarly, Andrzejak's algorithm [45] for computing the Tutte polynomials of graphs of tree-width t has a worse-case run time of $O(n^{109.6})$ when $t = 2$. Thus, these algorithms are principally of theoretical value.

In addition to formulas (1.1) and (1.2), there are several other reduction formulas for chromatic polynomials which are outlined next; see Tutte [14] for detailed proofs. If G has two subgraphs whose intersection is a clique, then the chromatic polynomials of those subgraphs can be combined to compute the chromatic polynomial of G in the following way:

$$\text{If } G = G_1 \cup G_2 \text{ and } G_1 \cap G_2 = K_r, \text{ then } P(G) = \frac{P(G_1)P(G_2)}{P(K_r)}. \quad (1.3)$$

Note that cut vertices are cliques of size 1, so this formula can be used to separate a graph into biconnected components to find its chromatic polynomial. Furthermore, disjoint components of a graph can be colored independently; thus, to compute the chromatic polynomial of a disconnected graph, it suffices to compute the chromatic polynomials of each component separately:

$$\text{If } G = G_1 \cup G_2 \text{ and } G_1 \cap G_2 = \emptyset, \text{ then } P(G) = P(G_1)P(G_2). \quad (1.4)$$

If G has a vertex v which is connected to every other vertex in G , then

$$P(G; t) = tP(G - v; t - 1). \quad (1.5)$$

Equivalently, this formula can be used to compute the chromatic polynomial of a vertex join G_V of G as shown below:

$$P(G_V; t) = tP(G; t - 1). \quad (1.6)$$

Finally, multiple edges between vertices u and v have no more effect on the coloring of G than a single edge between u and v ; thus,

$$\text{If } e \in E \text{ is a multiple edge, then } P(G) = P(G - e). \quad (1.7)$$

This implies that the chromatic polynomial of a multigraph is equal to the chromatic polynomial of its underlying simple graph and that the chromatic polynomial is invariant under amalgamorphism. However, it is still sometimes useful to consider the chromatic polynomials of graphs with multiple edges because the duals of these graphs form a more general family. This matter is discussed further in Remark 3.1.

Using the decomposition techniques outlined thus far along with simple combinatorial arguments, closed formulas for the chromatic polynomials of some specific graphs can be derived. Some of these closed formulas will be used in Chapter 3 to compute the chromatic polynomials of more complex graphs. For more detailed proofs and other examples, see [12].

Example 1.1. Let $(K_n)^c$ be the empty graph on n vertices. Given t available colors, each vertex in $(K_n)^c$ can be colored independently in t ways. Thus,

$$P((K_n)^c; t) = t^n. \quad (1.8)$$

Example 1.2. Let K_n be the complete graph on n vertices. Since all vertices in K_n are mutually adjacent, after fixing the color of one arbitrary vertex, each successive vertex can be colored in one fewer ways. With t available colors, the first vertex can be colored in t ways, the second in $t - 1$ ways, etc., so

$$P(K_n; t) = t(t - 1) \cdots (t - (n - 1)) = \prod_{i=0}^{n-1} (t - i). \quad (1.9)$$

Example 1.3. Let G be an arbitrary tree on n vertices. A tree on one vertex can be colored in t ways, and adding a leaf vertex to a tree increases the number of colorings by a factor of $t - 1$, since the added vertex cannot have the same color as its neighbor. Thus,

$$P(G; t) = t(t - 1)^{n-1}. \quad (1.10)$$

Example 1.4. Let C_n be the cycle on n vertices. Applying the deletion-contraction formula to an arbitrary edge yields a tree and a cycle on $n - 1$ vertices; using this formula recursively yields

$$P(C_n; t) = (t - 1)^n + (-1)^n(t - 1). \quad (1.11)$$

Example 1.5. Let W_n be the wheel with n spokes. A wheel is a vertex join of a cycle on n vertices; applying (1.6) to (1.11) yields

$$P(W_n; t) = t((t - 2)^{n-1} + (-1)^{n-1}(t - 2)). \quad (1.12)$$

1.1.2 The flow polynomial

A plane graph G has a well-defined dual whose vertices correspond to faces of G , so in plane graphs, the concept of face-coloring is essentially equivalent to vertex-coloring. However, face-coloring cannot be defined on general graphs in the same sense as on planar graphs, since non-planar graphs do not have well-defined faces (in a planar embedding). To this end, Tutte introduced the theory of group- and integer-valued flows as a way to extend face-coloring from planar graphs to general graphs. A group-valued (respectively, integer-valued) flow on an orientation of G is an assignment of values to the edges of G from an Abelian group (respectively, from a set of integers) so that flow is conserved at each vertex of G .

The theory of group- and integer-valued flows is connected to some of the deepest and most challenging notions in graph theory such as the cycle-double cover conjecture [46] and the Four Color Theorem. Many problems which involve flows are concerned with whether a graph admits a certain flow, and if so — in how many different ways. Just as the chromatic polynomial counts the number of graph colorings, there is a flow polynomial which counts the number of group-valued flows on a given graph. Defining and studying this polynomial will be the subject of this section. To this end, recall first the definition of an Abelian algebraic group.

Definition 1.2. An *Abelian group* $(A, +)$ is an ordered pair consisting of a set A and a binary operation ‘+’ which together satisfy the following conditions:

Closure: $a, b \in A \implies a + b \in A$

Associativity: $a, b, c \in A \implies a + (b + c) = (a + b) + c$

Identity element: $\exists 0 \in A$ such that $0 + a = a + 0 = a \quad \forall a \in A$

Inverse elements: $\forall a \in A \exists (-a) \in A$ such that $a + (-a) = (-a) + a = 0$

Commutativity: $a, b \in A \implies a + b = b + a$.

When there is no scope for confusion, the group $(A, +)$ is abbreviated as A ; the *cardinality* of group $(A, +)$ is equal to $|A|$, the number of elements in A . A *finite Abelian group* is an Abelian group of finite cardinality. A simple example of a finite Abelian group is the cyclic group $(\mathbb{Z}_t, +)$ where $\mathbb{Z}_t = \{0, 1, \dots, t - 1\}$ and ‘+’ is addition modulo t . In fact, \mathbb{Z}_t is essentially the only finite Abelian group that will be required in the context of this chapter. With this in mind, a group-valued flow can be defined as follows.

Definition 1.3. Let A be a finite Abelian group and $G = (V, E)$ be a graph with a fixed orientation. An A -flow on this orientation of G is a function $\phi : E \rightarrow A$ such that for each $v \in V$, $\sum_{h(e)=v} \phi(e) = \sum_{t(e)=v} \phi(e)$. An A -flow ϕ is *nowhere-zero* if $\phi(e) \neq 0$ for all $e \in E$.

The condition $\sum_{h(e)=v} \phi(e) = \sum_{t(e)=v} \phi(e)$ in Definition 1.3 is sometimes called *Kirchhoff's Law* (commonly applied as the principle of conservation of energy in electrical circuits) and means that the total flow entering each vertex is equal to the total flow leaving each vertex. Nowhere-zero A -flows are also called *group-valued* flows and *modular* flows. Taking $A = \mathbb{Z}$ and $|\phi(e)| < t$ for each $e \in E$ in the definition above yields another type of flow called a nowhere-zero t -flow. This is stated more formally as follows.

Definition 1.4. Let $G = (V, E)$ be a graph with a fixed orientation. A *nowhere-zero t -flow* on this orientation of G is a function $\phi : E \rightarrow \{-(t-1), \dots, (t-1)\} \setminus \{0\}$, such that for each $v \in V$, $\sum_{h(e)=v} \phi(e) = \sum_{t(e)=v} \phi(e)$.

Nowhere-zero A -flows and nowhere-zero t -flows (also called *integer-valued* flows) are closely related but not identical; the likeness in meaning and nomenclature of these two concepts warrants caution. The similarities and differences between group-valued and integer-valued flows are now discussed.

First, it is easy to see that the orientation of G in the definitions above is irrelevant. If ϕ is a nowhere-zero A -flow on a certain orientation of G and a new orientation of G is obtained by reversing the direction of some edge e_0 , then

$$\hat{\phi}(e) = \begin{cases} \phi(e) & \text{if } e \neq e_0 \\ -\phi(e) & \text{if } e = e_0 \end{cases} \quad (1.13)$$

is a nowhere-zero A -flow on the new orientation of G (where $-\phi(e)$ is the inverse element of $\phi(e)$). Thus, if some orientation of G has a nowhere-zero A -flow, every orientation of G has a nowhere-zero A -flow; moreover, the number of nowhere-zero A -flows is the same in all orientations of G . An analogous result holds for nowhere-zero t -flows.

In addition, Tutte [7] showed that when A is finite, the existence and number of nowhere-zero A -flows does not depend on the algebraic structure of A but only on its cardinality. Thus, without loss of generality, the group A in a nowhere-zero A -flow with $|A| = t$ can be taken to be \mathbb{Z}_t . Tutte also showed that G has a nowhere-zero t -flow if and only if it has a nowhere-zero \mathbb{Z}_t -flow. However, the number of nowhere-zero t -flows on G is not necessarily the same as the number of nowhere-zero \mathbb{Z}_t -flows. Finally, a graph with a bridge b does not admit a nowhere-zero t -flow or a nowhere-zero \mathbb{Z}_t -flow, since a non-zero flow on b would create a non-zero total outflow from some component of $G - b$ (see Lemma 1.1 for more details). These results are summarized in the following theorem.

Theorem 1.3 (Tutte [7, 14]). *Let G be a bridgeless graph with a fixed orientation and A be an Abelian group with $|A| = t$. Then, the following statements are equivalent:*

1. *G has a nowhere-zero t -flow*
2. *G has a nowhere-zero \mathbb{Z}_t -flow*
3. *G has a nowhere zero A -flow*
4. *Every orientation of G has a nowhere-zero t -flow*
5. *Every orientation of G has a nowhere-zero \mathbb{Z}_t -flow*
6. *Every orientation of G has a nowhere-zero A -flow.*

Let $G = (V, E)$ be a graph with n vertices, m edges, and k components and let $f(G; t)$ denote the number of nowhere-zero \mathbb{Z}_t -flows on G for each positive integer t . Figure 1.5 shows the two nowhere-zero \mathbb{Z}_3 -flows and the six nowhere-zero \mathbb{Z}_4 -flows on an orientation of the house graph H ; thus $f(H; 3) = 2$ and $f(H; 4) = 6$.

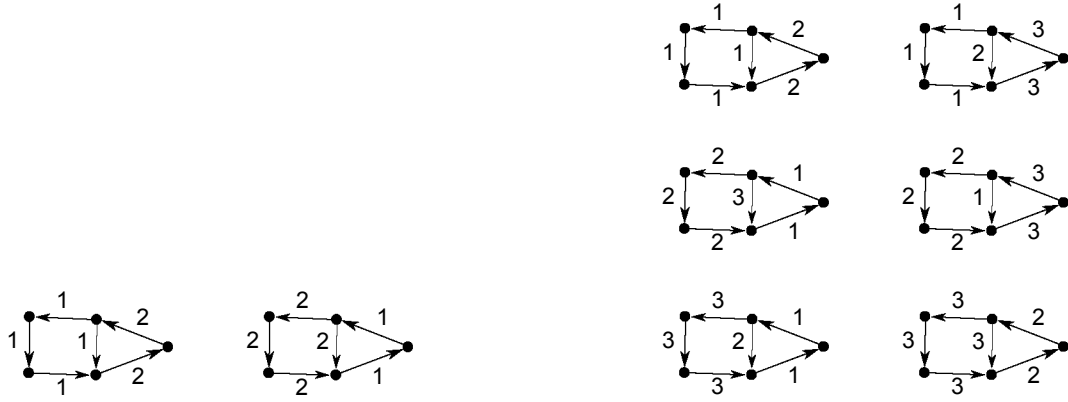


Figure 1.5 : *Left:* All nowhere-zero \mathbb{Z}_3 -flows on an orientation of the house graph H ; $f(H; 3) = 2$. *Right:* All nowhere-zero \mathbb{Z}_4 -flows on H ; $f(H; 4) = 6$.

The following lemma of Tutte [14] allows nowhere-zero \mathbb{Z}_t -flows to be counted recursively; it will be useful in defining and computing the flow polynomial.

Lemma 1.1. *Let G be a graph and e be an edge of G . Then, for each positive integer t ,*

$$f(G; t) = \begin{cases} f(G/e; t) - f(G - e; t) & \text{if } e \text{ is not a loop} \\ (t - 1)f(G - e; t) & \text{if } e \text{ is a loop} \\ (t - 1) & \text{if } G = C_1 \\ 0 & \text{if } G \text{ has a bridge} \end{cases} \quad (1.14)$$

Proof. If e is a loop, then $f(G; t) = (t - 1)f(G - e; t)$, since there are $f(G - e; t)$ nowhere-zero \mathbb{Z}_t -flows on $G - e$, and any of the $t - 1$ nonzero members of \mathbb{Z}_t can be assigned to e to produce a nowhere-zero flow on G .

Next, suppose e is not a loop. Let f_1 be the set of \mathbb{Z}_t -flows which are nowhere-zero on $G - e$ and in which e may have 0 flow. For every \mathbb{Z}_t -flow in f_1 , there is exactly one nowhere-zero \mathbb{Z}_t -flow on G/e . Let f_2 be the set of \mathbb{Z}_t -flows which are nowhere-zero on $G - e$ and in which e does have 0 flow. For every \mathbb{Z}_t -flow in f_2 , there is exactly one nowhere-zero \mathbb{Z}_t -flow on $G - e$. The nowhere-zero \mathbb{Z}_t -flows on G can be obtained by subtracting f_2 from f_1 . Thus, $f(G, t) = f(G/e, t) - f(G - e, t)$ for any non-loop edge e .

Now consider the graph C_1 consisting of one loop; $f(C_1; t) = t - 1$ since each of the $t - 1$ nonzero members of \mathbb{Z}_t can be assigned to the loop to produce a non-zero flow.

Finally, suppose G has a bridge b , let B be the component of G containing b , and let B_1 and B_2 be the two disconnected subgraphs of B obtained by deleting b . Without loss of generality, suppose b is directed from B_1 to B_2 in some orientation of G . Next, let ϕ be a nowhere-zero \mathbb{Z}_t -flow and for any $v \in V$ and $S \subset V$, define:

$$\begin{aligned} \phi^+(v) &= \sum_{e:t(e)=v, h(e) \neq v} \phi(e) && \text{(total flow into } v) \\ \phi^-(v) &= \sum_{e:h(e)=v, t(e) \neq v} \phi(e) && \text{(total flow out of } v) \\ \phi^+(S) &= \sum_{e:t(e) \in S, h(e) \notin S} \phi(e) && \text{(total flow into } S) \\ \phi^-(S) &= \sum_{e:h(e) \in S, t(e) \notin S} \phi(e) && \text{(total flow out of } S). \end{aligned}$$

Since ϕ must obey conservation of flow, $\phi^+(v) - \phi^-(v) = 0$ for all $v \in B_1$. Also, $\phi^+(B_1) \neq 0$ since b is the only edge satisfying $\{e : t(e) \in B_1, h(e) \notin B_1\}$, and $\phi^-(B_1) = 0$ since there are no edges satisfying $\{e : h(e) \in B_1, t(e) \notin B_1\}$. Then, $0 = \sum_{v \in B_1} (\phi^+(v) - \phi^-(v)) = \phi^+(B_1) - \phi^-(B_1) \neq 0$, which is a contradiction. Thus,

if G has a bridge, it cannot admit a nowhere-zero \mathbb{Z}_t -flow, so $f(G; t) = 0$ for all positive integers t .

□

With this in mind, the flow polynomial can be defined as the polynomial that counts nowhere-zero \mathbb{Z}_t -flows on G . There are several equivalent definitions of the flow polynomial; the one given below mirrors the definition of the chromatic polynomial and gives intuition into the nature and purpose of the flow polynomial. See (1.19) for an alternate definition.

Definition 1.5. The flow polynomial $F(G; t)$ is the unique interpolating polynomial of degree at most $m - n + k$ of the integer points $\{(t, f(G; t))\}_{t=1}^{m-n+k+1}$, where m , n , and k are the number of edges, vertices, and components in G , respectively.

Figure 1.6 shows a graphical representation of the flow polynomial of the house graph H and the points it interpolates. Since H has 6 edges, 5 vertices, and 1 connected component, by definition $F(H; t)$ is guaranteed to interpolate $f(H; t)$ for $t = \{1, 2, 3\}$; however, notice that $F(H; 4) = f(H; 4)$. In fact, at each positive integer t , $F(H; t) = f(H; t)$, and this is true for the flow polynomial of any graph. This fact is stated in the following theorem.

Theorem 1.4. *At each positive integer t , $F(G; t) = f(G; t)$.*

Proof. By Lemma 1.1, if G has a bridge, $f(G; t) = 0$ for all integers $t > 0$, so $F(G; t) = 0$ and the claim is true. We will now show by induction on the number of edges that for any bridgeless graph G , there exists a degree $m - n + k$ polynomial $F(G; t)$ such that $F(G; t) = f(G; t)$ at each positive integer t . This polynomial must be the flow polynomial, since two polynomials of degree $m - n + k$ which agree at $m - n + k + 1$ points must be identical by Theorem 1.1.

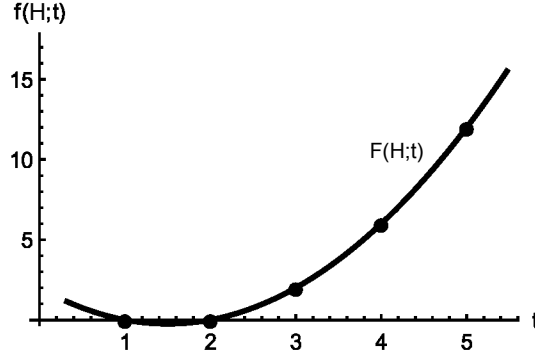


Figure 1.6 : The flow polynomial of the House Graph H and the points it interpolates evaluated for 1 through 5. Note that $F(H;3) = 2$ and $F(H;4) = 6$ as found in Figure 1.5.

If G is a bridgeless graph with one edge, that edge must be a loop, so $f(G;t) = t - 1$ by Lemma 1.1. Moreover, $m - n + k = 1$ since $n = k$ and $m = 1$, so the degree 1 polynomial $F(G;t) = t - 1$ satisfies the conditions of the theorem.

Now, suppose G is a bridgeless graph with $m > 1$ edges, n vertices, and k components, and let e be an edge of G . If e is not a loop, it is easy to see that G/e is bridgeless and has k components, $n - 1$ vertices and $m - 1$ edges. Thus, by induction, there exists a polynomial $F(G/e;t)$ of degree $m - n + k$ equal to $f(G/e;t)$ for all positive integers t . Similarly, $G - e$ has k components, n vertices, and $m - 1$ edges. If $G - e$ is bridgeless, by induction there exists a polynomial $F(G - e;t)$ of degree $m - n + k - 1$ equal to $f(G - e;t)$ for all positive integers t ; if $G - e$ has a bridge, $F(G - e;t) = 0$. Thus, we define $F(G;t) = F(G/e;t) - F(G - e;t)$, so that $F(G;t) = F(G/e;t) - F(G - e;t) = f(G/e;t) - f(G - e;t) = f(G;t)$, at all positive integers t , and $F(G;t)$ has degree $m - n + k$.

If e is a loop, it is easy to see that G/e is bridgeless and has k components, n vertices and $m - 1$ edges. Thus, by induction, there exists a polynomial $F(G/e;t)$

of degree $m - n + k - 1$ equal to $f(G/e; t)$ for all positive integers t . By Lemma 1.1 $f(G; t) = (t - 1)f(G - e; t)$; thus, we define $F(G; t) = (t - 1)F(G - e; t)$, so that $F(G; t) = (t - 1)F(G - e; t) = (t - 1)f(G - e; t) = f(G; t)$, and $F(G; t)$ has degree $m - n + k$. This completes the induction. \square

Remark 1.1. It should be noted that while the chromatic polynomial is often introduced rigorously in textbooks and papers, such introductions to the flow polynomial are somewhat rare in the literature. The proofs of Lemma 1.1 and Theorem 1.4 included here are modeled after the discussions and proofs in [47, 13, 14] and attempt to provide (perhaps for the first time) a unified and rigorous introduction to the flow polynomial.

Note that the number of nowhere-zero t -flows is generally not equal to $F(G; t)$; however, $F(G; t) > 0$ if and only if G admits a nowhere-zero t -flow. The dependence of the flow polynomial on t is often implied in the context; if there is no scope for confusion, $F(G; t)$ can be abbreviated to $F(G)$. By convention, the graph with zero edges has flow polynomial equal to 1; this graph will be excluded from further considerations in this section.

Just as the chromatic number $\chi(G)$ is the smallest number of colors needed to color G , the *flow number* of G , written $\psi(G)$, is the smallest positive t for which G has a nowhere-zero t -flow (and therefore a nowhere-zero \mathbb{Z}_t -flow). While the chromatic number of a graph can be arbitrarily high (for example, $\chi(K_n) = n$), the same is not true for the flow number – in fact, Seymour [48] showed that the flow number of any graph is at most 6. Nevertheless, for any t , deciding whether G has a nowhere-zero t -flow is an NP-hard problem. Indeed, even for planar graphs and $t = 3$, this problem is NP-complete [49].

Properties

Knowing the flow polynomial of a graph allows the flow number to be determined in linear time. In addition, the coefficients, roots, and values of the flow polynomial at certain points contain various information about the graph. Below are several characteristics of the flow polynomial of a bridgeless graph G with n vertices, evaluated at specific points.

- $F(G; t)$ is the number of nowhere-zero \mathbb{Z}_t -flows on G for any positive integer t .
- In general, for a positive integer t , $F(G; t)$ is not the number of nowhere-zero t -flows on G ; however, G admits a nowhere-zero t -flow if and only if $F(G; t) > 0$. Recently, Kochol [50] showed that there is a polynomial $F_{\mathbb{Z}}(G; t) \neq F(G; t)$ which counts the number of nowhere-zero t -flows in G , and that F and $F_{\mathbb{Z}}$ can be used to estimate one another.
- 6-flow Theorem: $F(G; 6) > 0$, i.e., every bridgeless graph admits a nowhere-zero 6-flow. This result is due to Seymour [48] and is an improvement over the earlier result of Jaeger and Kilpatrick who showed that every bridgeless graph has a nowhere-zero 8-flow [51, 52].
- 5-flow Conjecture [7]: $F(G; 5) > 0$, i.e., every bridgeless graph has a nowhere-zero 5-flow. This conjecture cannot be strengthened to “ $F(G; 4) > 0$ ”, since, e.g., the Petersen graph does not admit a nowhere-zero 4-flow.
- The flow number of G is the smallest positive integer t for which $F(G; t) > 0$. In view of the 6-flow Theorem, $\psi(G)$ can be determined by evaluating $F(G; t)$ at $t = 1, \dots, 5$.
- For any integers $t_2 \geq t_1 \geq \psi(G)$, $F(G; t_2) \geq F(G; t_1)$.

- Stanley and Noy [17, 53] give a combinatorial interpretation of the flow polynomial evaluated at negative integers in terms of orientations of G . In particular, $|F(G; -1)|$ is the number of totally cyclic orientations of G .
- $F(G; 1) = 0$, i.e., no graph admits a nowhere-zero 1-flow.
- $F(G; 2) > 0$ if and only if G is Eulerian.
- $F(G; t) > 0$ for all real $t > 2 \log_2 n$ [54]. This result is particularly useful for graphs with few vertices but many double edges and loops.

Let $F(G; t) = f_\nu t^\nu + \dots + f_1 t + f_0$. The coefficients of the flow polynomial have the following properties:

- f_0, \dots, f_ν are integers.
- $f_\nu = 1$.
- Computing f_0, \dots, f_ν is #P-hard, even for bipartite planar graphs [18].
- Dong and Koh [55] have shown that for $0 \leq i \leq \nu$, $|f_i|$ is bounded above by the coefficient of t^i in the expansion of a fixed polynomial of degree ν .

The degree ν of the flow polynomial has several interesting properties as well. Let G be a graph with m edges, n vertices, and k components; then,

- The degree ν of $F(G; t)$ is equal to $m - n + k$. This quantity is called the *cyclomatic number* of G , written $\nu(G)$.
- The cyclomatic number is equal to dimension of the *cycle space* of G — the set of all even subgraphs of G . There are many other connections between

integer flows and the cycle space of G , especially involving cycle covers; Zhang's monograph [13] is dedicated to this subject.

- A *control flow graph* is a directed graph representation of the different decision paths that can be taken in all possible executions of a computer program (cf. [56]). McCabe [57] introduced the cyclomatic complexity number of a program as a measure of a program's complexity; this number is the cyclomatic number of the corresponding control flow graph. It is interpreted as the amount of decision logic in a program and a high cyclomatic complexity number correlates with a high error rate of the program.

Finally, the flow polynomial contains information about the edge-connectivity of the graph. Below are some results about the existence of flows under given conditions. Recall that a graph has a nowhere-zero t -flow if and only if it has a nowhere-zero \mathbb{Z}_t -flow if and only if $F(G; t) > 0$; thus, the following results can also be stated in terms of the flow polynomial; for example, the first item below can be interpreted as "If $F(G; 3) = 0$, then G is not 6-edge-connected".

- Every 6-edge-connected graph has a nowhere-zero 3-flow [58]. This was an improvement over Thomassen's result that every 8-edge-connected graph has a nowhere-zero 3-flow [59]. Both of these important results are very recent, and have encouraged further research in the field.
- 4-flow Theorem [60]: Every 4-edge-connected graph has a nowhere-zero 4-flow.
- 3-flow Conjecture [61]: Every 4-edge-connected graph has a nowhere-zero 3-flow. Some progress on this conjecture has recently been made in [62, 63, 64].
- A 3-regular graph admits a nowhere-zero 3-flow if and only if it is bipartite [60].

In addition to its algebraic and combinatorial properties, the flow polynomial also has applications in crystallography and statistical mechanics as it is related to models of ice and crystal lattices (see [65]). In ice, each oxygen atom is connected by hydrogen bonds to four other oxygen atoms; each hydrogen bond contains a hydrogen atom which is closer to one of the two oxygen atoms it connects [66]. This structure can be represented by a directed graph, where the oxygen atoms are the vertices and the hydrogen bonds are directed edges pointing to the closer oxygen atom. The flow polynomial of such a graph can be used to count the number of permissible atomic configurations which conform to physical restrictions. In turn, this can be used to model the physical properties of ice and several other crystals, including potassium dihydrogen phosphate [67].

Computation

Just as the chromatic polynomial can be computed for general graphs using the deletion-contraction and addition-contraction formulas, so too can the flow polynomial be computed using the equations in Lemma 1.1 recursively; an algorithm based on these recursions is featured in Version 10 of Mathematica. The computational analysis of such an algorithm is analogous to the one given in the previous section for the chromatic polynomial.

Furthermore, if a graph G has $k > 1$ components, the following identity allows the flow polynomial of each component to be found separately:

$$\text{If } G = G_1 \cup G_2 \text{ and } G_1 \cap G_2 = \emptyset, \text{ then } F(G) = F(G_1)F(G_2). \quad (1.15)$$

In fact, since flow is measured over edges and not vertices, this claim can be strengthened to biconnected components:

$$\text{If } G = G_1 \cup G_2 \text{ and } G_1 \cap G_2 = \{v\}, \text{ then } F(G) = F(G_1)F(G_2). \quad (1.16)$$

See [14] for proofs of these identities and [13, 68] for other decomposition formulas. In addition, for planar graphs, many of the identities developed for chromatic polynomials can also be used in the computation of flow polynomials. Intuitively, in planar graphs, vertex coloring is equivalent to face-coloring, and face-coloring is generalized by nowhere-zero flows. Thus, it can be expected that the chromatic and flow polynomials are very similar in planar graphs. Indeed, the following result by Jaeger [69] confirms this intuition:

$$\text{If } G \text{ is planar, then } F(G) = \frac{1}{t}P(G^*). \quad (1.17)$$

Thus, in planar graphs, the identities developed for chromatic polynomials can also be used in the computation of flow polynomials. For example, this duality relation is used in Chapter 3 to compute the flow polynomials of outerplanar graphs from the chromatic polynomials of their duals.

1.1.3 Other graph polynomials

As mentioned earlier, the chromatic and flow polynomials are special cases of the two-variable Tutte polynomial $T(G; x, y)$, which contains a great deal of information about the graph and has many far-reaching connections. A detailed study of the Tutte polynomial is outside the scope of this thesis, but it is worth noting the following relations between the Tutte polynomial and the chromatic and flow polynomials.

$$P(G; t) = (-1)^{n-k}t^k T(G; 1-t, 0) = \sum_{S \subseteq E} (-1)^{|S|} t^{\text{comp}(G:S)} \quad (1.18)$$

$$F(G; t) = (-1)^{m-n+k} T(G; 0, 1-t) = (-1)^m \sum_{S \subseteq E} (-1)^{|S|} t^{\nu(G:S)} \quad (1.19)$$

Here, $\text{comp}(G : S)$ is the number of connected components in the spanning graph $G : S$, and $\nu(G : S)$ is the cyclomatic number of $G : S$. These closed formulas are alternate definitions for the chromatic and flow polynomials of any graph. However, the summation in these closed formulas is over an exponentially large set, since $|\{S : S \subset E\}| = O(2^{n^2})$. Thus, these closed formulas cannot be efficiently used in practice to compute chromatic and flow polynomials. Another useful identity of Kook, Reiner, and Stanton [70] expresses the Tutte polynomial in terms of the chromatic and flow polynomials of its minors:

$$T(G; x, y) = \sum_{S \subset E} T(G/S; x, 0) T(G : S; 0, y). \quad (1.20)$$

Sokal [71] surveys the applications of the Tutte polynomial and its single-variable specializations to statistical mechanics, solid-state physics, and electrical circuit theory. In addition, his survey brings out many connections and relations between graph polynomials, matroids, and practical models.

Besides the chromatic and flow polynomials, the Tutte polynomial of a connected graph G also contains as special cases the *bad coloring polynomial*, which counts all possible colorings of G (not just proper colorings), the *reliability polynomial*, which measures the probability that deleting an edge of G with fixed probability p does not disconnect G , the *shelling polynomial*, which deals with orderings of the facets of simplicial complexes associated with G , and the *Jones polynomial*, which deals with knots associated with G ; see [72, 73, 74] for more details. These polynomials have a number of important applications in physics, engineering, and combinatorics; they are also useful for computing or estimating certain graph invariants, which are of interest in extremal graph theory and are important characteristics of large networks [75, 76, 77]. For example, the reliability polynomial is used in network theory to

model the resilience of a network to random edge failures [73].

Graph polynomials which are not direct specializations of the Tutte polynomial have also been studied. For example, the *domination polynomial* [78] of a graph G counts the number of dominating sets of G of size d . Work in this direction includes derivations of recurrence relations [79], analysis of the roots [80], and characterizations for specific graphs [81, 82]. Similar results have been obtained for the *connected domination polynomial* [83], *independence polynomial* [84], *clique polynomial* [85], *vertex cover polynomial* [86], and *edge cover polynomial* [87], which are defined as the generating functions of their eponymous sets. For more definitions, results, and applications of graph polynomials, see the survey of Ellis-Monaghan and Merino [88] and the bibliography therein.

1.2 Zero forcing and related problems

In contrast to the classical static vertex and face colorings discussed in the previous sections, an assortment of dynamically evolving graph colorings have gained prominence in recent years. These coloring processes — known as *zero forcing*, *infection*, *propagation*, and by several other names — arose independently, and almost simultaneously, in several different settings including linear algebra [89], quantum physics [90], theoretical computer science [91], and electrical engineering [92]. The wide interest in zero forcing has created a rapidly growing research community, which has generated a large volume of work in the last several years. In this section, I survey some of the problems which led to the formulation of zero forcing, as well as some variants of zero forcing and related problems. I first give a purely graph theoretic introduction to zero forcing, which is the primary perspective taken throughout this thesis.

1.2.1 Zero forcing

Let $G = (V, E)$ be a graph and $S \subset V$ be a set of *initially colored* vertices, all other vertices being *uncolored*. The *color change rule* dictates that at each integer-valued timestep, a colored vertex u with a single uncolored neighbor v *forces* that neighbor to become colored; such a *force* is denoted $u \rightarrow v$. The *derived set* of S is the set of colored vertices obtained after the color change rule is applied until no new vertex can be forced; it can be shown that the derived set of S is uniquely determined by S . A *zero forcing set* is a set whose derived set is all of V ; the *zero forcing number* of G , denoted $Z(G)$, is the minimum cardinality of a zero forcing set. See Figure 1.7 for an illustration.

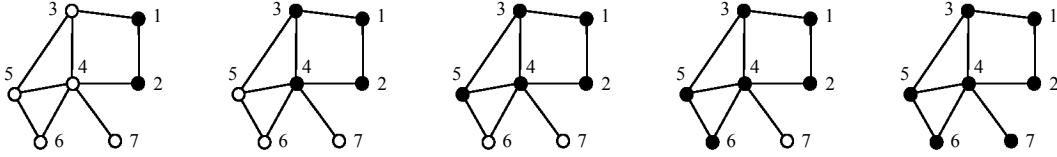


Figure 1.7 : *Left*: A minimum zero forcing set of the graph is marked by colored vertices. Then, from left to right, the following forces are applied: $1 \rightarrow 3$ and $2 \rightarrow 4$; $3 \rightarrow 5$; $5 \rightarrow 6$; $4 \rightarrow 7$.

A *chronological list of forces* of S is a sequence of forces applied to obtain the derived set of S in the order they are applied; there can also be initially colored vertices which do not force any vertex. Generally, the chronological list of forces is not uniquely determined by S ; for example, it may be possible for several colored vertices to force an uncolored vertex at a given step. A *forcing chain* for a chronological list of forces is a maximal sequence of vertices (v_1, \dots, v_k) such that $v_i \rightarrow v_{i+1}$ for $1 \leq i \leq k - 1$. A *singleton chain* is a forcing chain consisting of a single vertex, i.e., an initially colored vertex which does not force any vertex. If a vertex forces another

vertex at some step of the forcing process, then it cannot force a second vertex at a later step, since that would imply it had two uncolored neighbors when it forced for the first time. Thus, each forcing chain induces a distinct path in G , one of whose endpoints is an initially colored vertex and the rest of which is uncolored at the initial timestep; we will say the initially colored vertex *initiates* the forcing chain, and we will call the other endpoint of the forcing chain a *terminal* vertex. The set of all forcing chains for a chronological list of forces is uniquely determined by the chronological list of forces and forms a path cover of G . In Figure 1.7, the forcing chains associated with the chronological list of forces given in the caption are $(1, 3, 5, 6)$ and $(2, 4, 7)$.

Zero forcing was introduced in a 2006 AIM workshop on linear algebra and graph theory [89] and was used to bound the maximum nullity (equivalently, the minimum rank) of the family of symmetric matrices associated with a graph; see Section 1.2.2 for details. The zero forcing number is generally more attainable than the maximum nullity, which makes it a valuable tool in the study of this algebraic parameter. Zero forcing was also independently studied in quantum physics [90], theoretical computer science [91], and electrical engineering [92]; see Sections 1.2.3 and 1.2.4 for details. Subsequently, applications of zero forcing in physics [93], logic circuits [94], modeling the spread of diseases and information in social networks [95, 96], and bounding or approximating various other graph parameters [97, 98, 99, 100, 101, 102] have also been explored.

Computing the zero forcing number was shown to be NP-complete [103, 91]; thus, the majority of research in this area has focused on developing structural results on zero forcing sets [89, 104, 105], bounds on the zero forcing number [106, 107, 108], relating the zero forcing number to other graph parameters [99, 97, 109], and characterizing the zero forcing numbers of graphs with special structure [110, 111,

112, 113]. The number of timesteps in which an initially colored set of vertices forces the graph, called the *propagation time*, is also a problem of interest — not just for zero forcing [114, 115, 116], but also for its variants [117, 118]. The state-of-the-art solver for computing the zero forcing numbers of moderately-sized graphs is the Wavefront algorithm developed by Grout et al. [119]. Several integer programming alternatives have been proposed in [120], which are generally slower but more flexible than Wavefront, and can accommodate additional constraints to the problem.

1.2.2 Minimum rank

Let $S_n(\mathbb{R})$ denote the set of real symmetric $n \times n$ matrices. For a matrix $A \in S_n(\mathbb{R})$, $\mathcal{G}(A)$ denotes the graph with vertex set $\{1, \dots, n\}$ and edge set $\{\{1, j\} : a_{ij} \neq 0, 1 \leq i < j \leq n\}$. Note that the diagonal of A is not used when constructing $\mathcal{G}(A)$. The set of *symmetric matrices associated with a graph G* is defined to be $\mathcal{S}(G) = \{A \in S_n(\mathbb{R}) : \mathcal{G}(A) = G\}$. The *minimum rank* of G is defined as the minimum rank over all symmetric matrices associated with G , i.e., $\text{mr}(G) = \min\{\text{rank}(A) : A \in \mathcal{S}(G)\}$. Similarly, the *maximum nullity* of G is defined as the maximum nullity over all symmetric matrices associated with G , i.e., $M(G) = \max\{\text{null}(A) : A \in \mathcal{S}(G)\}$. Since for any symmetric $n \times n$ matrix, $\text{rank}(A) + \text{null}(A) = n$, it follows that $\text{mr}(G) + M(G) = n$. Thus, the minimum rank and maximum nullity problems are essentially equivalent.

The minimum rank problem has received considerable attention (see, e.g., [121, 122, 123, 124, 125, 126, 127]) and is related to various other problems in linear algebra and graph theory. However, since by definition the minimum rank is computed over an infinite family of matrices, direct computation of this parameter is usually impossible; instead, research has often focused on approximating or bounding the minimum rank

by combinatorial or algebraic means. It was this goal of finding an accessible bound to the minimum rank (equivalently, the maximum nullity) of a graph that gave rise to zero forcing. To illustrate the motivation, I give the following example of computing the maximum nullity of a path P_4 .

Example 1.6. $M(P_4) = 1$.

Proof. Consider the family of matrices $\mathcal{S}(P_4)$; these matrices can be described by the sparsity pattern

$$A = \begin{bmatrix} a_1 & a_2 & 0 & 0 \\ a_2 & a_3 & a_4 & 0 \\ 0 & a_4 & a_5 & a_6 \\ 0 & 0 & a_6 & a_7 \end{bmatrix},$$

where a_2, a_4, a_6 are nonzero real numbers, and a_1, a_3, a_5, a_7 are arbitrary real numbers.

In particular, we may assume that A is a matrix which realizes $M(P_4)$, i.e., $\text{null}(A) = M(P_4)$. Let $x = [x_1 \ x_2 \ x_3 \ x_4]^T$ be a null vector of A . Then, we have the following equations:

$$a_1x_1 + a_2x_2 = 0 \tag{1.21}$$

$$a_2x_1 + a_3x_2 + a_4x_3 = 0 \tag{1.22}$$

$$a_4x_2 + a_5x_3 + a_6x_4 = 0 \tag{1.23}$$

$$a_6x_3 + a_7x_4 = 0$$

Suppose $x_1 = 0$; then, in (1.21), x_2 is forced to be zero; in (1.22), x_3 is forced to be zero (since x_1 and x_2 are zero), and in (1.23), x_4 is forced to be zero (since x_2 and x_3 are zero). Thus, if $Ax = 0$ and $x \neq 0$, it follows that $x_1 \neq 0$.

Now, let $X = \{x \in \mathbb{R}^4 : x_1 = 0\}$ and $K = \ker(A)$. Clearly, $\dim(X) = 3$; moreover, since we found that every nonzero x in $\ker(A)$ has $x_1 \neq 0$, it follows that

$\dim(X \cap K) = 0$. By the well-known formula for the dimension of the intersection and sum of finite-dimensional subspaces, $\dim(X + K) + \dim(X \cap K) = \dim(X) + \dim(K)$.

Thus,

$$\begin{aligned} M(G) &= \text{null}(A) = \dim(K) = \dim(X + K) + \dim(X \cap K) - \dim(X) \\ &= \dim(X + K) + 0 - 3 \leq 4 - 3 = 1, \end{aligned}$$

where the last inequality follows from the fact that $\dim(X + K) \leq \dim(\mathbb{R}^4) = 4$. On the other hand, the matrix

$$A_0 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

is in $\mathcal{S}(P_4)$ and has nullity 1, so $1 = \text{null}(A_0) \leq M(P_4) \leq 1$. □

What allows us to bound $M(G)$ from above in Example 1.6 is the choice of an appropriate set of indices I , such that when $x_i = 0$ for all $i \in I$, all other entries of x are also forced to be zero. In particular, the principle by which a set of zero entries of x forces other entries to be zero is precisely the color change rule in the corresponding graph: when all-but-one terms in the left-hand-side of $A_i^T x = 0$ are zero for some row A_i^T of A , the remaining term (and the corresponding entry of x) must also be zero. For example, this is the case in (1.21) when choosing x_1 to be zero forces x_2 to be zero, which in turn forces x_3 and x_4 to be zero*. Equivalently, selecting x_1 to be zero corresponds to choosing the end-vertex of P_4 as a zero forcing set of P_4 . The same principle can be applied to any family of symmetric matrices and the corresponding graph to obtain an upper bound on $M(G)$; this is stated formally below.

*This process is the source of the nomenclature of “zero forcing”.

Theorem 1.5 ([89]). *For any graph G , $M(G) \leq Z(G)$.*

Thus, the zero forcing number can be used as a relatively tractable bound on the maximum nullity of a graph. Further connections between these two parameters have been investigated in subsequent papers, including, e.g., the effects of vertex and edge operations on $M(G)$ and $Z(G)$ [113, 104, 128], conditions for equality or disparity between $M(G)$ and $Z(G)$ [106, 129, 130], and characterizations of $M(G)$ and $Z(G)$ for specific families of graphs [131, 132]. In addition, many variants of zero forcing have been developed to bound other linear algebraic quantities; see Section 1.2.4 for more details.

1.2.3 Power domination

Electric power networks consist of energy-producing generators, energy-consuming loads, transmission lines connecting the generators and loads, and busses where transmission lines intersect. An electric power company must constantly monitor the state of its network in order to detect system failures and assure that demands are being met. To this end, phase measurement units (PMUs) are placed at select locations around the network; these devices measure the voltage at the bus where they are placed, and the phase angle at the transmission lines incident with the bus (cf. [133]). The PMU readings are then synchronized in processing stations, where the data from multiple PMUs is leveraged with physical laws governing the behavior of electrical circuits (such as Ohm's laws and Kirchoff's laws), in order to gain information about parts of the network which are not being directly monitored. Thus, the state of the entire network can be determined from partial information measured at appropriate locations. Because of the high cost of the equipment, labor, and communication infrastructure associated with installing and maintaining PMUs, electric power com-

panies aim to use the smallest number of PMUs necessary to maintain full control of the network.

Electrical power networks can naturally be represented as graphs, where the generators, loads, and busses are vertices, and the transmission lines are edges. The problem of interest is then to select a minimum set of vertices from which the entire graph can be observed according to certain rules. The selected vertices correspond to the locations where PMUs should be placed in the electrical network in order to monitor the entire network at minimum cost. In particular, below are the rules (introduced in [92]) by which vertices and edges can be observed; these rules reflect information gained by direct measurements of PMUs, as well as information gained through Ohm's and Kirchoff's laws about locations in the network which are not directly monitored by PMUs.

1. All vertices at which a PMU is placed are observed
2. All edges incident to a vertex at which a PMU is placed are observed
3. A vertex incident to an observed edge is observed
4. An edge joining two observed vertices is observed
5. If a vertex is incident to $k > 1$ edges and $k - 1$ of these edges are observed, then all k are observed

It can be verified that this process is identical to zero forcing (where observed vertices are colored), with the exception that at the first timestep, by rules 2 and 3, the closed neighborhood of the set of initially colored vertices is colored. In other words, the initially colored set of vertices performs a "domination" step, and then the color change rule is applied iteratively as in zero forcing. Due to this domination

step and the application to power network monitoring, this problem is referred to as *power domination*; a *power dominating set* is a set of vertices which causes the entire graph to be observed under the rules above, and the *power domination number* of a graph G , denoted $\gamma_P(G)$, is the minimum cardinality of a power dominating set of G .

Since a set is contained in its closed neighborhood, it follows that any zero forcing set is also a power dominating set. Thus, for any graph $G = (V, E)$, $\gamma_P(G) \leq Z(G)$. Moreover, $S \subset V$ is a power dominating set of G if and only if $N[S]$ is a zero forcing set; thus, if S is a minimum power dominating set,

$$\gamma_P(G) \leq Z(G) \leq |N[S]| \leq |S|(\Delta(G) + 1) = \gamma_P(G)(\Delta(G) + 1).$$

Empirically, the zero forcing number is typically easier to compute than the power domination number, and has been used as a technical tool in the derivation of certain results about power domination; see [109, 134] for some recent work in this direction. The power domination problem is NP-complete even for bipartite graphs, chordal graphs, and split graphs [135, 92], although efficient algorithms and approximations are available for the power domination numbers of trees [92], interval graphs [135], graphs of bounded treewidth [136], block graphs [137], grid graphs [138], product graphs [139, 140], claw-free cubic graphs [141], and several other families.

1.2.4 Graph searching, quantum control, and zero forcing variants

Zero forcing also arose independently in the context of graph search problems. The objective of graph search problems is to capture a fugitive hidden on the vertices or edges of a graph while using a limited number of guards, search actions, or other resources. The earliest graph search problems which have received considerable attention are the edge search and node search problems, introduced by Megiddo et al.

[142] and Kirousis and Papadimitriou [143], respectively. In these problems, the aim is to capture an invisible fugitive using the smallest number of guards, where search actions consist of placing a guard at a vertex, removing a guard from a vertex, or sliding a guard along an edge. Bienstock and Seymour [144] introduced the mixed search problem which combines the edge and node search problems, and focuses on minimizing the number of guards used at any step. Dyer et al. [145] introduced the fast search problem, which focuses on minimizing the number of search actions in which the fugitive is captured; see also [146, 147] for some recent work on this model.

Yang [91] introduced the fast-mixed search problem, which combines the fast search and the mixed search models. In the fast-mixed search problem, an invisible fugitive can freely move at great speed from one vertex to another along a guard-free path. A vertex or edge where the fugitive may hide is *contaminated*, otherwise it is *cleared*; a vertex is *occupied* if it has a guard on it. The search actions include *placing* a guard on a contaminated vertex and *sliding* a guard along a contaminated edge uv from u to v if v is contaminated and all edges incident to u except uv are cleared. A contaminated edge becomes cleared if both endpoints are occupied by guards or if a guard slides along it from one endpoint to the other. The *fast-mixed search number* of a graph G , denoted $\text{fms}(G)$ is the minimum number of guards required to clear all edges of G (i.e., to capture a fugitive hiding in G).

In [148], it was shown that the search actions in the fast-mixed search model can be interpreted precisely as the zero forcing color change rule, leading to the following identity.

Theorem 1.6 ([148]). *For any graph G , $Z(G) = \text{fms}(G)$.*

This relation allows results and techniques developed for graph search algorithms to be applied to zero forcing and minimum rank problems, and vice versa.

Finally, zero forcing was also studied in quantum control theory under the name *propagation* [90], where it was introduced as a scheme for controlling large quantum systems by acting on small subsystems satisfying certain conditions. A quantum system in this setting can be represented as a graph, and the protocol for operating on the system through local quantum transformations can be described in graph terminology as follows: each vertex in an initially selected set has a packet of information which has to be diffused among all the vertices of the graph; a vertex v can pass its packet to an adjacent vertex w only if w is the only neighbor of v which still does not have the information [149]. It is easy to see that this propagation protocol is identical to the zero forcing color change rule, and that the smallest number of particles sufficient to control the system is the zero forcing number of the corresponding graph. This equivalence has enabled the use of techniques and results from zero forcing in quantum control theory, paving the way for applications like control of quantum hard drives and quantum RAM [90]; see [150, 151] for other recent work in quantum control theory involving propagation.

In addition to the equivalent formulations of zero forcing in different disciplines, there are numerous variants of zero forcing, obtained by modifying the color change rule or by adding certain restrictions to the structure of a zero forcing set. These variants can roughly be divided into two categories: those which are designed to bound different linear algebraic parameters, and those which are obtained by natural graph theoretic alterations. I briefly discuss a representative sample of these variants below; this survey is not meant to be exhaustive.

One of the most popular variants of zero forcing is *positive semidefinite zero forcing* [128, 152, 153, 148, 154, 118], obtained by modifying the color change rule to act separately on certain induced subgraphs; the minimum cardinality sets which force a

graph using this modified rule are used to study the maximum nullity of the positive semidefinite matrices associated with a graph. A different modification of the color change rule yields a variant called *Z_q -forcing* [100], which can be used to bound the maximum nullity of the matrices associated with a graph that have q negative eigenvalues. Similarly, *skew zero forcing* [102] can be used to bound the minimum rank among all skew-symmetric matrices associated with a graph, and *signed zero forcing* [101] can be used to bound the maximum nullity of a matrix with a given sign pattern; *fractional zero forcing* [155] generalizes skew zero forcing and positive semidefinite zero forcing. *Failed zero forcing* [156, 157] and *failed skew zero forcing* [158] are respectively concerned with finding the largest set of vertices which is not a zero forcing set, and the largest set of vertices which is not a skew zero forcing set. In *probabilistic zero forcing* [159], colored vertices can force their uncolored neighbors independently with a certain probability. In *k -forcing* [160, 161, 95], a colored vertex with at most k uncolored neighbors can force those neighbors to become colored. *Total forcing* [162, 163] is concerned with zero forcing sets which induce subgraphs without isolated vertices. Variants of zero forcing for directed graphs [164, 165] and graphs with loops [99, 166] have also been studied.

The next section describes and motivates another variant of zero forcing, obtained by requiring connectivity of the solution set; studying this variant is the second main direction of this thesis.

1.2.5 Connected zero forcing

A natural graph theoretic variant of zero forcing is obtained by requiring every set of initially colored vertices to induce a connected subgraph. More precisely, a *connected zero forcing set* of a connected graph G is a zero forcing set of G which induces a

connected subgraph. The *connected zero forcing number* of G , denoted $Z_c(G)$, is the cardinality of a minimum connected zero forcing set of G . For short, these will be referred to as *connected forcing set* and *connected forcing number*. Note that a disconnected graph cannot have a connected forcing set.

In this thesis, I explore the differences and similarities between zero forcing and connected forcing, establish numerous structural, extremal, and enumerative results about connected forcing sets, and characterize the connected forcing numbers of several families of graphs. Studying connected zero forcing can further the understanding of the zero forcing process and the underlying structure of zero forcing sets in general. Requiring a zero forcing set to be connected also has meaningful interpretations in many of the applications and physical phenomena modeled by zero forcing.

For example, in power network monitoring, there are often significant costs associated with the high-speed communication infrastructure between the PMUs and the processing stations which collect and manage PMU data; there may also be costs associated with dispatching a technician to regulate or maintain the PMUs and related equipment. In a scenario where these costs outweigh the production costs of the PMUs, an electric power company may seek to place all PMUs in a compact, connected region in the network in order to decrease the costs incurred by processing stations, communication infrastructure, and technician travel times. This motivates the *connected power domination* problem (and the associated *connected power domination number* $\gamma_{P,c}(G)$), where the number of PMUs necessary to monitor the network is minimized subject to the condition of connectivity. It is easy to see that in general, the zero forcing number is neither an upper nor lower bound on the connected power domination number; however, the connected forcing number is an upper bound on the connected power domination number, since any connected zero forcing set is a con-

nected power dominating set. Thus, connected forcing can be used as a technical tool in the derivation of certain results about connected power domination, just as zero forcing is used in relation to power domination. In the fast mixed search perspective on zero forcing, connectivity reflects a scenario where all the guards are starting their search from a connected region in the graph. As another example, it is often the case that ideas or diseases originate from a single connected source in a social network or geographic region; thus, connected forcing may be better suited than zero forcing to model propagation in those scenarios.

More generally, the connected variants of many other graph problems have been extensively studied. For example, the connected vertex cover [167, 168, 169, 170] connected k -path vertex cover [171], and t -total vertex cover [172] problems have been investigated in theoretical computer science, mainly in the context of parameterized complexity and approximation algorithms. The connected domination [173, 174, 175, 176] and connected power domination problems [177] have been investigated in combinatorics and graph theory, and have been linked to other graph parameters like the maximum leaf number (the largest number of leaves in any spanning tree).

Imposing connectivity often fundamentally changes the nature of a problem, including its complexity, structural properties, and applications. For example, while both domination and connected domination are NP-complete [178], the latter is generally much harder to solve exactly. The current best algorithm for solving the domination problem has runtime $O(1.5048^n)$ [179], while until recently, the only exact algorithm for the connected domination problem was brute force enumeration with runtime $O(2^n)$; this was marginally improved to $O(1.9407^n)$ [176] and later to $O(1.8966^n)$ [180]. This disparity has been attributed in [176] to the non-locality of the connected domination problem, since exact algorithms often rely on the local structure of the

graph and are unable to capture global properties such as connectivity.

In some cases, however, efficient computation of connected variants of problems (and zero forcing in particular) is possible. Trivially, if the connected forcing number of a graph G is known to be very small or very large, an enumeration approach can be used to find a minimum connected forcing set in polynomial time. For example, if $k_1 \leq Z_c(G) \leq k_2 < \frac{n}{2}$, it can be checked whether each of the $\binom{n}{k_1} + \dots + \binom{n}{k_2}$ sets of vertices of appropriate size is connected and forcing in $O(n^2)$ time, so $Z_c(G)$ can be computed in $O((k_2 - k_1)n^{2+k_2})$ time. An enumeration approach can also be used to efficiently compute the connected forcing number of graphs with polynomially many connected induced subgraphs. This graph class includes arbitrary subdivisions of fixed graphs, and graphs with bounded maximum leaf numbers; see [181] for another dynamic graph coloring process which can be solved efficiently in such graphs. In [120], my coauthors and I proposed several combinatorial and integer programming approaches for computing the zero forcing and connected forcing numbers of a graph, and compared their performance in a variety of random graphs. Our computational experiments showed that the proposed algorithms for connected forcing were faster, and able to handle larger graphs, than the zero forcing algorithms. This work is not discussed further in this thesis, but is presented in detail in the thesis of Fast [182].

Chapter 2

Preliminaries

This chapter recalls select graph theoretic notions and operations; some additional definitions will be included in later chapters when needed. The background material presented here is meant to be a quick reference rather than a self-contained foundation; the reader is referred to [47, 183] for a detailed introduction to graph theory.

2.1 Graph definitions

Different contexts call for different types of graphs. In the study of graph polynomials, it is often useful to consider graphs with loops and multiple edges, and — when studying flow — graphs with directed edges. Zero forcing is typically studied in the context of simple undirected graphs, but there are also variants defined on directed graphs and graphs with loops.

Definition 2.1. A *simple graph* $G = (V, E)$ consists of a vertex set V and an edge set E of distinct two-element subsets of V .

Definition 2.2. A *multigraph* $G = (V, E)$ consists of a vertex set V and an edge multiset E of not necessarily distinct one- or two-element subsets of V . A *loop* in a multigraph is an edge which is a one-element subset of V and a *multiple edge* is an edge which appears more than once in E .

Definition 2.3. A *directed graph* $G = (V, E)$ consists of a vertex set V and an edge set E of ordered pairs of elements of V . Given a directed edge $e = (u, v)$, u is the

tail of e — denoted $t(e)$ — and v is the *head* of e — denoted $h(e)$. If E is allowed to contain multiple copies of the same edge, or loop edges of the form (v, v) , then G is a *directed multigraph*.

Unless otherwise stated, the first of the three definitions will be intended in the sequel by the term ‘graph’. In addition, for notational simplicity, $e = uv$ will stand for an undirected edge $e = \{u, v\}$ or a directed edge $e = (u, v)$ when there is no scope for confusion.

Given a multigraph $G = (V, E)$, the number of times an edge e appears in E is the *multiplicity* of e . The *underlying set* of E is the set E' which contains the (unique) elements of E . For example, if $E = \{e_1, e_1, e_2, e_3, e_3, e_3\}$, then $E' = \{e_1, e_2, e_3\}$. The *underlying simple graph* of $G = (V, E)$ is the graph $G' = (V, E' \setminus \{e : e \text{ is a loop}\})$.

An *orientation* of G is an assignment of directions to the edges of G ; more precisely, for each edge $e = uv \in E$, one of u and v is assigned to be $h(e)$ and the other is assigned to be $t(e)$. An orientation of G is *acyclic* if it creates no directed cycles and it is *totally cyclic* if it makes every edge belong to a directed cycle.

The *disjoint union* of sets S_1 and S_2 , denoted $S_1 \dot{\cup} S_2$, is a union operation that indexes the elements of the union set according to which set they originated in; the *disjoint union* of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, denoted $G_1 \dot{\cup} G_2$, is the graph $(V_1 \dot{\cup} V_2, E_1 \dot{\cup} E_2)$. The *join* of two graphs G_1 and G_2 , denoted $G_1 \vee G_2$, is the graph obtained from $G_1 \dot{\cup} G_2$ by adding an edge between each vertex of G_1 and each vertex of G_2 . The *corona* of two graphs G_1 and G_2 , denoted $G_1 \circ G_2$, is the graph obtained by taking one copy of G_1 and $|V(G_1)|$ copies of G_2 , and adding an edge between the i^{th} vertex of G_1 and each vertex of the i^{th} copy of G_2 , $1 \leq i \leq |V(G_1)|$. The *Cartesian product* of two graphs G_1 and G_2 , denoted $G_1 \square G_2$, is the graph with vertex set $V(G_1) \times V(G_2)$, where vertices (u, u') and (v, v') are adjacent in $G_1 \square G_2$ if

and only if either $u = v$ and u' is adjacent to v' in G_2 , or $u' = v'$ and u is adjacent to v in G_1 . The *complement* of a graph $G = (V, E)$ is the graph $G^c = (V, E^c)$.

2.2 Graph parameters and operations

Let $G = (V, E)$ be a graph. The *order* and *size* of G will be denoted by $n = |V|$ and $m = |E|$, respectively. Two vertices $v, w \in V$ are said to be *adjacent*, or *neighbors*, if there exists the edge $(v, w) \in E$. If v is adjacent to w , we write $v \sim w$; otherwise, we write $v \not\sim w$. The *neighborhood* of $v \in V$ is the set of all vertices which are adjacent to v , denoted $N(v; G)$. The *degree* of $v \in V$ is defined as $d(v; G) = |N(v; G)|$. The minimum degree and maximum degree of G will be denoted as $\delta(G)$ and $\Delta(G)$, respectively. The dependence of these parameters on G can be omitted when it is clear from the context. A *leaf*, or *pendant*, is a vertex with degree 1. An *isolated vertex* or *isolate* is a vertex with degree 0; such a vertex may also be called a *trivial (connected) component* of G . The number of connected components of G will be denoted by $\text{comp}(G)$.

A *separating set* of G is a set of vertices which, when removed, increases the number of connected components of G . A *cut vertex* is a separating set of size one. The *vertex connectivity* of G , denoted $\kappa(G)$, is the largest integer such that G remains connected whenever fewer than $\kappa(G)$ vertices of G are removed; a disconnected graph has vertex connectivity zero. A *cut edge* or *bridge* is an edge which, when removed, increases the number of connected components of G . A *biconnected component*, or *block*, of G is a maximal subgraph of G which has no cut vertices; G is *biconnected* if it has no cut vertices. An *outer block* is a block which contains at most one cut vertex of G . A *trivial block* is a block with two vertices, i.e., a cut edge of G . The *block tree* of G is the bipartite graph with parts A and B , where A is the set of cut

vertices of G and B is the set of blocks of G , and where $a \in A$ is adjacent to $b \in B$ if and only if b contains a in G .

Given $S \subset V$, the *induced subgraph* $G[S]$ is the subgraph of G whose vertex set is S and whose edge set consists of all edges of G which have both endpoints in S . If \mathcal{F} is a set of graphs, a graph is \mathcal{F} -free if it does not contain F as an induced subgraph for each $F \in \mathcal{F}$. Given $S \subset E$, the *spanning subgraph* $G : S$ is the subgraph of G whose vertex set is V and whose edge set is S . A set $S \subset V$ is an *independent set* if $G[S]$ has no edges, and it is a *clique* if $G[S]$ is a complete graph. The *clique number* of G , denoted $\omega(G)$, is the size of the largest clique in G . A subset $S \subset V$ is a *dominating set* if $N[S] = V$; the *domination number* of G , denoted $\gamma(G)$ is the size of the smallest dominating set in G .

Given $u, v \in V$, the *contraction* G/uv is obtained by deleting edge uv if it exists, and identifying u and v into a single vertex. Note that the edge uv does not have to be in E for G/uv to be defined, and G/uv results in the same graph regardless of whether or not uv is in E . The *subdivision* of edge $e = uv$ is obtained by adding a new vertex w and replacing uv with edges uw and wv ; a subdivision of G is a graph obtained by subdividing some of the edges of G . Given $v \in V$ and $e \in E$, the notations $G - e$, $G + e$, and $G - v$ respectively denote the graphs $(V, E \setminus \{e\})$, $(V, E \cup \{e\})$, and $(V \setminus \{v\}, \{e \in E : v \text{ is not an endpoint of } e\})$.

An isomorphism between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ will be denoted by $G_1 \simeq G_2$. We will also say G_1 and G_2 are *homeomorphic* if there exist subdivisions of G_1 and G_2 which are isomorphic. Likewise, G_1 and G_2 are *amallamorphic* if (V_1, E'_1) is isomorphic to (V_2, E'_2) , where E'_1 and E'_2 are the underlying sets of E_1 and E_2 .

A *vertex join* of $G = (V, E)$ is the graph $G_V = (V \cup \{v^*\}, E \cup \{vv^* : v \in V\})$,

where $v^* \notin V$. In other words, it is the graph obtained by joining a new vertex to each of the existing vertices of G . See Figure 2.1 for an illustration of a vertex join.

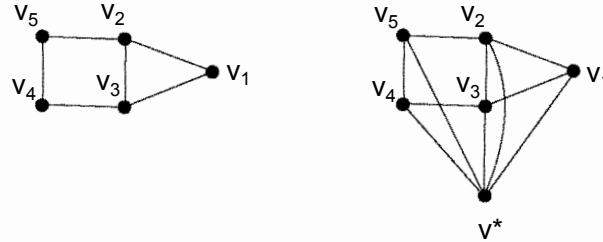


Figure 2.1 : *Left:* A graph H . *Right:* H_V , the vertex join of H .

2.3 Special types of graphs

The complete graph, path, and cycle, on n vertices are respectively denoted K_n , P_n , and C_n . A graph with no edges will be called an *empty graph* and denoted $(K_n)^c$. A graph is *bipartite* with *parts* A and B if every edge of G has one endpoint in A and the other endpoint in B . A *complete bipartite* graph, denoted $K_{p,q}$ is the complement of $K_p \dot{\cup} K_q$ (we may allow these indices to equal 0, in which case $K_{n,0} \simeq K_{0,n} \simeq \dot{\bigcup}_{i=1}^n K_1$). A *complete multipartite* graph, denoted K_{a_1, \dots, a_k} is the complement of $\dot{\bigcup}_{i=1}^k K_{a_i}$. A *tree* is a connected graph with no cycles, and a *forest* is the disjoint union of trees. A *unicyclic graph* is a connected graph with exactly one cycle. A *uniclique graph* is a connected graph with exactly one block of size greater than 2, with the additional property that this block is a clique. A *cactus graph* is a graph in which every block is a cycle or a cut edge, and a *block graph* is a graph in which every block is a clique.

A *planar* graph is a graph which can be drawn in the plane so that none of its edges cross. A graph drawn in such a way is called a *plane* graph. If G is a plane graph, its *dual* G^* is a graph that has a vertex corresponding to each face of G , and

an edge joining the vertices corresponding to faces of G which share an edge. Note that if G is connected, $G = (G^*)^*$. The *weak dual* of G is the subgraph of G^* whose vertices correspond to the bounded faces of G . See Figure 2.2 for an illustration of the dual of the *house graph* H . An *outerplanar* graph is a planar graph which has a plane embedding where all of its vertices lie on the outer face; a graph drawn in such a way is called an *outerplane* graph. Trees, cactus graphs, and minimal polygon triangulations are examples of outerplanar graphs.

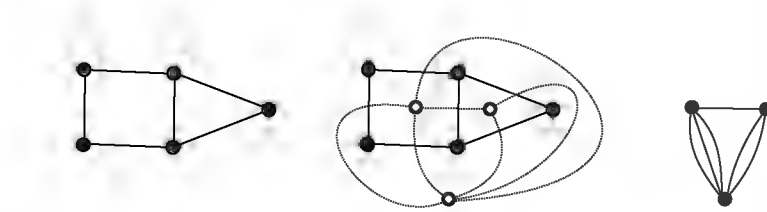


Figure 2.2 : *Left:* The house graph H . *Middle:* H and its dual. *Right:* H^* , the dual of H . The graphs H and H^* are outerplanar and biconnected.

Chapter 3

Characterizations of chromatic and flow polynomials

As discussed in Chapter 1, computing the chromatic and flow polynomials of a graph are very challenging tasks. These problems are $\#P$ -hard for general graphs, and even for bipartite planar graphs and sparse graphs with $|E| = O(|V|)$ [18]. In fact, most of the coefficients of the chromatic and flow polynomials of general graphs cannot even be approximated (by a fully polynomial randomized approximation scheme; see [184, 18]). Thus, a large volume of work has focused on exploiting the structure of special families of graphs in order to derive closed formulas, algorithms, or heuristics for computing their chromatic and flow polynomials. In particular, classes of graphs which are generalizations of trees, cliques, and cycles are frequently investigated.

For example, Wakelin and Woodall [185] characterized the chromatic polynomials of polygon trees and biconnected outerplanar graphs; Whitehead [186, 187] characterized the chromatic polynomials of q -trees; Lazuka [188] derived closed formulas for the chromatic polynomials of cactus graphs; Gordon [189] studied the Tutte polynomials of rooted trees; Mphako-Banda [190, 191] derived closed formulas for the chromatic, flow, and Tutte polynomials of flower graphs.

In this chapter, I consider several other families of graphs and present closed form expressions and efficient algorithms for computing their chromatic and flow polynomials. These results are also computationally compared against a general-purpose solver and shown to have superior performance.

3.1 Characterizations of chromatic polynomials

Recall the definition of a vertex join from Chapter 2; this operation adds a new vertex to a graph G and joins it to each of the existing vertices of G . I now propose a generalization of this concept, and characterize the chromatic and flow polynomials of several families of graphs obtained using this operation. Given a graph $G = (V, E)$, a *multiset* S over V is a collection of vertices of V , each of which may appear more than once in S .

Definition 3.1. Let $G = (V, E)$ be a graph, S be a multiset over V , and $v^* \notin V$. The *generalized vertex join of G using S* is the graph $G_S = (V \cup \{v^*\}, E \cup \{vv^* : v \in S\})$.

In other words, G_S is the graph obtained by joining a new vertex to some (or all) of the existing vertices of G , possibly more than once. Note that if the multiplicity of v in S is p , there are p parallel edges between v and v^* in G_S . See Figure 3.1 for an illustration of a generalized vertex join.

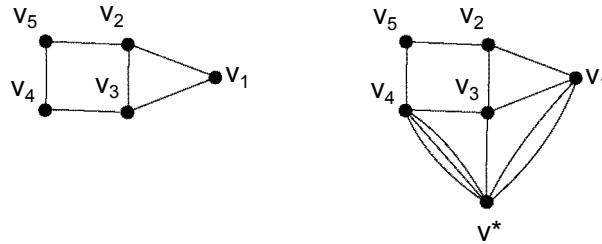


Figure 3.1 : *Left:* A graph H . *Right:* H_S , the generalized vertex join of H using $S = \{v_1, v_1, v_3, v_4, v_4, v_4\}$.

Remark 3.1. Let $G = (V, E)$ be any graph, S be a multiset over V , and S' be the underlying set of S . Recall that by (1.7), $P(G_S) = P(G_{S'})$. Thus, when computing the chromatic polynomial of G_S , we can assume without loss of generality that the

multiplicity of every element in S is 1. The reason the definition of a generalized vertex join allows multisets instead of sets of vertices is because allowing certain multiple edges in a class of graphs corresponds to a larger class of dual graphs. In turn, this can lead to broader dual results about flow polynomials.

For instance, in the next section, I compute the chromatic polynomials of generalized vertex joins of trees. I then show that the duals of these graphs are outerplanar graphs, where the added vertex v^* is the one corresponding to the outer face. Allowing multiple edges between v^* and each vertex of the tree means the family of duals includes all outerplanar graphs, instead of ones for which at most one edge from each bounded face borders the outer face. Thus, I am able to state a broader result about flow polynomials. A similar principle is used in Section 3.2.2 with the flow polynomials of generalized vertex join cycles.

3.1.1 Generalized vertex join trees

Let $T = (V, E)$ be a tree with $|V| = n$, S be a multiset over V , and let T_S be the generalized vertex join of T using S . See Figure 3.2 for an illustration. For short, T_S will be called a *generalized vertex join tree*. In this section, I present an efficient algorithm to compute $P(T_S)$, the chromatic polynomial of a generalized vertex join tree.

First, by Remark 3.1, $P(T_S) = P(T_{S'})$ where S' is the underlying set of S ; however, to simplify notation, we will simply assume that the multiplicity of every element in S is 1 when computing $P(T_S)$. This restriction will be lifted when computing flow polynomials of outerplanar graphs in the next chapter.

Two special cases of T_S occur when $|S| = 0$ and when $|S| = 1$. In the first case, T_S consists of a tree on n vertices and an isolated vertex. Thus, by (1.4) and (1.10),

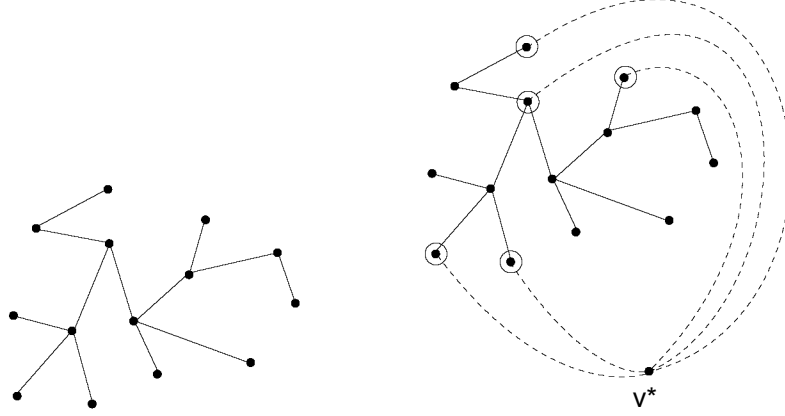


Figure 3.2 : Forming a generalized vertex join tree T_S from a given tree T and a subset of its nodes S .

$P(T_S) = t^2(t-1)^{n-1}$. In the second case, T_S is a tree on $n+1$ vertices, so by equation (1.10), $P(T_S) = t(t-1)^n$. Thus, from now on, we will assume that $|S| \geq 2$.

Next, suppose there are b bridges in T_S , and let B be the set of vertices in T_S which are an endpoint of some bridge, but do not belong to a cycle. Note that since $|S| \geq 2$, there is at least one cycle, so not all edges of T_S are bridges. Let $T'_S = T_S - B$. Using (1.3), each bridge with a degree 1 endpoint can be separated from the rest of the graph, adding a factor of $\frac{P(K_2)}{P(K_1)} = \frac{t(t-1)}{t}$ to the chromatic polynomial of the resulting graph; once every bridge in T_S is removed, the resulting graph is T'_S and

$$P(T_S) = P(T'_S)(t-1)^b. \quad (3.1)$$

See Figure 3.3 for an illustration of T'_S . In this graph, we define the indicator function $f : V(T'_S) \setminus \{v^*\} \rightarrow \{0, 1\}$ by $f(v) = 1$ if $v \in S$, $f(v) = 0$ if $v \notin S$.

We now introduce some definitions which are analogous to standard notions in graph theory and are slightly modified to suit our purposes. For simplicity, we will refer to these terms by the names of their standard analogues (cf. [192]).

First, select an arbitrary vertex $r \neq v^*$ in T'_S called a *root*. The *level* of a node in

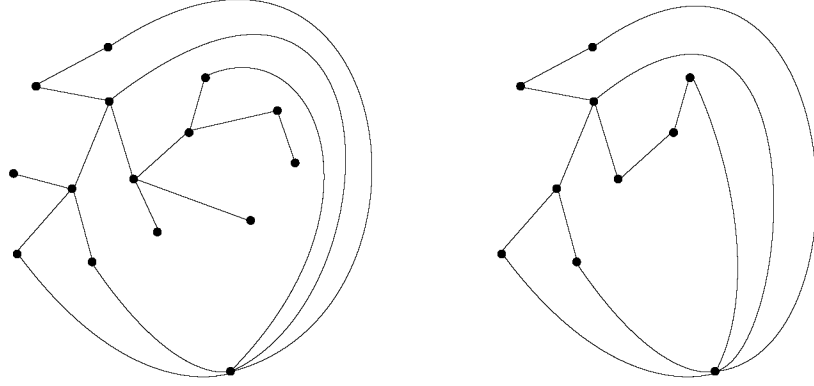


Figure 3.3 : Removing the bridges of T_S to form T'_S .

T'_S is given by the function $L : V(T'_S) \setminus \{v^*\} \rightarrow \mathbb{N} \cup \{0\}$ by $L(v) = \text{dist}(r, v)$, where $\text{dist}(r, v)$ is the length of the shortest path between r and v in $T'_S - v^*$. Denote by $L_i(T'_S)$ the set of nodes at the i^{th} level; more precisely, $L_i(T'_S) = \{v : L(v) = i\}$. Let \mathcal{L} be the *height* of T'_S , i.e., $\mathcal{L} = \max\{L(v) : v \in V(T'_S) \setminus \{v^*\}\}$.

If $L(v) = i$, w is a *child* of v if w is adjacent to v and $L(w) = i + 1$. Vertex z is a *descendant* of v if $z = v^*$ or if there is a path v, p_1, \dots, p_r, z such that $L(v) < L(p_1) < \dots < L(p_r) < L(z)$. The set of all descendants of v is denoted $D(v)$. See Figure 3.4 for an illustration of the levels in T'_S .

Finally, we will specify some subgraphs of T'_S to be used in the sequel. The purpose of these subgraphs is to facilitate an expression of $P(T'_S)$ in terms of the chromatic polynomials of smaller generalized vertex join trees, which in turn facilitates a recursive computation of $P(T'_S)$. For any $a \in V(T'_S) \setminus \{v^*\}$, we define:

- $T_a = T'_S[a \cup D(a)]$; this is a generalized vertex join tree with root a , which includes all of the descendants of a in T_S .
- $\tilde{T}_c = T'_S[\{a\} \cup \{c\} \cup D(c)]$; this is a generalized vertex join tree with root a ,

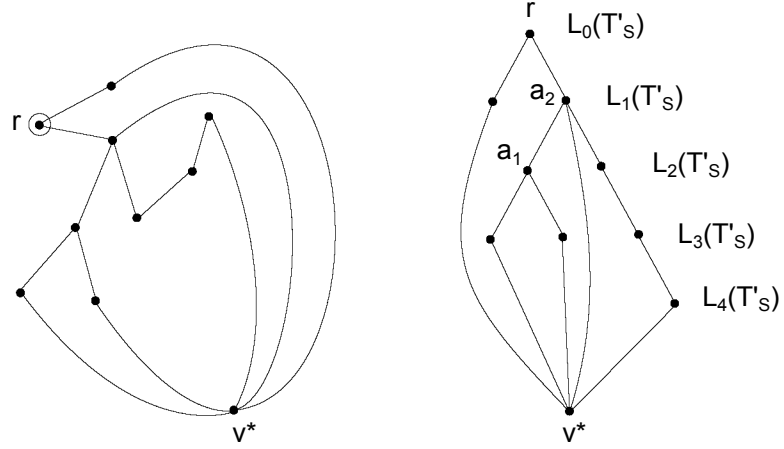


Figure 3.4 : Selecting a node r and finding $L(v)$ and $L_i(T'_S)$.

which includes only the descendants of c in T_S .

- $H_a = T_a/av^*$; this is essentially a generalized vertex join of a forest with root a : since $T_a - v^*$ is a tree, $T_a - v^* - a$ is a forest, and a is connected to some subset of the other vertices.
- $\tilde{H}_c = \tilde{T}_c/av^*$, this is one ‘branch’ of the generalized vertex join forest H_a , and is also a generalized vertex join tree with root c , which includes all of the descendants of c in T'_S (possibly with an extra connection between c and v^*).

See Figure 3.5 for an illustration of these subgraphs.

With this in mind, let $a \neq v^*$ be a vertex with children c_1, \dots, c_k , and suppose we know $P(T_{c_i})$ and $P(H_{c_i})$ for $1 \leq i \leq k$. Let $I = \{i : f(c_i) = 1\}$ and $Z = \{i : f(c_i) = 0\}$ be the sets of children of a which are connected and not connected to v^* , respectively. Then, we can compute $P(H_a)$ as follows.

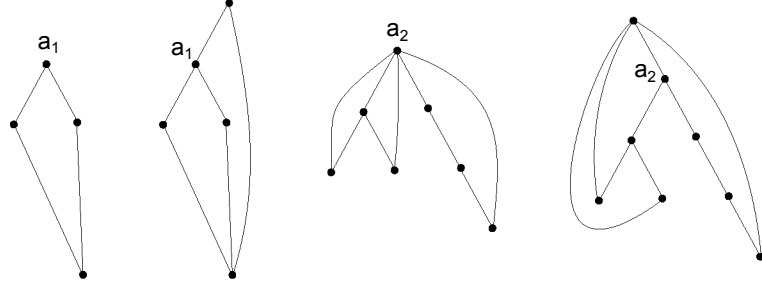


Figure 3.5 : *From left to right: T_{a_1} ; \tilde{T}_{a_1} ; H_{a_2} ; \tilde{H}_{a_2} , for two vertices a_1 and a_2 of the graph T'_S shown in Figure 3.4, right.*

$$\begin{aligned}
 P(H_a) &= \frac{1}{t^{k-1}} \prod_{i=1}^k P(\tilde{H}_{c_i}) \\
 &= \frac{1}{t^{k-1}} \prod_I P(\tilde{H}_{c_i}) \prod_Z P(\tilde{H}_{c_i}) \\
 &= \frac{1}{t^{k-1}} \prod_I P(T_{c_i}) \prod_Z (P(T_{c_i}) - P(H_{c_i}))
 \end{aligned}$$

Here, the first equality follows from (1.3) and the definition of \tilde{H}_{c_i} , since $\tilde{H}_{c_1}, \dots, \tilde{H}_{c_k}$ all have only the vertex a in common. The second equality is obtained by partitioning $\{1, \dots, k\}$ into I and Z . Finally, if a was originally connected to v^* , then $\tilde{H}_{c_i} = T_{c_i}$; otherwise, the deletion-contraction formula (1.2) yields $P(\tilde{H}_{c_i}) = P(\tilde{H}_{c_i} - ac) - P(\tilde{H}_{c_i}/ac) = P(T_{c_i}) - P(H_{c_i})$, and the third equality follows.

Next, we compute $P(T_a)$ by considering two cases: a is either in S or not. Let $P_1(T_a) = P(T_a)$, where $f(a) = 1$, and $P_0(T_a) = P(T_a)$, where $f(a) = 0$. Clearly, $P(T_a) = f(a)P_1(T_a) + (1 - f(a))P_0(T_a)$. We now find $P_1(T_a)$ and $P_0(T_a)$ separately as follows.

$$\begin{aligned}
P_1(T_a) &= \frac{1}{(t(t-1))^{k-1}} \prod_{i=1}^k P(\tilde{T}_{c_i}) \\
&= \frac{1}{(t(t-1))^{k-1}} \prod_I P(\tilde{T}_{c_i}) \prod_Z P(\tilde{T}_{c_i}) \\
&= \frac{\prod_I (P(T_{c_i})(t-2))}{(t(t-1))^{k-1}} \prod_Z (P(\tilde{T}_{c_i} + c_i v^*) + P(\tilde{T}_{c_i}/c_i v^*)) \\
&= \frac{\prod_I (P(T_{c_i})(t-2))}{(t(t-1))^{k-1}} \prod_Z (P(T_{c_i} + c_i v^*)(t-2) + P(H_{c_i})(t-1)) \\
&= \frac{\prod_I (P(T_{c_i})(t-2))}{(t(t-1))^{k-1}} \prod_Z ((P(T_{c_i}) - P(H_{c_i}))(t-2) + P(H_{c_i})(t-1)) \\
&= \frac{1}{(t(t-1))^{k-1}} \prod_I (P(T_{c_i})(t-2)) \prod_Z ((t-2)P(T_{c_i}) + P(H_{c_i}))
\end{aligned}$$

$$P_0(T_a) = P(T_a + av^*) + P(T_a/av^*) = P_1(T_a) + P(H_a)$$

In the computation of $P_1(T_a)$, the first equality follows from (1.3) and the definition of \tilde{T}_{c_i} , since $\tilde{T}_{c_1}, \dots, \tilde{T}_{c_k}$ all have the edge av^* in common, which is a clique of size 2. The second equality is obtained by partitioning $\{1, \dots, k\}$ into I and Z . In the third equality, the vertices a , c_i , and v^* form a clique of size 3 in \tilde{T}_{c_i} for $\{c_i : i \in I\}$; this clique is connected to the rest of \tilde{T}_{c_i} by the edge $c_i v^*$, which is a clique of size 2. Moreover, the rest of the graph is precisely T_{c_i} ; thus, (1.3) is applied to obtain $P(\tilde{T}_{c_i}) = \frac{P(T_{c_i})P(K_3)}{P(K_2)} = P(T_{c_i})(t-2)$. For the vertices $\{c_i : i \in Z\}$, the addition-contraction formula (1.1) is applied to add the edge $c_i v^*$ to get $P(\tilde{T}_{c_i}) = P(\tilde{T}_{c_i} + c_i v^*) + P(\tilde{T}_{c_i}/c_i v^*)$. In the fourth equality, the graph $\tilde{T}_{c_i} + c_i v^*$ for $\{c_i : i \in Z\}$ is the same as the graph \tilde{T}_{c_i} for $\{c_i : i \in I\}$; thus, a similar argument as before can be used to show that $P(\tilde{T}_{c_i} + c_i v^*) = P(T_{c_i} + c_i v^*)(t-2)$ (by separating a clique of size 3 using (1.3)). Moreover, $\tilde{T}_{c_i}/c_i v^*$ is precisely H_{c_i} with the additional edge ac_i . This edge can be separated from H_{c_i} using (1.3): $P(\tilde{T}_{c_i}/c_i v^*) = \frac{P(H_{c_i})P(K_2)}{P(K_1)} = P(H_{c_i})(t-1)$. The

fifth equality follows from the deletion-contraction formula (1.2) applied to the edge $c_i v^*$, so that $P(T_{c_i} + c_i v^*) = P(T_{c_i} + c_i v^* - c_i v^*) - P((T_{c_i} + c_i v^*) / c_i v^*) = P(T_{c_i}) - P(H_{c_i})$. Finally, the last equality is obtained by simple algebraic manipulations.

In the computation of $P_0(T_a)$, the first equality follows from the addition contraction formula (1.1), as the edge av^* is added. Then, by the definitions of P_1 and H_a , $P(T_a + av^*) = P_1(T_a)$ and $P(T_a / av^*) = P(H_a)$ and the second equality follows.

Thus, I have shown how to express $P(T_a)$ and $P(H_a)$ in terms of $P(T_{c_i})$ and $P(H_{c_i})$, $1 \leq i \leq k$. Using these identities, I propose the following algorithm for finding the chromatic polynomial of a generalized vertex join tree T_S .

Algorithm 1

1. Find and remove the bridges of T_S to acquire T'_S
2. For $i = \mathcal{L}$ to 0

Compute $P(T_a)$ and $P(H_a)$ for each $a \in L_i(T'_S)$
3. Compute $P(T_S)$ using (3.1)

Theorem 3.1. *Algorithm 1 finds the correct chromatic polynomial of a generalized vertex join tree T_S using $O(n^2 \log n)$ time and $O(n)$ space.*

Proof. It was shown in (3.1) and the preceding discussion that by finding the bridges of T_S and the chromatic polynomial of T'_S , $P(T_S)$ can be easily computed as well. Thus, we only need to verify that Step 2 of the algorithm correctly computes $P(T'_S)$.

It was already established that for every $a \in V(T'_S) \setminus \{v^*\}$, $P(T_a)$ and $P(H_a)$ can be expressed in terms of $P(T_c)$ and $P(H_c)$ for every child c of a . Note that this

expression is trivially satisfied for vertices which have no children. By construction, vertices in $L_{\mathcal{L}}(T'_S)$ have no children, so $P(T_a)$ and $P(H_a)$ can be found immediately for any vertex $a \in L_{\mathcal{L}}(T'_S)$. For $\mathcal{L} > i \geq 0$, a vertex a in $L_i(T'_S)$ either has no children, or has all of its children in $L_{i+1}(T'_S)$. In either case, $P(T_c)$ and $P(H_c)$ are known for every child c of a — either vacuously or inductively. Thus, $P(T_a)$ and $P(H_a)$ can also be computed using the formulas derived earlier in this section. Since by construction, $P(T'_S) = P(T_r)$ and $L_0(T'_S) = \{r\}$, Algorithm 1 indeed finds the correct chromatic polynomial of T_S .

To verify the time-complexity of the algorithm, let $|V(T)| = n$. The bridges in T_S and the graph T'_S can be found in $O(n^2)$ time* by successively finding and deleting degree 1 vertices of T_S . Also, the level and list of children of each vertex of $T'_S - v^*$ can be found with $O(n)$ time by a breadth-first scan.

Each evaluation of $P(T_a)$ and $P(H_a)$ requires the multiplication of $O(a_k)$ polynomials, where a_k is the number of children of a . Since we evaluate $P(T_a)$ and $P(H_a)$ for $O(n)$ vertices, and the total number of children in T'_S is $O(n)$, the evaluation of $P(T'_S)$ requires the multiplication of $O(n)$ polynomials. Each of these polynomials has degree at most $O(n)$, since $P(T'_S)$ has degree $O(n)$. The time-complexity of multiplying two polynomials of degree n , using a Fast Fourier Transform, is $O(n \log n)$, so the total time complexity of Algorithm 1 is $O(n^2 \log n)$.

Finally, to verify the space-complexity, note that the total number of vertices in the set of graphs $\{T_a, H_a : a \in L_i(T'_S)\}$ is at most $O(n)$. Recall that the chromatic polynomial of a graph with k vertices has degree k ; hence, the sum of the degrees of the set of polynomials $\{P(T_a), P(H_a) : a \in L_i(T'_S)\}$ is $O(n)$. A set of polynomials

*By the restriction that every element in S has multiplicity 1, $|E(T_S)| = O(n)$. Thus, using the algorithm of Tarjan [192], all bridges of T_S can actually be found in $O(n)$ time.

whose degrees add up to n can be stored with $O(n)$ space. Thus, since we only have to store the polynomials $P(T_a)$ and $P(H_a)$ for a in one level at a time, the total space-complexity of Algorithm 1 is $O(n)$. \square

3.1.2 Generalized vertex join cycles

Let $C = (V, E)$ be a cycle, S be a multiset over V and let C_S be the generalized vertex join of C using S . For short, C_S will be called a *generalized vertex join cycle*. In the literature, graphs of a similar form have also been called “generalized wheel” graphs, and have been investigated by other approaches and in different contexts (cf. [193, 194, 12]). In the remainder of this section and in Section 3.2.2, I will present closed formulas for $P(C_S)$ and $F(C_S)$ in a unified framework.

Suppose the generalized vertex join cycle C_S is equipped with a “wheel” plane embedding obtained by placing v^* in the bounded face of a plane drawing of C , and drawing edges from the vertices in S to v^* so that the resulting graph remains plane. Since cycles have a unique plane embedding, the “wheel” embedding of C_S is unique up to topological conjugacy. Moreover, since chromatic and flow polynomials are independent of embedding, this embedding can be considered without loss of generality. The vertices along the outer face of C_S will be labeled in clockwise order as v_1, \dots, v_n ; see Figure 3.6 for an illustration. The edges incident to v^* will be called *spokes*.

If $S = \emptyset$, then by (1.11), $P(C_S) = tP(C_n) = t((t-1)^n + (-1)^n(t-1))$; thus, suppose hereafter that $S \neq \emptyset$ and consider S' , the underlying set of S . By (1.7), $P(C_S) = P(C_{S'})$. Without loss of generality, suppose that $S' = \{v_{a_1}, \dots, v_{a_s}\}$ where $1 = a_1 < \dots < a_s$. Also, let e_1, \dots, e_s be the spokes of $C_{S'}$, with $e_i = v_{a_i}v^*$, and F_1, \dots, F_s be the faces of $C_{S'}$, with F_i clockwise of edge e_i ; see Figure 3.7 for an

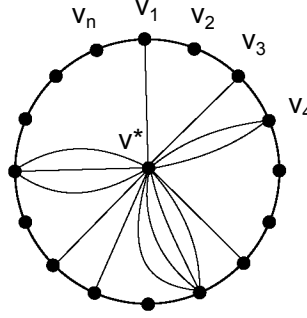
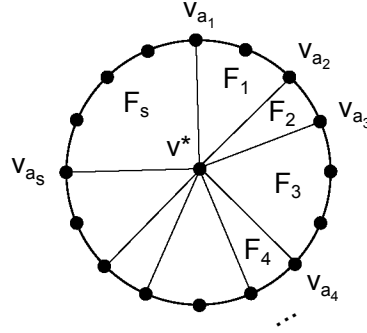
Figure 3.6 : Generalized vertex join cycle C_S .

illustration.

Figure 3.7 : $C_{S'}$, the underlying graph of C_S .

Let f_i be the size of face F_i , i.e., the number of edges along the boundary of F_i , with cut edges being counted twice. It is easy to see that $f_i = 2 + a_{i+1} - a_i$ for $1 \leq i \leq s-1$ and $f_s = 2 + (n+1) - a_s$.

With this in mind, some auxiliary graphs will be introduced in order to express $P(C_S)$ as a combination of the chromatic polynomials of simpler graphs. For $1 \leq i \leq s$, define $C_{S'}^i = C_{S'} - \{e_i, \dots, e_s\}$ and for notational simplicity, $C_{S'}^{s+1} = C_{S'}$. Then, applying the deletion-contraction formula (1.2) consecutively on the edges e_s, \dots, e_1

yields the following identity; see Figure 3.8 for an illustration of this decomposition.

$$\begin{aligned}
P(C_S) &= P(C_S - e_s) - P(C_S/e_s) \\
&= P(C_S^s) - P(C_S^{s+1}/e_s) \\
&= P(C_S^{s-1}) - P(C_S^s/e_{s-1}) - P(C_S^{s+1}/e_s) \\
&\vdots \\
&= P(C_S^1) - P(C_S^2/e_1) - \dots - P(C_S^s/e_{s-1}) - P(C_S^{s+1}/e_s) \\
&= tP(C_n) - \sum_{i=1}^s P(C_S^{i+1}/e_i).
\end{aligned} \tag{3.2}$$

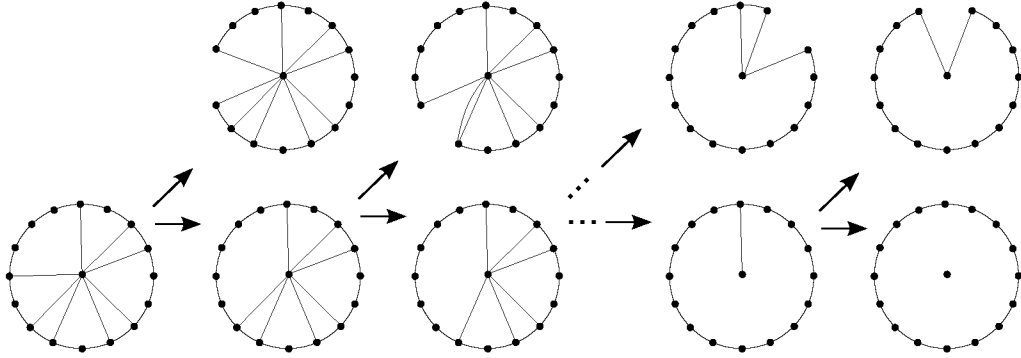


Figure 3.8 : Decomposing $C_{S'}$ (on far left) into simpler graphs as described in (3.2). Using (1.3), the graphs in the top row can be further decomposed into the cycles making up their bounded faces.

Thus, $P(C_{S'})$ is decomposed into the chromatic polynomials of the collection of graphs $\{C_{S'}^{i+1}/e_i\}_{i=1}^s$. The faces of $C_{S'}$ can be regarded as cycles of sizes f_1, \dots, f_s ; thus, the graphs $\{C_{S'}^{i+1}/e_i\}_{i=1}^s$ can be further decomposed into the cycles making up their bounded faces. Let U_i be the face of $C_{S'}^{i+1}$ corresponding to the union of F_i, \dots, F_s after edges e_{i+1}, \dots, e_s are deleted. Then, the faces of $C_{S'}^{i+1}$ have sizes f_1, \dots, f_{i-1}, u_i ,

where u_i is the size of U_i ; more precisely,

$$\begin{aligned}
u_i &= 2 + (f_i - 2) + \dots + (f_s - 2) \\
&= 2 + (-2)(s - i + 1) + \sum_{j=i}^s f_j \\
&= 2(i - s) + \sum_{j=i}^s f_j.
\end{aligned} \tag{3.3}$$

Let J_i be the multiset of sizes of faces of $C_{S'}^{i+1}/e_i$, i.e., $J_1 = \{n\}$ and for $2 \leq i \leq s$,

$$J_i = \{f_1, \dots, f_{i-2}, f_{i-1} - 1, u_i - 1\}. \tag{3.4}$$

Then, starting from a face of $C_{S'}^{i+1}/e_i$ which borders the contracted edge, and using the fact that this face shares just one edge (which is a clique of size 2) with the rest of the graph, (1.3) can be successively applied to decompose $C_{S'}^{i+1}/e_i$ into cycles with sizes in J_i in order to evaluate $P(C_{S'}^{i+1}/e_i)$. In particular, $P(C_{S'}^{i+1}/e_i) = P(K_2)^{1-i} \prod_{j \in J_i} P(C_j)$. Thus, by (3.2) we have

$$\begin{aligned}
P(C_S) &= P(C_{S'}) = tP(C_n) - \sum_{i=1}^s \frac{\prod_{j \in J_i} P(C_j)}{P(K_2)^{i-1}} \\
&= t((t-1)^n + (-1)^n(t-1)) - \sum_{i=1}^s \frac{\prod_{j \in J_i} ((t-1)^j + (-1)^j(t-1))}{(t(t-1))^{i-1}}.
\end{aligned} \tag{3.5}$$

Note that formula (3.5) depends only on the sequence of face-sizes of $C_{S'}$ and hence only on S .

3.1.3 Generalized vertex join cliques

Let $K = (V, E)$ be a complete graph, S be a multiset over V and let K_S be the generalized vertex join of K using S . For short, we will call K_S a *generalized vertex join clique*. Let $|V| = n$, S' be the underlying set of S , and $|S'| = s$. Then,

$$P(K_S) = P(K_{S'}) = \frac{P(K_n)P(K_{s+1})}{P(K_s)} = (t-s) \prod_{i=0}^{n-1} (t-i), \quad (3.6)$$

where the first equality follows from equation 1.7, the second follows from (1.3) — since $K_{S'}[S' \cup \{v^*\}] = K_{s+1}$, and the third follows from (1.9).

Since in general complete graphs are not planar, graph duality cannot be applied to generalized vertex join cliques to obtain a result about flow polynomials. However, it would be interesting to investigate the flow polynomials of generalized vertex join cliques directly. This will likely be a challenging task: Tutte [8] derived a formula and a generating function for the flow polynomial of a complete graph which is quite complicated; adding a vertex with arbitrary connections to the others will complicate this formula even more. Such investigations will be the focus of future work.

3.2 Characterizations of flow polynomials

In this section, I show that the family of outerplanar graphs is dual to the family of generalized vertex join trees, and adapt Algorithm 1 from Section 3.1.1 to compute the flow polynomials of outerplanar graphs. These results complement and expand the work of Wakelin and Woodall [185] on chromatic polynomials of outerplanar graphs, by characterizing the flow polynomials of outerplanar graphs and the chromatic polynomials of their duals. I also show that the family of generalized vertex join cycles is self-dual, and transform (3.5) into a closed formula for computing the flow polynomials of generalized vertex join cycles.

3.2.1 Outerplanar graphs

Let B be a biconnected outerplane graph with bounded faces F_1, \dots, F_s and outer face F_* . The weak dual of B is a tree $T = (V, E)$, where vertex $v_i \in T$ corresponds

to face $F_i \in B$ (see [195] for more details). Suppose F_i shares f_i edges with F_* , and let v^* be the vertex in the dual of B corresponding to F_* . Then, the dual of B is the generalized vertex join tree T_S , where S is the multiset over V in which v_i appears f_i times.

With this in mind, the flow polynomial of an arbitrary outerplanar graph G can be computed by applying Algorithm 1 to the dual of each biconnected component of G . This procedure is formally outlined in Algorithm 2 below.

Algorithm 2

1. Find the biconnected components G_1, \dots, G_k of G
2. Find the dual graphs G_1^*, \dots, G_k^*
3. Compute $P(G_1^*), \dots, P(G_k^*)$ using Algorithm 1
4. Compute $F(G)$ by $F(G) = \frac{1}{t^k} \prod_{i=1}^k P(G_i^*)$

Theorem 3.2. *Algorithm 2 finds the correct flow polynomial of an outerplanar graph G using $O(n^2 \log n)$ time and $O(n)$ space.*

Proof. Consider the biconnected components of G as separate graphs, i.e., $G_i = G[V_i]$ where V_i is a maximal subset of $V(G)$ such that $G[V_i]$ is biconnected. Then, each G_i is a biconnected outerplanar graph and by (1.16) and (1.17),

$$F(G) = \prod_{i=1}^k F(G_i) = \frac{1}{t^k} \prod_{i=1}^k P(G_i^*).$$

Since the dual of a biconnected outerplanar graph is a generalized vertex join tree,

Algorithm 1 can be used to compute $P(G_i^*)$ for $1 \leq i \leq k$, so Algorithm 2 indeed finds the correct flow polynomial of G .

To verify the time- and space-complexity, let $|V(G)| = n$ and $|V(G_i)| = n_i$; clearly $n_1 + \dots + n_k = O(n)$. By the algorithms of Hopcroft and Tarjan [196, 197], the biconnected components G_1, \dots, G_k of G can be found, embedded in the plane, and have their dual graphs G_1^*, \dots, G_k^* computed, with $O(n)$ time and space. Finally, note that Algorithm 1 runs with $O(n_i^2 \log n_i)$ time on the generalized vertex join tree G_i^* and that

$$\sum_{i=1}^k (n_i^2 \log n_i) \leq \left(\sum_{i=1}^k n_i \right)^2 \log \left(\sum_{i=1}^k n_i \right) = O(n^2 \log n).$$

Hence, Algorithm 1 can be applied to find $P(G_1^*), \dots, P(G_k^*)$ in $O(n^2 \log n)$ time and $O(n)$ space. Thus, the total time complexity of Algorithm 2 is $O(n^2 \log n)$ and the total space complexity is $O(n)$. \square

I conclude this section with a characterization of the duality between outerplanar graphs and generalized vertex join trees.

Proposition 3.1. *Let G be a simple biconnected outerplane graph and T_S be its dual generalized vertex join tree. G is simple if and only if every vertex of T_S has degree at least 3.*

Proof. Suppose G is a simple biconnected outerplane graph. G has no parallel edges or loops, so G has no faces of size 1 or 2. Thus, each face of G is incident to at least 3 edges, so each vertex of T_S has degree at least 3.

Now, suppose T_S is a generalized vertex join tree, and that every vertex of T_S has degree at least 3. We will show that T_S is the dual of a simple biconnected outerplanar graph by induction on the number of vertices of T_S . If T_S has two vertices v and v^* ,

all the edges in T_S must join v to v^* since by construction, T_S can have no loops. Thus, T_S is the dual of some cycle of size at least 3 (which is simple, biconnected, and outerplanar). Next, let T_S be a generalized vertex join tree on $k + 1$ vertices with minimum vertex degree at least 3, and let v be a leaf of T . Since T is a tree, v has a unique neighbor u in T with exactly one edge between u and v . Moreover, by assumption, v must be connected to v^* by $\ell \geq 2$ edges and u must be incident to at least two edges other than uv . Thus, if we delete v from T_S and add an edge from u to v^* , we obtain a generalized vertex join tree on k vertices, which by induction is the dual of some simple biconnected outerplanar graph G . In this graph, u corresponds to some bounded face F and v^* corresponds to the outer face F_* . Since we added an edge uv^* , F shares at least one edge e with F_* . Now, if we glue a cycle of size $\ell + 1$ to e , we obtain a simple biconnected outerplanar graph whose dual is T_S . \square

3.2.2 Generalized vertex join cycles

Let C_S be a generalized vertex join cycle. If $S = \emptyset$, then $F(G) = t - 1$; thus assume hereafter that $S \neq \emptyset$. To compute the flow polynomial of C_S , note that by (1.17), $F(C_S) = \frac{1}{t}P(C_S^*)$, where C_S^* is the dual of C_S . But C_S^* is again a generalized vertex join cycle. To see why, note that each bounded face of C_S is incident to two spokes — hence the weak dual of C_S is a cycle; in addition, each bounded face of C_S may share any number of edges with the outer face, making the vertex of C_S^* corresponding to the outer face of C_S a generalized vertex join. See Figure 3.9 for an illustration.

Let $\tilde{s} = |S|$ and $s = |S'|$ where S' is the underlying set of S . Let \tilde{C} be the weak dual of C_S ; \tilde{C} is a cycle with \tilde{s} vertices. Let \tilde{S} be the multiset of vertices of \tilde{C} such that $C_S^* = \tilde{C}_{\tilde{S}}$ and let \tilde{S}' be the underlying set of \tilde{S} . Then,

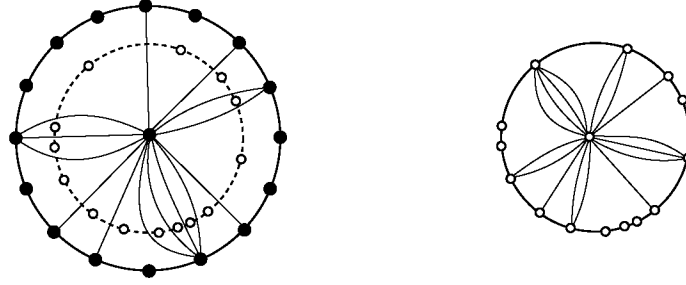


Figure 3.9 : *Left:* C_S and its weak dual. *Right:* C_S^* , the dual of C_S , is also a generalized vertex join cycle.

$$F(C_S) = \frac{1}{t}P(C_S^*) = \frac{1}{t}P(\tilde{C}_{\tilde{S}}) = \frac{1}{t}P(\tilde{C}_{\tilde{S}'}). \quad (3.7)$$

It is easy to see that $C_{S'}$ and $\tilde{C}_{\tilde{S}'}$ have the same number of faces. Moreover, if $\tilde{f}_1, \dots, \tilde{f}_s$ are the sizes of the faces of $\tilde{C}_{\tilde{S}'}$ in clockwise order, then \tilde{f}_i equals the multiplicity of v_{a_i} in S plus 2. Thus, to find $F(C_S)$, we simply plug in the sequence of face-sizes of C_S^* into (3.3), (3.4), and (3.5) as follows:

$$F(C_S) = (t-1)^{\tilde{s}} + (-1)^{\tilde{s}}(t-1) - \frac{1}{t} \sum_{i=1}^s \frac{\prod_{j \in \tilde{J}_i} ((t-1)^j + (-1)^j(t-1))}{(t(t-1))^{i-1}},$$

where $\tilde{J}_1 = \{\tilde{s}\}$ and $\tilde{J}_i = \{\tilde{f}_1, \dots, \tilde{f}_{i-2}, \tilde{f}_{i-1} - 1, 2(i-s) + \sum_{j=i}^s \tilde{f}_j - 1\}$ for $2 \leq i \leq s$. Note that this closed formula again depends only on S , since the face-sizes of C_S^* are determined from S .

3.3 Computational results

In this section, Algorithm 1 and formulas (3.5) and (3.6) are computationally compared to the general-purpose `ChromaticPolynomial` function found in Version 10.0 of the computer algebra system Mathematica. The `ChromaticPolynomial` function is an implementation of the deletion-contraction algorithm for finding the chromatic

polynomials of general graphs. The computations described in this section were performed on an HP-Pavilion desktop with an *Intel®Core™ 2 Quad Q9300 2.50GHz* processor. All coding was done in Mathematica, which has a native ability to multiply and divide polynomials and manipulate graphs. Other than that, no high-level functions were used in the implementation of Algorithm 1 and formulas (3.5) and (3.6).

3.3.1 Generalized vertex join trees

A test graph T_S was created by first generating a random tree T on n vertices, then adding a new vertex and connecting it to all of the leaves of T plus a random subset of the other vertices of T .

The level of each vertex was found by computing its distance from a randomly chosen root r , and the children of each vertex were identified as the adjacent vertices with a higher level. Finally, the sets I and Z were computed for each vertex by intersecting the set of its children with the set of neighbors of v^* . Then, the polynomials $P(T_a)$ and $P(H_a)$ were computed for each a as described in Section 3.1.1, with $P(T_r)$ giving $P(T_S)$. The order of the graph n was varied and the corresponding computation time was recorded.

The chromatic polynomials given by Algorithm 1 exactly matched those given by the `ChromaticPolynomial` function. However, the `ChromaticPolynomial` function was only able to handle graphs with $n \leq 65$ before running out of memory, and Algorithm 1 was able to handle much larger graphs. In addition, for $n \leq 65$, the computation time of Algorithm 1 was between 0.001 and 0.15 seconds whereas the `ChromaticPolynomial` function was more than 10 times slower. See Figure 3.10, right, for the run times of Algorithm 1 and Figure 3.10, left, for the run times of the

ChromaticPolynomial function on graphs of increasing order.

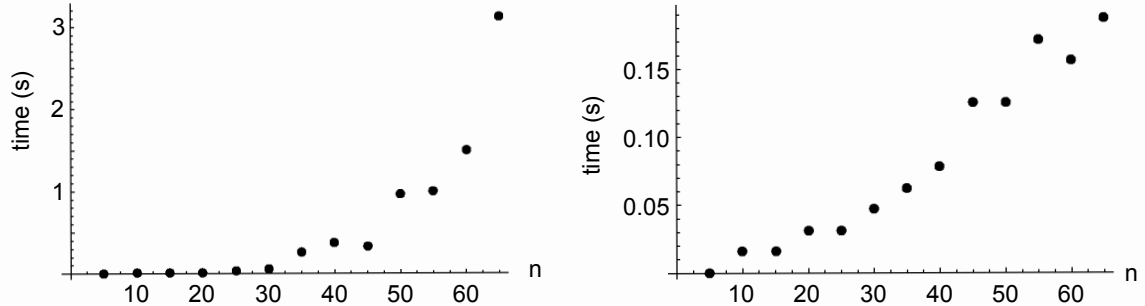


Figure 3.10 : Computing the chromatic polynomial of a generalized vertex join tree using the `ChromaticPolynomial` function (*left*) and Algorithm 1 (*right*). The `ChromaticPolynomial` function fails to run for $n > 65$.

By inspection, the growth rate in Figure 3.10, left, appears to be exponential, while the growth rate in Figure 3.10, right, appears to be polynomial; this agrees with the theoretical complexities of the two approaches. In addition, Algorithm 1 was used on graphs with up to 500 vertices, and ran relatively quickly; see Figure 3.11. The long-term growth rate of the run-time appears to be polynomial as expected.

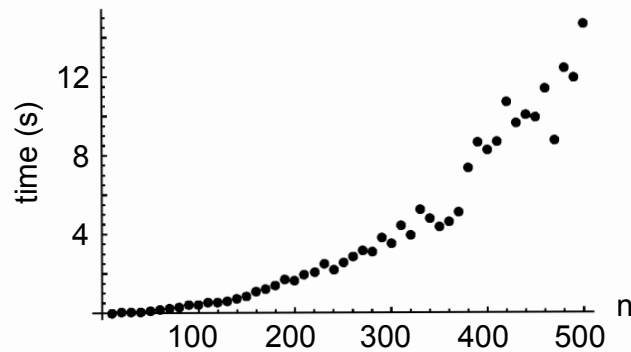


Figure 3.11 : Computing the chromatic polynomial of a generalized vertex join tree using Algorithm 1.

3.3.2 Generalized vertex join cycles

For short, the implementation of formula (3.5) will be called Algorithm 3. A test graph C_S was created by first generating a cycle C with n vertices, then adding a new vertex and connecting it to a random subset of size between $\frac{3n}{10}$ and $\frac{7n}{10}$ of the other vertices of C . The sets J_i were computed as described in (3.4) and the following discussion, and $P(C_S)$ was computed by summing the products of polynomials as described in (3.5). The order of the graph n was varied and the corresponding computation time was recorded.

The polynomials given by Algorithm 3 exactly matched the polynomials given by the `ChromaticPolynomial` function. However, the `ChromaticPolynomial` function was only able to handle graphs with $n \leq 60$ before running out of memory, and Algorithm 3 was able to handle much larger graphs. In addition, for $n \leq 60$, the computation time of Algorithm 3 was predominantly less than 0.001 seconds (the minimum time interval recorded on the system) whereas the `ChromaticPolynomial` function ran nearly 100 times slower. See Figure 3.12, left, for the run times of Algorithm 3 and Figure 3.12, right, for the run times of the `ChromaticPolynomial` function on graphs of increasing order.

Note the difference in n when comparing Figure 3.12, left, and Figure 3.12, right. By inspection, the growth rate in the former appears to be polynomial, while the growth rate in the latter appears to be exponential. Detailed analysis of the growth rates are outside the scope of this study.

3.3.3 Generalized vertex join cliques

For short, the implementation of formula (3.6) will be called Algorithm 4. A test graph K_S was created by first generating a clique K with n vertices, then adding a

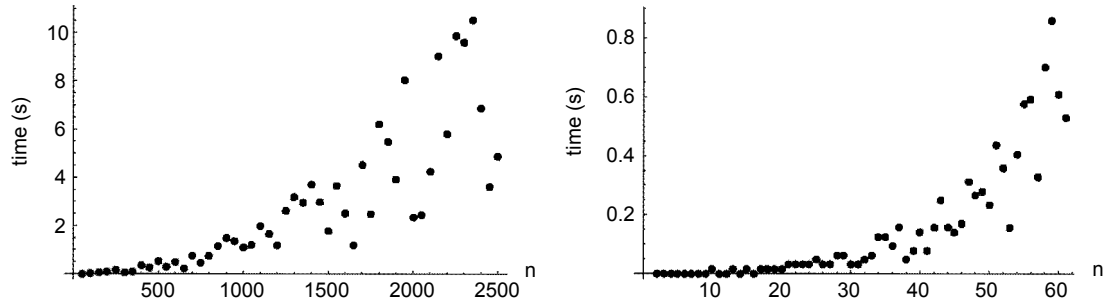


Figure 3.12 : Computing the chromatic polynomial of a generalized vertex join cycle using Algorithm 3 (*left*) and using the `ChromaticPolynomial` function of Mathematica (*right*). The `ChromaticPolynomial` function fails to run for $n > 60$.

new vertex and connecting it to a random subset of the other vertices of K . $P(K_S)$ was computed by multiplying a number of terms as described in (3.6). The order of the graph n was varied and the corresponding computation time was recorded.

The polynomials given by Algorithm 4 exactly matched the polynomials given by the `ChromaticPolynomial` function. However, the `ChromaticPolynomial` function was only able to handle graphs with $n \leq 16$ before running out of memory, and Algorithm 4 was able to handle significantly larger graphs. In addition to running out of memory for much smaller graphs, the `ChromaticPolynomial` function is more than 100 times slower, even for small graphs. See Figure 3.13, left, for the run times of Algorithm 4 and Figure 3.13, right, for the run times of the `ChromaticPolynomial` function on graphs of increasing order.

The deletion-contraction algorithm used by Mathematica performs very poorly on dense graphs, as is to be expected. A clear exponential trend in its run time is exhibited in Figure 3.13, right; moreover the function fails for generalized vertex join cliques with more than 16 vertices. On the other hand, Algorithm 4 is much faster, is able to handle graphs with thousands of vertices in less than 5 seconds, and has a

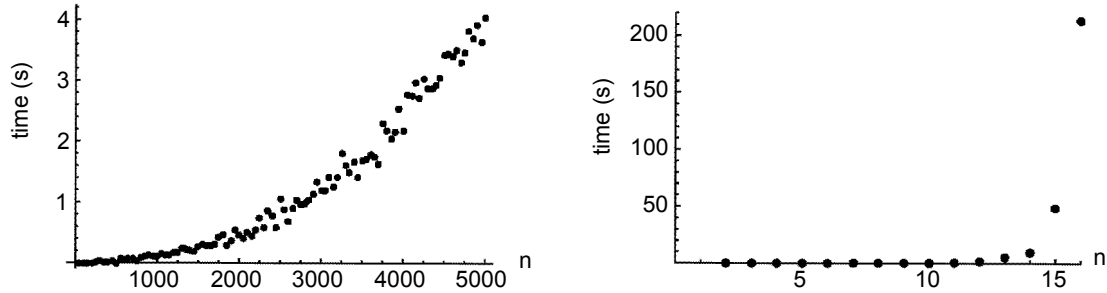


Figure 3.13 : Computing the chromatic polynomial of a generalized vertex join clique using Algorithm 4 (*left*), and using Mathematica (*right*). The program fails to run for $n \geq 16$.

polynomial growth rate as can be seen from Figure 3.13, left.

3.3.4 Discussion

All three computational experiments reveal that my algorithms were much faster than the `ChromaticPolynomial` function and able to handle much larger graphs from the appropriate family. This is not surprising, since generality is often achieved at the expense of speed. However, this motivates the inclusion of a preprocessing step in general purpose algorithms: if a general graph is suspected to contain one or more subgraphs whose chromatic polynomials can be found efficiently, it may be worth to locate those subgraphs and modify the deletion-contraction algorithm so that they appear as components in some step of the recursion. This will remove a large part of the recursion tree and may speed up the computation of the chromatic polynomial significantly.

Finally, note that simply having a closed formula for a chromatic polynomial does not mean it can be used efficiently. For instance, recall from Chapter 1 that the chromatic polynomial of any graph $G = (V, E)$ can be computed by the closed formula

$P(G; t) = \sum_{S \subseteq E} (-1)^{|S|} t^{\text{comp}(G:S)}$. However, this formula requires, among other things, a summation with an exponential number of terms. In contrast, Algorithms 1, 3, and 4 can be used in polynomial time in practice, as seen from the preceding experiments.

Chapter 4

Structural results and complexity of connected forcing

In this chapter I present a variety of structural results about connected forcing, such as the effects of certain vertex and edge operations on the connected forcing number, the relations between the connected forcing number and other graph parameters, and the computational complexity of connected forcing. I also present several technical lemmas which are used in later chapters to give efficient algorithms for computing the connected forcing numbers of certain graphs, and to characterize graphs with extremal connected forcing numbers.

4.1 Basic properties and relation to other parameters

Since any connected forcing set is also a zero forcing set, any minimum connected forcing set contains a (not necessarily minimum) zero forcing set. Thus, the following simple result follows.

Observation 4.1. *For any connected graph G , $Z_c(G) \geq Z(G)$, and this bound is sharp.*

Observation 4.1 is sharp, e.g., for paths, cycles, and complete graphs; other graphs with $Z_c(G) = Z(G)$ are identified in the following chapters. Conversely, it is natural to ask whether there are families of graphs for which the connected forcing number is arbitrarily larger than the zero forcing number; the next result answers this in the

affirmative.

Observation 4.2. *For any $c > 0$, there exists a graph G such that $Z_c(G) \geq Z(G) + c$.*

Proof. Consider the graph G_k obtained by attaching a pair of pendant vertices to each end of a path P_k . It is easy to see that $Z(G_k) = 3$ (where any 3 leaves of G_k form a minimum zero forcing set), and for $k \geq 2$, $Z_c(G_k) = k + 2$ (where all vertices except 2 leaves must be included in a minimum connected forcing set); see Figure 4.1 for an illustration. Thus, $Z_c(G_k) - Z(G_k) = k - 1$, which can be made larger than any constant c . \square



Figure 4.1 : $Z_c(G)$ and $Z(G)$ can differ arbitrarily. *Left:* minimum connected forcing set. *Right:* minimum zero forcing set.

Another natural question to ask is whether every graph has some minimum connected forcing set that contains a minimum zero forcing set (or equivalently, whether some minimum zero forcing set can be extended to obtain a minimum connected forcing set). The following observation provides a counterexample to this question.

Observation 4.3. *It is not necessarily the case that some minimum connected forcing set of a connected graph G contains a minimum zero forcing set of G .*

Proof. Consider the graph G in Figure 4.2, obtained by joining two copies of P_5 by an edge. The minimum connected forcing sets of G consist of the two degree 3 vertices, together with one degree 2 neighbor of each degree 3 vertex. The minimum zero forcing sets of G consist of two degree 1 vertices which are not in the same copy of P_5 .

Thus, no subset of any minimum connected forcing set of G is a minimum zero forcing set, and no superset of any minimum zero forcing set of G is a minimum connected forcing set. \square



Figure 4.2 : It is possible that no minimum connected forcing set of a graph contains a minimum zero forcing set. *Left*: minimum connected forcing set. *Right*: minimum zero forcing set.

Let $L(G)$ denote the number of leaves in G . A *path cover* of G is a set of vertex-disjoint induced paths in G which contain all the vertices of G . The *path cover number* of G , denoted $P(G)$, is the minimum size of a path cover. Recall that $M(G)$ denotes the maximum nullity of G . It has been shown that the zero forcing number is an upper bound on the minimum degree, maximum nullity [89], path cover number [166], and chromatic number minus one [198] of a graph. Thus, by Observation 4.1, we have the following relations.

Observation 4.4. *For any connected graph G , $Z_c(G) \geq \delta(G)$, $Z_c(G) \geq M(G)$, $Z_c(G) \geq P(G)$, and $Z_c(G) \geq \chi(G) - 1$; moreover, all these bounds are sharp.*

The bounds in Observation 4.4 are sharp, e.g., for paths. Some additional relations between connected forcing, vertex degrees, and path covers are included in the following chapters. The next result shows that the connected forcing number is also an upper bound on the number of leaves in the graph. The same relation does not always hold for the zero forcing number (see, e.g., Figures 4.1 and 4.2, right); however, it holds that $Z(G) \geq L(G)/2$, since each leaf must be either the beginning or end of a forcing chain.

Proposition 4.1. *For any connected graph G different from a path, $Z_c(G) \geq L(G)$, and this bound is sharp.*

Proof. Let R be an arbitrary connected forcing set of G . Each vertex, and in particular each leaf of G has to be in some forcing chain. A forcing chain cannot contain more than two leaves, since then it would not induce a path. Suppose some chain contains two leaves. Then one of these leaves must be in R , and every other vertex in the chain must not be in R . In particular, the neighbor of the colored leaf is not in R . However, since G is not a path, there must be other members of R outside of this forcing chain; thus, R is not connected – a contradiction. Thus, each forcing chain can contain at most one leaf. Since each forcing chain contains one element of R , it follows that $Z_c(G) \geq L(G)$. This bound is sharp, e.g., for the graph in Figure 4.2, left. \square

4.2 Vertex and edge operations

In this section, I explore the effect of various vertex and edge operations on the connected forcing number. An important concept to studying the zero forcing process is that of *zero forcing spread* of a vertex v and edge e in graph G ; these parameters, defined as $z(G; v) = Z(G) - Z(G - v)$ and $z(G; e) = Z(G) - Z(G - e)$, respectively, describe the effects of deleting a vertex or edge from the graph on the zero forcing number of the graph. It has been shown in [104, 106] that the zero forcing spread of any vertex or edge is bounded by 1; more precisely, for any graph G , vertex v , and edge e , $-1 \leq z(G; v) \leq 1$ and $-1 \leq z(G; e) \leq 1$. Similar results are also known for the *rank spread* and *path spread* – parameters which describe the change in the minimum rank and path cover number of a graph when an edge or vertex is deleted. In particular, it has been shown that $0 \leq r(G; v) \leq 2$ [199], $-1 \leq r(G; e) \leq 1$ [199],

$-1 \leq p(G; v) \leq 1$ [127, 125], and $-1 \leq p(G; e) \leq 1$ [132] (where the rank spread r and path spread p are defined analogously to the zero forcing spread).

I now define the concept of *connected forcing spread* of a vertex v and edge e in graph G as $z_c(G; v) = Z_c(G) - Z_c(G - v)$ and $z_c(G; e) = Z_c(G) - Z_c(G - e)$, respectively. In these definitions, it is convenient to restrict v to be a non-cut vertex and e to be a non-cut edge of G , since a disconnected graph cannot have a connected forcing set. The next result shows that unlike the zero forcing spread, the connected forcing spread of a vertex and edge can be arbitrarily large.

Proposition 4.2. *For any $c_1 < 0$ and $c_2 > 0$, there exist graphs G_1 and G_2 , vertices $v_1 \in G_1$ and $v_2 \in G_2$, and edges $e_1 \in G_1$ and $e_2 \in G_2$ such that $z_c(G_1; v_1) < c_1$, $z_c(G_2; v_2) > c_2$, $z_c(G_1; e_1) < c_1$ and $z_c(G_2; e_2) > c_2$.*

Proof. Let G_1 be the graph obtained by appending a pendant vertex to each endpoint of two maximally distant edges of an even cycle C_{2k} , $k \geq 5$. Let v be a vertex at distance at least 3 from any leaf of G , and let e be an edge incident to v . See Figure 4.3, left for an illustration. It is easy to see that $Z_c(G_1) = 4$ and $Z_c(G_1 - v) = Z_c(G_1 - e) = k + 4$. Thus, $z_c(G_1; v) = z_c(G_1; e) = -k$, which can be made smaller than any constant c_1 .

Let G_2 be the graph obtained by appending a copy of K_3 to each end of a path P_k , $k \geq 1$. Let v be a degree 2 vertex in one of the copies of K_3 , and let e be an edge in one of the copies of K_3 whose endpoints have degrees 2 and 3. See Figure 4.3, right for an illustration. It is easy to see that $Z_c(G_2) = k + 2$ and $Z_c(G_2 - v) = Z_c(G_2 - e) = 2$. Thus, $z_c(G_2; v) = z_c(G_2; e) = k$, which can be made larger than any constant c_2 . \square

Clearly, Proposition 4.2 also implies that adding a vertex or edge to a graph can change the connected forcing number arbitrarily. I now show that the same holds for

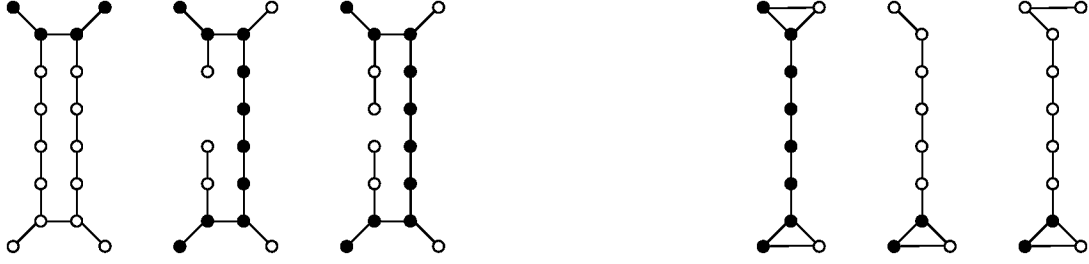


Figure 4.3 : *Left:* Deleting a vertex or edge from G_1 makes $Z_c(G_1)$ increase arbitrarily. *Right:* Deleting a vertex or edge from G_2 makes $Z_c(G_2)$ decrease arbitrarily.

edge contractions. The effect of contractions has also been considered for the zero forcing number [200] and the positive semidefinite zero forcing number [198]; both of these parameters can be increased or decreased by at most one when an edge is contracted.

Proposition 4.3. *For any $c_1 < 0$ and $c_2 > 0$, there exist graphs G_1 and G_2 and edges $e_1 \in G_1$ and $e_2 \in G_2$ such that $Z_c(G_1) - Z_c(G_1/e_1) < c_1$ and $Z_c(G_2) - Z_c(G_2/e_2) > c_2$.*

Proof. Let e_1 and f be two maximally distant edges of an even cycle C_{2k} , $k \geq 3$; let G_1 be the graph obtained by appending a pendant vertex to each endpoint of e_1 , and two pendant vertices to each endpoint of f . See Figure 4.4, left for an illustration. It is easy to see that $Z_c(G_1) = 6$ and $Z_c(G_1/e_1) = k + 5$. Thus, $Z_c(G_1) - Z_c(G_1/e_1) = -k + 1$, which can be made smaller than any constant c_1 .

Let G_2 be the graph obtained by attaching a pair of pendant vertices to each end of a path P_k . Let e_2 be an edge incident to some leaf of G_2 . See Figure 4.4, right for an illustration. It is easy to see that $Z_c(G_2) = k + 2$ and $Z_c(G_2/e_2) = 3$. Thus, $Z_c(G_2) - Z_c(G_2/e_2) = k - 1$, which can be made larger than any constant c_2 . \square

Lastly, I consider the operation of edge subdivision. I show in the next two results that subdividing an edge can increase the connected forcing number arbitrarily, or

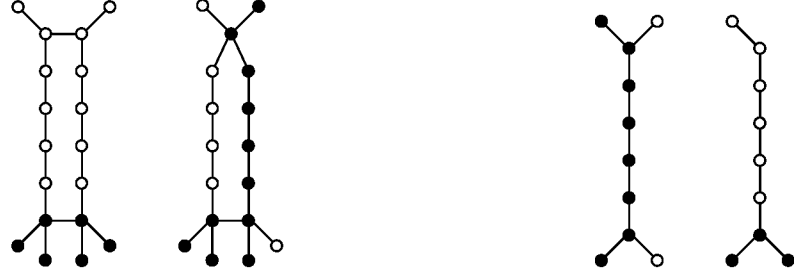


Figure 4.4 : *Left:* Contracting an edge in G_1 makes $Z_c(G_1)$ increase arbitrarily. *Right:* Contracting an edge in G_2 makes $Z_c(G_2)$ decrease arbitrarily.

keep it the same, but cannot decrease it. The effects of edge subdivision have also been investigated for other parameters like the zero forcing number [200] and the positive semidefinite zero forcing number [198]; when an edge is subdivided, the former can increase by one or remain unchanged, and the latter always remains unchanged.

Proposition 4.4. *Let G be a connected graph, e be an edge in G , and H be the graph obtained by subdividing e in G . Then $Z_c(H) \geq Z_c(G)$, and this bound is sharp.*

Proof. Let $e = uv$ be the subdivided edge, and w be the new vertex introduced by the subdivision, i.e., if $G = (V, E)$, then $H = (V \cup \{w\}, E \setminus \{uv\} \cup \{uw, vw\})$. Let R be a minimum connected forcing set of H and fix some chronological list of forces associated with R . Suppose first that R contains both u and v . Regardless of whether or not R contains w , R (or $R \setminus \{w\}$) remains connected in G . Moreover, any force that is valid in H remains valid in G , since u and v are already colored and have no uncolored neighbors in G which they do not have in H (and all other vertices have the same neighbors as in H). Thus, R (or $R \setminus \{w\}$) is also a connected forcing set of G .

Next suppose that R contains neither u nor v ; in this case R cannot contain w , since if it did, it would not be connected. Then, one of u and v , say u , will be forced

first by a vertex other than w in some timestep, and then either u will force w which can force v , or v will be forced by some other vertex in a later timestep. In either case, these forces remain valid in G (where in the former case, the forcing sequence $u \rightarrow w \rightarrow v$ is replaced by $u \rightarrow v$). Thus, R is also a connected forcing set of G .

Finally, suppose R contains only one of u and v , say u . If R also contains w , then w can force v in the first timestep in H . Then, by the same reasoning as in the first case above, the set $R \setminus \{w\} \cup \{v\}$ is a connected forcing set of G , and has the same size as R . If R does not contain w , then in some timestep either u will force w which can force v , or v will be forced by some other vertex. Then, by the same reasoning as in the second case above, R is also a connected forcing set of G . Thus, in all cases, $Z_c(H) \geq Z_c(G)$; moreover, this bound is attained with equality, e.g., by subdividing any edge in a path. \square

Proposition 4.5. *For any $c > 0$, there exists a graph G and edge $e \in G$ such that if H is obtained by subdividing e in G , then $Z_c(H) - Z_c(G) > c$.*

Proof. Let e and f be two maximally distant edges of an even cycle C_{2k} , $k \geq 3$; let G be the graph obtained by appending a pendant vertex to each endpoint of e , and two pendant vertices to each endpoint of f . Let H be the graph obtained by subdividing e in G ; see Figure 4.5 for an illustration. It is easy to see that $Z_c(G) = 6$ and $Z_c(H) = k + 5$. Thus, $Z_c(H) - Z_c(G) = k - 1$, which can be made larger than any constant c . \square

I will now show that a leaf incident to a vertex of degree 2 does not belong to any minimum connected forcing set. In view of Proposition 4.4, this result implies that subdividing an edge incident to a leaf does not affect the connected forcing number.

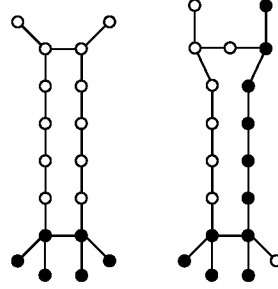


Figure 4.5 : Subdividing an edge in G makes $Z_c(G)$ increase arbitrarily.

Lemma 4.1. *Let $G = (V, E)$ be a connected graph different from a path and ℓ be a leaf adjacent to a vertex of degree 2. Then ℓ does not belong to any minimum connected forcing set of G , and $Z_c(G) = Z_c(G - \ell)$.*

Proof. Let R be an arbitrary minimum connected forcing set of G . Let v be the neighbor of ℓ , and w be the other neighbor of v . We will first show that $\ell \notin R$. Suppose on the contrary that $\ell \in R$, $v \in R$, and $w \in R$; then $R \setminus \{\ell\}$ is a connected forcing set of smaller size, since v can force ℓ in the first timestep. Next, suppose that $\ell \in R$, $v \in R$, and $w \notin R$. If ℓ and v are the only vertices in R , then G must be a path, contradicting our assumption; if there are other vertices in R besides ℓ and v , then R is not connected. Similarly, it cannot be the case that $\ell \in R$, $v \notin R$, and $w \notin R$, or that $\ell \in R$, $v \notin R$, and $w \in R$. Thus, ℓ is not in R .

Now, if R is a minimum connected forcing set of G , then R is also a minimum connected forcing set of $G - \ell$, since ℓ cannot be initially colored and cannot be a non-terminal vertex of any forcing chain. Conversely, if R is a minimum connected forcing set of $G - \ell$ and $v \in R$, then w must also be in R since otherwise R will be disconnected; then, v can force ℓ in G in the first timestep. If $v \notin R$, then at some timestep w will force v , and the addition of ℓ can have no effect on the forcing chain

of w or any other vertex in G ; at the next timestep, v can force ℓ . Thus, in both cases, R is also a minimum connected forcing set of G , so $Z_c(G) = Z_c(G - \ell)$. \square

4.3 Bounds and mandatory vertices

This section identifies sets of vertices which are contained in every minimum connected forcing set, and uses them to establish bounds on the connected forcing number. These results are in some contrast to a result of Barioli [128] that no vertex belongs to every minimum zero forcing set. An asymptotic bound on the connected forcing number in terms of graph density is also presented. I begin with the following technical lemma, which identifies certain vertices that must belong to every connected forcing set, using the vertex connectivity of the graph.

Lemma 4.2. *Let G be a connected graph, S be a separating vertex set of G , and V_1, \dots, V_k be the vertex sets of the connected components of $G - S$. If each vertex of S is incident to each V_i , $1 \leq i \leq k$, then every connected forcing set of G contains a vertex from at least $k - 1$ of V_1, \dots, V_k . Moreover, if $k = 2$ and $Z(G[V_i]) > |S|$ for $i \in \{1, 2\}$, or if $k \geq 3$, then every connected forcing set of G contains a vertex of S .*

Proof. Let R be an arbitrary connected forcing set of G and \mathcal{F} be an arbitrary chronological list of forces associated with R . Suppose for contradiction that R does not contain vertices from two components of $G - S$, say $G[V_1]$ and $G[V_2]$. Let v be the first vertex in $V_1 \cup V_2$ to be forced; since $N(v; G) \subset S \cup V_1 \cup V_2$, v must be forced by a vertex of S . However, no vertex of S can force v , since at that timestep every vertex of S has at least two uncolored neighbors — one in V_1 and one in V_2 . Thus, R must contain a vertex from at least $k - 1$ of V_1, \dots, V_k . In particular, if $k \geq 3$, R must contain a vertex from at least two components of $G - S$. Since R is connected,

and since any path between two vertices from different components of $G - S$ must contain a vertex of S , R contains a vertex of S .

Now suppose $k = 2$, and suppose for contradiction that $R \subset V_1$. Let Z be the set of vertices of V_2 which are forced (according to \mathcal{F}) by vertices in S ; let \mathcal{F}' be a list of forces whose i^{th} element is the i^{th} instance of a vertex of V_2 forcing another vertex of V_2 (according to \mathcal{F}). We claim that Z is a zero forcing set of $G[V_2]$, and that \mathcal{F}' is a chronological list of forces for Z in $G[V_2]$. To see why, note that if $v \in V_2$ is the i^{th} vertex to force another vertex in V_2 at some timestep of \mathcal{F} , since $N(v; G[V_2]) \subset N(v; G)$, v and all-but-one of its neighbors are colored in $G[V_2]$ at the i^{th} step of \mathcal{F}' . Thus, v would be able to force the same vertex in $G[V_2]$ as in G , so each force between two vertices of V_2 in G can also be performed in $G[V_2]$. Since in G , each vertex in V_2 is forced either by a vertex of S or a vertex of V_2 , in $G[V_2]$ each vertex is either in Z or is in a forcing chain initiated by a vertex in Z ; thus Z is a forcing set of $G[V_2]$. However, $|Z| \leq |S|$ since each vertex in S forces at most one vertex of V_2 in G ; this contradicts the assumption that $Z(G[V_2]) > |S|$. Thus, $R \not\subset V_1$; similarly, $R \not\subset V_2$. Also, R cannot contain elements of both V_1 and V_2 without containing elements of S , since R is connected and S is a separating set. Thus, R contains a vertex of S . \square

Below is some terminology and notation which will be used in the sequel.

Definition 4.1. Let $G = (V, E) \not\cong P_n$ be a graph and v be a vertex of degree at least 3. A *pendant path attached to v* is a maximal set $P \subset V$ such that $G[P]$ is a connected component of $G - v$ which is a path, one of whose ends is adjacent to v in G . The neighbor of v in P will be called the *base* of the path, and $p(v)$ will denote the number of pendant paths attached to $v \in V$. We will also say that $p(u) = 1$ if u

is a cut vertex which belongs to a pendant path.

Definition 4.2. Let $G = (V, E) \not\cong P_n$ be a connected graph. Define

$$\begin{aligned} R_1(G) &= \{v \in V : \text{comp}(G - v) = 2, p(v) = 1\} \\ R_2(G) &= \{v \in V : \text{comp}(G - v) = 2, p(v) = 0\} \\ R_3(G) &= \{v \in V : \text{comp}(G - v) \geq 3\} \\ \mathcal{L}(G) &= \bigcup_{v \in V, d(v) \geq 3} \{\text{all-but-one bases of pendant paths attached to } v\} \\ \mathcal{M}(G) &= R_2(G) \cup R_3(G) \cup \mathcal{L}(G). \end{aligned}$$

When there is no scope for confusion, the dependence on G will be omitted. Note that the sets R_1 , R_2 , and R_3 partition the set of cut vertices of G .

The next result is a useful implication of Lemma 4.2 and the sets defined above. In view of this result, $\mathcal{M}(G)$ can be understood as a set of “mandatory vertices”, which appear in every connected forcing set.

Lemma 4.3. *Let $G = (V, E)$ be a connected graph different from a path and R be an arbitrary connected forcing set of G . Then $\mathcal{M}(G) \subset R$.*

Proof. Since a cut vertex v is a separating set incident to each component of $G - v$, by Lemma 4.2, R must contain v for all $v \in R_2 \cup R_3$. If all components of $G - v$ are pendant paths attached to v , then it is easy to verify that R consists of v and all-but-one bases of pendant paths attached to v , i.e. $\mathcal{L} \subset R$. Now, suppose v is a cut vertex such that not all components of $G - v$ are pendant paths attached to v . If R does not include any vertices from some component of $G - v$, that component must be a pendant path attached to v , since otherwise the component cannot be forced by v alone. Since by Lemma 4.2, R includes a vertex from all or all-but-one components

of $G - v$, and since the excluded component can only be a pendant path attached to v , it follows that for each $u \in V$, R includes at least all-but-one bases of pendant paths attached to u . Thus, by definition, $\mathcal{L} \subset R$, and since $\mathcal{M} = R_2 \cup R_3 \cup \mathcal{L}$, it follows that $\mathcal{M} \subset R$. \square

Note that the converse of Lemma 4.3 is not true, in the sense that there could be vertices not in $\mathcal{M}(G)$ which are contained in every minimum connected forcing set of a graph G . For example, let u and w be the neighbors of a vertex v in C_{10} , and let G be the graph obtained by appending leaves to u , v , and w . Then v is contained in every minimum connected forcing set of G , even though v is in R_1 and not \mathcal{M} . The next result is also related to vertices which belong to every connected forcing set.

Proposition 4.6. *Let G be a connected graph different from a path and B be a block of G which is not a cut edge of a pendant path of G . Then every connected forcing set of G contains at least $\delta(G[B])$ vertices of B .*

Proof. Suppose there is a connected forcing set S of G which contains at most $\delta(G[B]) - 1$ vertices of B . Clearly there are uncolored vertices in B , since B has at least $\delta(G[B]) + 1$ vertices. Any forcing chain initiated by a vertex outside B and containing a vertex of B must pass through an uncolored cut vertex p of B ; by Lemma 4.3, $p \notin \mathcal{M}$, so $p \in R_1$. However, this means S contains a vertex of a pendant path, but not the vertex to which the path is attached — this contradicts S being connected. Thus, there can be no forcing chain initiated outside B and passing through B .

Now suppose there is a (non-singleton) forcing chain starting at $v \in B$ which contains another vertex of B . The vertex v has at least $\delta(G[B])$ neighbors in B ; however, by assumption, at most $\delta(G[B]) - 2$ of them can be colored. Since none

of these neighbors of v can get forced by a vertex outside of B , v cannot force any vertex — a contradiction. Thus, no uncolored vertex in B can be forced, so S must contain at least $\delta(G[B])$ vertices of B . \square

The next result considers the relationship between the density of a graph and its connected forcing number. It can be readily verified that sparse graphs can have both large and small zero forcing numbers and connected forcing numbers; path graphs and star graphs are extremal in this regard. The following theorem shows that in contrast, dense graphs can only have asymptotically large connected forcing numbers and zero forcing numbers.

Theorem 4.1. *Let $G = (V, E)$ be a graph with $|E| = \Omega(|V|^2)$. Then, $Z_c(G) = \Theta(|V|)$.*

Proof. Let $n = |V|$ and suppose for contradiction that for every $S \subset V$, $\delta(G[S]) = o(n)$. Let $G_0 = G$; for $1 \leq i \leq n$, let v_i be a vertex such that $d(v_i; G_{i-1}) = \delta(G_{i-1})$ and let $G_i = G_{i-1} - v_i$. In words, the graphs $\{G_i\}_{i=1}^n$ are obtained by repeatedly deleting a vertex of minimum degree. By our assumption, for $1 \leq i \leq n$, $\delta(G_i) = o(n)$ so each G_i has $o(n)$ fewer edges than G_{i-1} . However, this is a contradiction, since $n \cdot o(n) \neq \Omega(n^2)$. Thus, there must be some $S \subset V$ for which $\delta(G[S]) = \Omega(n)$.

Let R be a minimum connected forcing set of G ; clearly $|R| = O(n)$. Fix some chronological list of forces, and let v be the first vertex in S (if any) which forces another vertex. At the timestep when v performs a force, v and all-but-one of its neighbors must be colored; since $\delta(G[S]) = \Omega(n)$, there must be $\Omega(n)$ colored vertices at the timestep when v performs a force. Since v is the first vertex in S to perform a force, each of v 's neighbors in S is either in R , or has been forced by a distinct forcing chain (since if two vertices in S are in the same forcing chain, the one that comes first in the chain would have performed a force before v). If no vertex of S

ever performs a force, then again each vertex in S is either in R , or has been forced by a distinct forcing chain. Since each forcing chain is initiated by a unique element in R , and since $|R| \geq |S| \geq \delta(G[S]) = \Omega(n)$, it follows that $Z_c(G) = \Theta(n)$. \square

A similar argument as above can be used to show that for $G = (V, E)$ with $|E| = \Omega(|V|^2)$, then $Z(G) = \Theta(|V|)$, as well. It should be noted that Theorem 4.1 describes only the asymptotic relationship between the density of a graph and its connected forcing number. It is also useful to obtain non-asymptotic bounds on the connected forcing number in terms of the edge count and other easily computable parameters. Some progress to this end has been made in [201]; bounds on the zero forcing number have also been pursued, e.g., in [107, 108, 202]. The next result is also a step in this direction. Using Proposition 4.6 and the fact that the only vertices which can belong to more than one block which is not part of a pendant path are the vertices in $R_2 \cup R_3$, I formulate the following lower bound on the connected forcing number.

Corollary 4.1. *Let G be a connected graph, \mathcal{B} be the set of blocks of G which are not cut edges of pendant paths of G , and let $\mu(v)$ denote the number of blocks a vertex v is part of. Then,*

$$Z_c(G) \geq \sum_{B \in \mathcal{B}} \delta(G[B]) - \sum_{v \in R_2 \cup R_3} (|\mu(v)| - 1).$$

The bound in Corollary 4.1 is tight, for example in a cycle or complete graph; this bound can be used in conjunction with the bound $Z_c(G) \geq |\mathcal{M}(G)|$ implied by Lemma 4.3.

4.4 Computational complexity

In this section, I show that computing the connected forcing number of a graph is NP-complete. To begin, I state the decision version of this problem.

PROBLEM: Connected zero forcing (*CZF*)

INSTANCE: A simple undirected connected graph $G = (V, E)$ and a positive integer $k \leq |V|$.

QUESTION: Does G contain a zero forcing set S of size at most k such that $G[S]$ is connected?

Theorem 4.2. *CZF is NP-complete.*

Proof. We will first show that *CZF* is in NP. Given a set S of vertices of G , it can be checked in polynomial time whether there is a vertex in S with exactly one neighbor not in S . Moreover, there cannot be more than $|V|$ steps in a forcing process. Thus, a nondeterministic algorithm can check in polynomial time whether a subset of vertices of V is forcing, whether it induces a connected subgraph, and whether it has size at most k . Thus, *CZF* is in NP.

For our reduction, we select the problem of zero forcing, which was proved to be NP-complete in [203, 103]. The decision version of zero forcing is stated below.

PROBLEM: Zero forcing (*ZF*)

INSTANCE: A simple undirected graph $G = (V, E)$ and a positive integer $k \leq |V|$.

QUESTION: Does G contain a zero forcing set S of size at most k ?

Next, we construct a transformation f from *ZF* to *CZF*. Let $I = \langle G, k \rangle$ be an instance of *ZF*, where $G = (V, E)$ and $V = \{v_1, \dots, v_n\}$. We define $f(I) = \langle G', k+2 \rangle$, where $G' = (V \cup \{v^*, \ell_1, \ell_2\}, E \cup \{\{v^*, v_i\} : 1 \leq i \leq n\} \cup \{\{v^*, \ell_1\}, \{v^*, \ell_2\}\})$. See

Figure 4.6 for an illustration of G and G' .

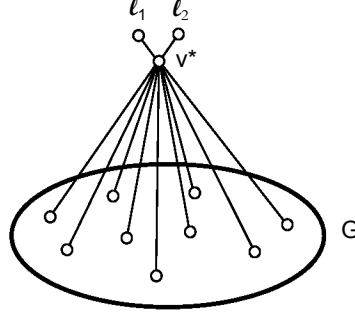


Figure 4.6 : Obtaining G' from G .

Finally, we will prove the polynomiality and correctness of f . Clearly, G' can be constructed from G in polynomial time, so f is a polynomial transformation.

Suppose $I = \langle G, k \rangle$ is a ‘yes’ instance of ZF , i.e., that $G = (V, E)$ has a zero forcing set S of size at most k . We claim that $S' := S \cup \{v^*, \ell_1\}$ is a connected forcing set of G' . To see why, first note that since v^* is adjacent to every vertex in $S' - \{v^*\}$, $G'[S']$ is connected. Next, given an arbitrary chronological list of forces for S in G , each force can also be applied for S' in G' , since for any $v \in V$, $N(v; G') = N(v; G) \cup \{v^*\}$ and v^* is initially colored; thus, when v has a single uncolored neighbor in G at some step of the forcing process, it will have the same uncolored neighbor in G' . When all vertices of V in G' are colored, ℓ_2 will be the only uncolored vertex in G' , and it will be forced by v^* . Thus S' is a connected forcing set of G' of size at most $k + 2$, so $f(I) = \langle G', k + 2 \rangle$ is a ‘yes’ instance of CZF .

Conversely, suppose $f(I) = \langle G', k + 2 \rangle$ is a ‘yes’ instance of CZF , i.e., that G' has a connected forcing set S' of size at most $k + 2$. Fix a chronological list of forces for S' in G' and suppose v^* forces a vertex $w \in V$. Then, both ℓ_1 and ℓ_2 must be in S' , since they are adjacent only to v^* , which cannot force them if it forces w . Moreover, w

must be the last uncolored vertex in G' , since if there was another uncolored vertex, v^* would have more than one uncolored neighbor and could not force w . If w is not an isolated vertex of G , then in the last step of the forcing process, w can be forced by one of its neighbors in V instead of by v^* . If w is an isolated vertex of G , then it is a leaf of G' , and the set $S'' = S' \setminus \{\ell_1\} \cup \{w\}$ is also a connected forcing set of G' , where v^* does not force any vertex of V (if we use the same chronological list of forces, except in the last step, $v^* \rightarrow \ell_1$ instead of $v^* \rightarrow w$).

Thus, we can choose a connected forcing set S' and a chronological list of forces for S' such that v^* does not force any vertex of V in G' . We claim that $S := S' \cap V$ is a forcing set of G . To see why, first note that v^* must be in S' by Lemma 4.3, and that for any $v \in V$, $N(v; G) = N(v; G') \setminus \{v^*\}$. Thus, each force between vertices of V in G' can also be applied for S in G , since if $v \in V$ has a single uncolored neighbor in G' at some step of the forcing process, it will have the same uncolored neighbor in G . Moreover, since v^* does not force any vertex in V , all vertices in V must be forced by the elements of S' which are in V . Thus, S is a forcing set of G . Finally, to verify the size of S , note that by Lemma 4.3, v^* and at least one of ℓ_1 and ℓ_2 must be in S' , so $k + 2 \geq |S'| \geq |S' \cap V| + 2 = |S| + 2$, so S has size at most k . Thus, if $f(I)$ is a ‘yes’ instance of CZF , then I is a ‘yes’ instance of ZF . \square

Chapter 5

Characterizations of connected forcing numbers

In view of Theorem 4.2, one cannot hope to efficiently compute the connected forcing number of a general graph. In this chapter, I give specialized algorithms for efficiently computing the connected forcing numbers of different families of graphs, including tree-like graphs and graphs with certain symmetries. I also characterize the graphs from these families whose connected forcing numbers are equal to their zero forcing numbers. Related parameters of such families of graphs have been investigated in the past: for example, [204] and [205] give efficient algorithms for computing the path cover numbers of trees; [206] and [89] respectively show that for trees, the path cover number equals the maximum nullity and the zero forcing number; [89] and [106] show that in block graphs, the maximum nullity equals the zero forcing number; [98] surveys several characterizations and gives polynomial time algorithms for the maximum nullity and path cover number of unicyclic graphs; [207] and [198] show that the zero forcing number of unicyclic graphs and cactus graphs equals the path cover number; [137] characterizes the power domination number of block graphs.

5.1 Graphs with symmetry

By definition, for any connected graph G , $Z_c(G) = Z(G)$ if and only if there exists some minimum zero forcing set of G which is connected. Thus, if $Z(G)$ is known, and a minimum zero forcing set of G is found which is connected, then $Z_c(G)$ will

immediately be determined as well. However, in general, it is harder to characterize or even count all distinct minimum zero forcing sets of a graph than to find its connected forcing number. Thus, even if $Z(G)$ is known, it may be difficult to determine whether or not any of the minimum zero forcing sets of the graph are connected. In particular, in Chapter 7, I show that a graph could have exponentially-many minimum zero forcing sets which are disconnected, and still have minimum zero forcing sets which are connected. Nevertheless, sometimes it is possible to easily find a minimum zero forcing set which is connected. This principle is applied below to easily characterize the connected forcing numbers of some graphs.

Proposition 5.1. *Let W_n be the wheel on n vertices, Q_s be the s^{th} hypercube on 2^s vertices, T_s be the s^{th} supertriangle on $\frac{1}{2}s(s+1)$ vertices, M_n be the Möbius ladder on $n = 2k$ vertices. Then,*

1. $Z_c(W_n) = 3$

2. $Z_c(M_n) = 4$

3. $Z_c(K_{a_1, \dots, a_k}) = n - 2$, where $k \geq 2$ and $a_i \geq 2$ for $1 \leq i \leq k$

4. $Z_c(G \circ K_2) = \frac{2}{3}n$, where G is a connected graph

5. $Z_c(Q_s) = 2^{s-1}$

6. $Z_c(T_s) = s$

7. $Z_c(C_a \square C_b) = \begin{cases} 2a - 1 & \text{if } m = n \text{ and } n \text{ is odd} \\ 2a & \text{otherwise} \end{cases}, \text{ where } b \geq a \geq 3$

8. $Z_c(K_a \times K_b) = n - 4$, where $a, b \geq 4$

9. $Z_c(P_a \boxtimes P_b) = a + b - 1$

10. $Z_c(G \square P_a) = |V|$, where $G = (V, E)$ is a connected graph and $|V| \leq a$

Proof. The numbers of the proofs below correspond to the numbers in the statement of the proposition.

1. The vertex of degree $n - 1$ of the wheel together with two other adjacent vertices form a connected forcing set; also, $Z_c(W_n) \geq \delta(W_n) = 3$. Thus, $Z_c(W_n) = 3$.
2. In [89], it was shown that $Z(M_n) = 4$, where the vertex set of a subgraph of M_n isomorphic to C_4 forms a zero forcing set; since this set is connected, it follows that $Z_c(M_n) = 4$.
3. Let R be a minimum connected forcing set of K_{a_1, \dots, a_k} . If R excludes two vertices in the same part of K_{a_1, \dots, a_k} , then every vertex outside that part will be adjacent to two uncolored vertices and will not be able to force them. If R excludes a vertex from more than two parts of K_{a_1, \dots, a_k} , then every vertex will have at least two uncolored neighbors and will not be able to force them. Thus, R can exclude at most two vertices of K_{a_1, \dots, a_k} , so $Z_c(K_{a_1, \dots, a_k}) \geq n - 2$. On the other hand, the set excluding one vertex from each of two parts of K_{a_1, \dots, a_k} is connected and forcing, so $Z_c(K_{a_1, \dots, a_k}) = n - 2$.
4. If $G \simeq K_1$, then $G \circ K_2 \simeq K_3$ and $Z_c(G \circ K_2) = 2$. Now suppose G has at least two vertices. In $G \circ K_2$, every vertex of G is a cut vertex and belongs to $R_2(G \circ K_2)$ or $R_3(G \circ K_2)$. Thus, by Lemma 4.3, every vertex of G is in every minimum connected forcing set of $G \circ K_2$. Moreover, by Proposition 4.6, every minimum connected forcing set of $G \circ K_2$ contains at least two vertices of each block of $G \circ K_2$ consisting of a copy of K_2 and a vertex of G ; thus,

- $Z_c(G \circ K_2) \geq \frac{2}{3}n$. The set consisting of the vertices of G and one vertex from each copy of K_2 in the corona is a connected forcing set, since the colored vertex in each copy of K_2 can force its uncolored neighbor. Thus $Z_c(G \circ K_2) = \frac{2}{3}n$.
5. In [89], it was shown that $Z(T_s) = s$, where the vertices along one edge of T_s (in a standard planar embedding) form a minimum zero forcing set; since this set is connected, it follows that $Z_c(T_s) = s$.
 6. In [89], it was shown that $Z(Q_s) = 2^{s-1}$, where the vertices of a subgraph of Q_s isomorphic to Q_{s-1} form a zero forcing set; since this set is connected, it follows that $Z_c(Q_s) = 2^{s-1}$.
 7. In [109] it was shown that $Z(C_a \square C_b) = 2a - 1$ if $m = n$ and n is odd, and $Z(C_a \square C_b) = 2a$ otherwise. In the former case, a subgraph of $C_a \square C_b$ isomorphic to $C_a \square P_2 - v$ (where v is any vertex) is a zero forcing set which is connected. Likewise, in the latter case, a subgraph of $C_a \square C_b$ isomorphic to $C_a \square P_2$ is a zero forcing set which is connected. Thus, $Z_c(C_a \square C_b)$ is as claimed.
 8. In [106], it was shown that $Z(K_a \times K_b) = n - 4$, and that if $V(K_a \times K_b) = \{(x_i, y_b) : 1 \leq i \leq a, 1 \leq j \leq b\}$, then $R = V(K_a \times K_b) \setminus \{(x_2, y_2), (x_3, y_1), (x_1, y_2), (x_2, y_1)\}$ is a minimum zero forcing set. Since $a, b \geq 4$, the set $\{(x_i, y_j) : i \geq 3 \text{ or } j \geq 4\}$ is connected in $K_a \times K_b$; moreover, $(x_1, y_1) \sim (x_3, y_2)$ and $(x_3, y_2) \sim (x_2, y_3)$. Thus, R is connected, so $Z_c(K_a \times K_b) = n - 4$.
 9. In [89], it was shown that $Z(P_a \boxtimes P_b) = a + b - 1$, where the vertices along two adjacent edges of $P_a \boxtimes P_b$ (in a grid embedding) form a minimum zero forcing set; since this set is connected, it follows that $Z_c(P_a \boxtimes P_b) = a + b - 1$.
 10. In [89], it was shown that $Z(G \square P_a) = |V|$; since the vertices of a copy of G

corresponding to an end of the path P_a form a zero forcing set, and since G is connected, it follows that $Z_c(G \square P_a) = |V|$. \square

Using Proposition 4.1, I now give a closed formula for the connected forcing numbers of sunlet graphs. The *sunlet graph* on $n = 2k$ vertices, $k \geq 3$, is obtained by adding a pendant vertex to every vertex of a cycle C_k . The path cover number, minimum rank, and zero forcing number of the sunlet graph have been investigated in [127, 125]; in particular, it has been shown that the zero forcing number of the sunlet graph on $n = 2k$ vertices is $\lceil k/2 \rceil$.

Proposition 5.2. *Let $G = (V, E)$ be the sunlet graph on $n = 2k$ vertices. Then, $Z_c(G) = k$.*

Proof. By Proposition 4.1, $Z_c(G) \geq L(G) = k$. On the other hand, $R = \{v \in V : d(v) = 3\}$ is a forcing set, since each $v \in R$ can force the leaf vertex adjacent to it in the first step of the forcing process. Moreover, $G[R]$ is connected, so $Z_c(G) \leq |R| = k$. Thus, we conclude $Z_c(G) = k$. \square

The *sun graph* on $n = 2k$ vertices, $k \geq 3$, is obtained by joining a vertex to the endpoints of each edge in a spanning cycle subgraph C_k of a clique K_k ; see Figure 5.1 for an illustration. The next result gives a closed formula for the connected forcing number of sun graphs.

Theorem 5.1. *Let $G = (V, E)$ be the sun graph on $n = 2k$ vertices. Then,*

$$Z_c(G) = \begin{cases} 3 & \text{if } k = 3 \\ k - 1 & \text{if } k \geq 4 \end{cases}$$

Proof. By inspection, the connected forcing number of the sun graph on 6 vertices is 3. Thus, assume G is a sun graph on $n = 2k$ vertices with $k \geq 4$. We will first construct a connected forcing set of G of size $k - 1$.

Let K be the set of vertices of G with degree greater than 2; pick $v \in K$ and let a, b be its two neighbors of degree 2, and u, w be the other neighbors of a and b besides v . Finally, let x be another vertex in K different from u, v , and w (which exists since $|K| = k \geq 4$ by assumption). We claim the set $R = K \setminus \{u, w, x\} \cup \{a, b\}$ is a connected forcing set of G . First, R is connected, since v is adjacent to every vertex in R . Next, note that a and b can force u and w in the first timestep, since they each only have one uncolored neighbor. At that step, every vertex in K except x will be colored; moreover, every neighbor of v except x will be colored, so v will force x . After v forces x , every vertex in K will be colored, and u and w will each have one uncolored neighbor of degree 2 which they will force. At that point, the other neighbors of the most recently colored degree 2 vertices will each have one uncolored neighbor of degree 2. By continuing this process, all degree 2 vertices of G will eventually become colored. Thus, R is a connected forcing set, so $Z_c(G) \leq |R| = k - 1$; see Figure 5.1 for an illustration of this construction.

On the other hand, by Observation 4.4, $Z_c(G) \geq \chi(G) - 1$; since the chromatic number is an upper bound to the clique number, and since the clique number of G is k , we have that $k - 1 \leq \omega(G) - 1 \leq \chi(G) - 1 \leq Z(G) \leq Z_c(G)$. Thus, we conclude that $Z_c(G) = k - 1$. \square

From the last chain of inequalities in the proof above and by Observation 4.1 (and by inspection for the case of $k = 3$), we conclude that the formula in Theorem 5.1 can also be used to compute the zero forcing number of G . This is stated formally below.

Corollary 5.1. *Let G be a sun graph on $n = 2k$ vertices. Then, $Z(G) = Z_c(G)$.*

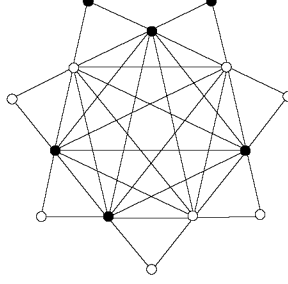


Figure 5.1 : A connected forcing set of the sun graph on 14 vertices.

5.2 Trees and uniclque graphs

In this section, I characterize the connected forcing numbers of trees and uniclque graphs. The proofs of these results are constructive, and can be used to find minimum connected forcing sets in linear time.

Theorem 5.2. *Let $T = (V, E)$ be a tree. Then,*

$$Z_c(T) = \begin{cases} 1 & \text{if } T \simeq P_n \\ |\mathcal{M}(T)| & \text{if } T \not\simeq P_n. \end{cases}$$

Moreover, a minimum connected forcing set of T can be found in $O(n)$ time.

Proof. If T is a path, then $Z_c(T) = 1$ and an endpoint of the path is a minimum connected forcing set. Assume henceforth that T is not a path; we will show that \mathcal{M} is a minimum connected forcing set of T . All vertices of T with degree at least 3 are in R_3 , and all vertices of T which have degree 2 and do not belong to pendant paths are in R_2 . Thus, any vertex of T which is not initially colored belongs to some pendant

path, and all other vertices of that pendant path are also initially uncolored, possibly except the base. Since deleting all vertices of a pendant path (possibly except the base) does not disconnect the graph, \mathcal{M} is a connected set. Next, let v be a vertex to which a pendant path is attached. Since v is in R_3 and therefore in \mathcal{M} , and since the base of one of the pendant paths attached to v is the only uncolored neighbor of v , v will be able to force this pendant path; all other pendant paths attached to v can be forced by their bases. Thus, each pendant path of T will be forced, so \mathcal{M} is a connected forcing set of T . Moreover, by Lemma 4.3, every minimum connected forcing set of T contains \mathcal{M} , so $Z_c(T) = |\mathcal{M}|$.

Clearly, it is possible to find the set of pendant paths of T in linear time (e.g. by starting from the degree 1 vertices of T and applying depth-first-search until a vertex of degree at least 3 is reached); then, for each vertex to which one or more pendant paths are attached, all-but-one neighbors which are bases of pendant paths can be selected in linear time, and added to the set of vertices which do not belong to pendant paths to form \mathcal{M} . Thus, \mathcal{M} can be found in $O(n)$ time. \square

The next result is a characterization of trees for which $Z_c(T) = Z(T)$.

Proposition 5.3. *Let T be a tree. Then, $Z_c(T) = Z(T)$ if and only if $T \simeq P_n$.*

Proof. If $T \simeq P_n$, it is easy to see that $Z_c(T) = 1 = Z(T)$. If $T \not\simeq P_n$, then $Z(T) \leq L(T) - 1 \leq Z_c(T) - 1 < Z_c(T)$, where the first inequality is the statement of Theorem 5.6 in [160] and the second inequality follows from Proposition 4.1. By contraposition, $Z_c(T) = Z(T)$ implies $T \simeq P_n$. \square

Proposition 5.3 can be generalized to the following result.

Lemma 5.1. *Let G be a connected graph different from a path. If G has a vertex v to which two or more pendant paths are attached, then $Z(G) < Z_c(G)$.*

Proof. Let R be a minimum connected forcing set of G . By definition v belongs to R_3 , so by Lemma 4.3 v is in R ; moreover, all or all-but-one bases of pendant paths attached to v are in R . Let Z be the set obtained by removing v from R and replacing each base of a pendant path attached to v by the leaf of that pendant path. Then Z is a zero forcing set of G , because each colored leaf will force its pendant path and then v ; at that timestep, the set of colored vertices contains R and is therefore forcing. Thus, Z is a zero forcing set of size $|R| - 1$, so $Z(G) \leq |Z| < Z_c(G)$. \square

Proposition 5.3 allows us to make the following characterization of the minimum zero forcing sets of trees.

Corollary 5.2. *Every minimum zero forcing set of a tree $T \not\cong P_n$ is disconnected.*

I conclude this section with a characterization of the connected forcing numbers of uniclque graphs. Recall that $p(v)$ denotes the number of pendant paths attached to vertex v .

Theorem 5.3. *Let $G = (V, E)$ be a uniclque graph whose maximum clique has vertex set K , $k = |K|$. Then,*

$$Z_c(G) = \begin{cases} |\mathcal{M}(G) \cup K| - 1 & \text{if } \exists u, v \in K : u \notin \mathcal{M}(G), p(v) = 0, u \neq v \\ |\mathcal{M}(G) \cup K| & \text{otherwise.} \end{cases}$$

Moreover, a minimum connected forcing set of G can be found in $O(n)$ time.

Proof. By Proposition 4.6 and Lemma 4.3, every minimum connected forcing set of G contains $\delta(G[K]) = k - 1$ vertices of K and all vertices in \mathcal{M} . Thus, $Z_c(G) \geq |\mathcal{M} \cup K| - 1$. Suppose first that there are distinct vertices $u, v \in K$ such that $u \notin \mathcal{M}$ and $p(v) = 0$, and let $R = \mathcal{M} \cup K \setminus \{u\}$. The only vertices not in R are the vertices belonging to pendant paths (possibly except the bases), and u ; thus, R is connected.

Since $p(v) = 0$, if v has any neighbors outside K , they are all in \mathcal{M} ; thus, v can force u in the first timestep. Then, all pendant paths can be forced either by their bases or by the vertices to which they are attached. Thus, R is a connected forcing set, so $Z_c(G) = |\mathcal{M}(G) \cup K| - 1$.

Now suppose there are no distinct vertices $u, v \in K$ such that $u \notin \mathcal{M}$ and $p(v) = 0$, and let $R = \mathcal{M} \cup K$. The only vertices not in R are the vertices belonging to pendant paths (possibly except the bases); thus, R is connected. Since all vertices in K are initially colored, all pendant paths can be forced either by their bases or by the vertices to which they are attached. Thus, R is a connected forcing set. Suppose there is a connected forcing set R' of size $|\mathcal{M} \cup K| - 1$; R' contains $\delta(G[K]) = k - 1$ vertices of K and all vertices in \mathcal{M} , so R' must exclude a vertex $u \in K$ which is not in \mathcal{M} . By assumption, either every vertex $v \in K$ has $p(v) \geq 1$, or u is the only vertex in K which satisfies $u \notin \mathcal{M}$, $p(u) = 0$. In either case, no vertex can force u because every vertex in K is adjacent to u and to the uncolored base of a pendant path attached to it. Thus, there cannot be a connected forcing set of size $|\mathcal{M} \cup K| - 1$, so $Z_c(G) = |\mathcal{M}(G) \cup K|$.

By Theorem 5.2 and since G is a uniclque graph, K and \mathcal{M} can be found in linear time; moreover, it can be determined in linear time whether there exist distinct vertices $u, v \in K$ such that $u \notin \mathcal{M}$ and $p(v) = 0$. Thus, a minimum connected forcing set can be found in $O(n)$ time. \square

Proposition 5.4. *For a uniclque graph G , $Z_c(G) = Z(G)$ if and only if G is in the family of graphs depicted in Figure 5.2.*

Proof. Let G be a uniclque graph satisfying $Z_c(G) = Z(G)$, let K be the maximum clique of G , and let $k = |K|$. By Lemma 5.1, no vertex in G can have more than

one pendant path attached to it. Thus, G cannot have any vertices in R_2 and R_3 , so G consists of the clique K with at most one pendant path attached to each vertex of K . If every vertex of K has a pendant path attached to it, then by Theorem 5.3 $Z_c(G) = k$, but $Z(G) = k - 1$ since any $k - 1$ leaves of G form a zero forcing set. Thus, at most $k - 1$ of the vertices of K can have pendants attached to them. On the other hand, by Theorem 5.3, any unicycle graph satisfying this condition has $Z_c(G) = k - 1$, and has $Z(G) = k - 1$ since $Z(G) \geq \chi(G) - 1 \geq \omega(G) - 1 = k - 1$. \square

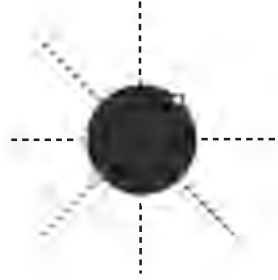


Figure 5.2 : Unicycle graphs for which the zero forcing number equals the connected forcing number. The shaded oval represents a clique, the dotted lines represent paths of arbitrary (possibly zero) length, which are attached to all-but-one vertices of the clique.

5.3 Unicyclic graphs

In this section, I will derive a closed formula for the connected forcing number of a unicyclic graph G and give a linear time algorithm for finding a minimum connected forcing set of G . I first establish two technical lemmas which are applicable to arbitrary graphs that contain a cycle block.

Let G be a connected graph and C be the vertex set of a block of G such that $G[C]$ is a cycle. Given vertices u and v of C , let $(u \leftrightarrow v)$ be the set of vertices of

C encountered while traveling counterclockwise from u to v , not including u and v ; note that $(u \hookrightarrow u)$ is also well-defined. Let $(u \hookrightarrow)$ be the neighbor of u which is counterclockwise of u in C , and $(\leftarrow u)$ be the neighbor of u which is clockwise of u in C . I will refer to $(u \hookrightarrow v)$ as a *segment* of C , and call u and v the *ends* of the segment.

Lemma 5.2. *Let G be a connected graph and C be the vertex set of a block of G such that $G[C]$ is a cycle. Then, any connected forcing set of G can exclude at most one segment of C .*

Proof. Let R be an arbitrary connected forcing set of G . Suppose $(u \hookrightarrow v)$ and $(x \hookrightarrow y)$ are two non-intersecting and non-adjacent segments of C which are not contained in R (note that two intersecting or adjacent segments can be represented as a single segment). Without loss of generality, suppose u, v, x , and y lie on C in this counterclockwise order. Then R contains at least one vertex between v and x , and at least one vertex between y and u ; however, these vertices cannot be connected in $G[R]$ since all paths between them pass through the missing segments in C . Thus, there can be at most one segment of C which is not contained in R . \square

Lemma 5.3. *Let G be a connected graph and C be the vertex set of a block of G such that $G[C]$ is a cycle. A segment of C excluded from a connected forcing set of G can contain at most two cut vertices, each of which is in $R_1(G)$.*

Proof. Let R be an arbitrary connected forcing set of G and $(u \hookrightarrow v)$ be a segment of C not contained in R ; by Lemma 4.3, $\mathcal{M} \subset R$, so $(u \hookrightarrow v)$ cannot contain a vertex of \mathcal{M} . Thus, each vertex in $(u \hookrightarrow v)$ is either a non-cut vertex, or a cut vertex in R_1 ; in the latter case, the entire pendant path attached to the vertex is also not in R since otherwise R could not be connected. Suppose $(u \hookrightarrow v)$ contains three

distinct cut vertices, p , q , and r , lying on C in this counterclockwise order. Every path from a vertex of C outside $(u \hookrightarrow v)$ to a vertex in $(p \hookrightarrow r)$ passes through p or r . However, once p and r are forced by some forcing chains starting outside $(u \hookrightarrow v)$, each of p and r will have two uncolored neighbors and will not be able to force another vertex. Thus, the vertices in $(p \hookrightarrow r)$ cannot be forced; note that $(p \hookrightarrow r) \neq \emptyset$ since $q \in (p \hookrightarrow r)$. This contradicts R being a forcing set, so $(u \hookrightarrow v)$ can contain at most two cut vertices. \square

Lemma 5.4. *Let G be a unicyclic graph, C be the vertex set of the cycle of G , and $(u^* \hookrightarrow v^*)$ be the largest segment of C such that $R^* := \mathcal{M} \cup C \setminus (u^* \hookrightarrow v^*)$ is a forcing set of G . Then R^* is a minimum connected forcing set of G .*

Proof. The vertices in V can be partitioned into \mathcal{M} , $C \setminus \mathcal{M}$, and X , where X is the set of vertices in pendant paths of G which are not in \mathcal{M} ; by Lemma 5.3, $(u^* \hookrightarrow v^*) \subset C \setminus \mathcal{M}$ and any cut vertices in $(u^* \hookrightarrow v^*)$ are in R_1 . Thus $V \setminus R^* = X \cup (u^* \hookrightarrow v^*)$, and deleting all vertices in $X \cup (u^* \hookrightarrow v^*)$ from G does not disconnect it, so R^* is a connected forcing set.

Now suppose there is a connected forcing set R' of G with $|R'| < |R^*|$. By Lemma 4.3, R' contains all vertices in \mathcal{M} . Thus R' must contain at most $|C \setminus \mathcal{M}| - |(u^* \hookrightarrow v^*)| - 1$ vertices of $(C \setminus \mathcal{M}) \cup X$. By Lemma 5.2, the vertices of C not contained in R' must form a segment $(u' \hookrightarrow v')$. If $R' = \mathcal{M} \cup C \setminus (u' \hookrightarrow v')$, then $(u' \hookrightarrow v')$ would be larger than $(u^* \hookrightarrow v^*)$, which contradicts our assumption about $(u^* \hookrightarrow v^*)$; thus, R' includes some vertices of X . These vertices cannot be in pendant paths attached to vertices of $(u' \hookrightarrow v')$, since then R' would be disconnected; if they are in pendant paths attached somewhere other than u' and v' , then a set R'' without them is a smaller connected forcing set than R' , and we can henceforth consider R''

instead of R' . Similarly, if the vertices of R' in X are not the bases of the pendant paths containing them, then since R' is connected, it must also include the bases of the pendant paths, and a set R'' without the non-base vertices of these pendant paths is a smaller connected forcing set than R' . Thus, without loss of generality, suppose the vertices of R' in X are the bases of pendant paths attached to u' or v' . Then, u' and v' would be able to initiate forcing chains. Let R'' be obtained from R' by replacing the vertices in X by the vertices forced by u' and v' . This resulting set is of the form $\mathcal{M} \cup C \setminus (u'' \hookrightarrow v'')$, and has the same cardinality as R' , but $(u'' \hookrightarrow v'')$ is larger than $(u^* \hookrightarrow v^*)$ — a contradiction. Thus, no connected forcing set of G can have cardinality less than $|R^*|$, so R^* is a minimum connected forcing set of G . \square

In view of Lemma 5.4, to find a minimum connected forcing set of a unicyclic graph G with cycle C , one could generate all connected subgraphs of C , check whether each subgraph together with \mathcal{M} is forcing, and find the smallest one, in polynomial time. However, I will include a more thorough case analysis which reduces the number of segments that have to be compared, eliminates the need to check whether a set is forcing, and gives a linear time algorithm for finding a minimum connected forcing set of G .

To this end, we define a *feasible* segment to be a segment $(u \hookrightarrow v)$ for which $R := \mathcal{M} \cup C \setminus (u \hookrightarrow v)$ is a forcing set of G and which is maximal in this regard (with respect to inclusion). Clearly, $(u^* \hookrightarrow v^*)$ described in Lemma 5.4 is the largest feasible segment (or rather, *a* largest feasible segment since there could be several feasible segments with the same maximum cardinality — see, e.g., Figure 5.3). Let $A(C) = \{p_1, \dots, p_k\}$ be the set of cut vertices in C in counterclockwise order. The following lemmas will allow us to enumerate the feasible segments of C ; recall that $p(v)$ denotes the number of pendant paths attached to vertex v .

Lemma 5.5. *Let G be a unicyclic graph, C be the vertex set of the cycle of G and suppose $|A(C)| \geq 3$. Let*

$$f_2(u, v) = \begin{cases} \{(u \hookrightarrow), (\hookleftarrow v)\} & \text{if } p(u) > 0 \text{ and } p(v) > 0 \\ \{(u \hookrightarrow)\} & \text{if } (p(u) > 0 \text{ and } p(v) = 0) \text{ or } ((p(u) = 0 \text{ and } v = u)) \\ \{(\hookleftarrow v)\} & \text{if } p(u) = 0 \text{ and } p(v) > 0 \\ \emptyset & \text{otherwise,} \end{cases}$$

$$I_2 = \{i : p_{i+1} \in R_1, p_{i+2} \in R_1, p_{i+1} \sim p_{i+2}\}, \quad (5.1)$$

and for $i \in I_2$ with i read modulo k , let

$$D_i^2 = (p_i \hookrightarrow p_{i+3}) \setminus f_2(p_i, p_{i+3}). \quad (5.2)$$

Then, the set $\{D_i^2 : i \in I_2\}$ contains the largest feasible segment which has two cut vertices.

Proof. We will first show that if $\{p_i, p_{i+1}, p_{i+2}, p_{i+3}\} \subset A(C)$ with $p_{i+1} \in R_1$, $p_{i+2} \in R_1$, and $p_{i+1} \sim p_{i+2}$, then $S := (p_i \hookrightarrow p_{i+3}) \setminus f(p_i, p_{i+3})$ is a feasible segment. First note that S is indeed a segment, since $f_2(p_i, p_{i+3})$ can only remove the leaves of $G[(p_i \hookrightarrow p_{i+3})]$ from $(p_i \hookrightarrow p_{i+3})$. If $R := \mathcal{M} \cup C \setminus S$ is a set of initially colored vertices, any uncolored vertex in a pendant path, except the ones adjacent to p_{i+1} and p_{i+2} , can be forced either by its base — if its base is in \mathcal{M} — or by the vertex the pendant path is attached to — if its base is not in \mathcal{M} . This includes any pendent paths attached to p_i and p_{i+3} , since if they exist, $(p_i \hookrightarrow)$ and $(\hookleftarrow p_{i+3})$ would respectively be added to the forcing set by f_2 , ensuring that the bases of these paths are the only uncolored neighbors of p_i and p_{i+3} . Thus, both ends of S are either able to initiate a forcing chain reaching p_{i+1} and p_{i+2} , or are themselves p_{i+1} or p_{i+2} (if p_i happens to be adjacent to p_{i+1} , or if $p_{i+2} \sim p_{i+3}$). In either case, p_{i+1} and p_{i+2} will be colored at

some step of the forcing process, whereupon each will be able to force their respective uncolored attached pendant paths. Thus, R is a forcing set of G .

We will now show that S is maximal, by showing that if either end of S is removed from R , the resulting set would not be forcing or would not be connected. First note that if $p(p_i) = 0$ and $p(p_{i+3}) = 0$, S is clearly maximal since by Lemma 5.3, neither p_i nor p_{i+3} can be excluded from the forcing set. Next, note that p_{i+1} and p_{i+2} must be forced by two distinct forcing chains, since if a single forcing chain were to force them, then the first of p_{i+1} and p_{i+2} to be forced would have two uncolored neighbors, and could not force the other. Thus, if one or both of p_i and p_{i+3} are attached to a pendant path, then $(p_i \hookrightarrow)$ and $(\hookleftarrow p_{i+3})$ cannot be removed from R since then one or both ends of the segment would not be able to initiate a forcing chain.

In the special case of $p_i = p_{i+3}$, which happens when $|A(C)| = 3$, p_{i+1} and p_{i+2} must still be forced by two distinct forcing chains; if p_i is attached to a pendant path, then both its clockwise and counterclockwise neighbors must be added to the forcing set; the first case in the definition of f_2 remains valid for this situation. If p_i is not attached to a pendant path, then one of its neighbors (say, the counterclockwise one) must nevertheless be added to the forcing set since p_i cannot initiate two distinct forcing chains on its own. This is reflected in the second case of the definition of f_2 .

Thus, every segment in $\mathcal{D}^2 := \{D_i^2 : i \in I_2\}$ is feasible. Suppose there is a feasible segment S' which contains two cut vertices p and q but which is not in \mathcal{D}^2 . The cut vertices p and q must be adjacent, since otherwise $(p \hookrightarrow q) \neq \emptyset$ and by a similar argument as in Lemma 5.3, the vertices in $(p \hookrightarrow q)$ cannot be forced. Moreover, by Lemma 5.3, p and q must be attached only to single pendant paths; thus, they are some adjacent p_{i+1} and p_{i+2} in R_1 ; note that p_i and p_{i+3} exist (and are possibly equal), since by assumption $|A(C)| \geq 3$. By a similar argument as above, S' cannot

contain either end of $S := (p_i \hookrightarrow p_{i+3}) \setminus f_2(p_i, p_{i+3})$ so S' can be at most equal to S . Moreover, since S' is maximal, it cannot be a proper subset of S since we have shown that $\mathcal{M} \cup C \setminus S$ is a forcing set of G ; therefore, S' is precisely equal to S . Thus, by construction, \mathcal{D}^2 contains every feasible segment which has two cut vertices, and in particular, the largest one. Note that \mathcal{D}^2 could also contain some segments that have fewer cut vertices, which happens if p_{i+1} or p_{i+2} is subtracted from $(p_i \hookrightarrow p_{i+3})$ by $f_2(p_i, p_{i+3})$. \square

Lemma 5.6. *Let G be a unicyclic graph, C be the vertex set of the cycle of G and suppose $|A(C)| \geq 2$. Let*

$$f_1(u, v, w) = \begin{cases} f_2(u, w) & \text{if } u \not\sim v \text{ and } v \not\sim w \\ \{(\hookleftarrow w)\} & \text{if } u \sim v \text{ and } v \not\sim w \text{ and } p(w) > 0 \\ \{(u \hookrightarrow)\} & \text{if } u \not\sim v \text{ and } v \sim w \text{ and } p(u) > 0 \\ v & \text{if } u \sim v, v \sim w, u \neq w, p(u) > 0, p(w) > 0 \\ \{(u \hookrightarrow), (\hookleftarrow u)\} \setminus \{v\} & \text{if } u = w \text{ and } u \sim v \\ \emptyset & \text{otherwise,} \end{cases}$$

$$I_1 = \{i : p_{i+1} \in R_1\}, \quad (5.3)$$

and for $i \in I_1$ with i read modulo k , let

$$D_i^1 = (p_i \hookrightarrow p_{i+2}) \setminus f_1(p_i, p_{i+1}, p_{i+2}). \quad (5.4)$$

Then, the set $\{D_i^1 : i \in I_1\}$ contains the largest feasible segment which has one cut vertex.

Proof. We will first show that if $\{p_i, p_{i+1}, p_{i+2}\} \subset A$ with $p_{i+1} \in R_1$, then $S := (p_i \hookrightarrow p_{i+2}) \setminus f_1(p_i, p_{i+1}, p_{i+2})$ is a maximal segment containing at most one cut vertex for

which $R := \mathcal{M} \cup C \setminus S$ is a forcing set of G . First, by a similar argument as in Lemma 5.5, all pendant paths of G attached to vertices other than p_i and p_{i+2} can get forced by their bases or the vertices to which they are attached. If neither p_i nor p_{i+2} is adjacent to p_{i+1} , then by a similar argument as in Lemma 5.5, two separate forcing chains are needed to color p_{i+1} and the pendant path attached to it; the first case in the definition of f_1 assures that this can happen in the same way as when the segment contains two cut vertices, and that any pendant paths attached to p_i and p_{i+2} whose bases are not in \mathcal{M} get colored as well. If p_i (but not p_{i+2}) is adjacent to p_{i+1} and if p_{i+2} is attached to a pendant path, then f_1 adds $(\leftarrow p_{i+2})$ to the forcing set, which initiates a forcing chain to color S and allows p_{i+1} to force its attached pendant path; then p_i and p_{i+2} will also be able to force any pendant paths attached to them whose bases are not in \mathcal{M} . Similarly, if p_{i+2} is not attached to a pendant path, then it is able to initiate a forcing chain to color S on its own; by symmetry, the same argument shows that S gets colored if p_{i+2} (but not p_i) is adjacent to p_{i+1} . If both p_i and p_{i+2} are adjacent to p_{i+1} , then p_{i+1} must be added to the forcing set only if both p_i and p_{i+2} are attached to pendant paths; this is reflected in the fourth case of the definition of f_1 .

Finally, in the special case of $p_i = p_{i+2}$, which happens when C has two cut vertices, there are several possible situations. If $p_i \not\sim p_{i+1}$, there must again be two forcing chains initiating outside S which force p_{i+1} ; the first line of the definition of f_1 is valid for this case, by a similar reasoning as in the special case of Lemma 5.5. If $p_i \sim p_{i+1}$ and p_i is attached to a pendant path, then a neighbor of p_i in C different from p_{i+1} must be added to R by f_1 , so that this neighbor can initiate a forcing chain around C to p_{i+1} and the pendant path attached to it. If $p_i \sim p_{i+1}$ and p_i is not attached to a pendant path, then any neighbor of p_i in C can be added to R by f_1

to ensure G is forced (including the one different from p_{i+1}). This is reflected in the fifth case of the definition of f_1 . Thus, we have seen that in all cases, R is a forcing set of G .

We will now show that S is maximal, by showing that if either end of S is removed from R , the resulting set would not be forcing, or would not be connected, or would contain two cut vertices. First note that if neither p_i nor p_{i+2} is adjacent to p_{i+1} , then by the same reasoning as in Lemma 5.5, S is maximal. If $p_i \sim p_{i+1}$, then the other end of S must be able to initiate a forcing chain. Thus, if p_{i+2} is attached to a pendant path, then $(\leftarrow p_{i+2})$ cannot be removed from R since then p_{i+2} would not be able to initiate a forcing chain. Similarly, if $p_{i+2} \sim p_{i+1}$, $(p_i \hookrightarrow)$ cannot be removed.

Thus, every segment in $\mathcal{D}^1 := \{D_i^1 : i \in I_1\}$ is a maximal segment containing at most one cut vertex, whose exclusion from $\mathcal{M} \cup C$ yields a forcing set of G . Suppose there is a feasible segment S' containing one cut vertex p which is not in \mathcal{D}^1 . By Lemma 5.3, the cut vertex p must be attached only to a single pendant path. Thus, this is some p_{i+1} in R_1 ; note that p_i and p_{i+2} exist (and are possibly equal), since by assumption $|A(C)| \geq 2$. By a similar argument as above, S' cannot contain either end of $S := (p_i \hookrightarrow p_{i+2}) \setminus f_1(p_i, p_{i+1}, p_{i+2})$, up to the arbitrary choice made by f_1 when it must subtract one of two possible vertices from the segment in order to assure the resulting set is forcing, which does not affect the size of the segment. Thus, S' can be at most equal in size to S ; moreover, since S' is maximal, it cannot be a proper subset of S since we have shown that $\mathcal{M} \cup C \setminus S$ is a forcing set of G . Thus, by construction, \mathcal{D}^1 contains a feasible segment of maximum size among all feasible segments with one cut vertex. □

Lemma 5.7. *Let G be a unicyclic graph, C be the vertex set of the cycle of G and*

suppose $|A(C)| \geq 1$. Let

$$f_0(u, v) = \begin{cases} (u \hookrightarrow) & \text{if } (p(u) \geq 1 \text{ and } p(v) \geq 1) \text{ or } u = v \\ \emptyset & \text{otherwise,} \end{cases}$$

$$I_0 = \{1, \dots, k\}, \quad (5.5)$$

and for $i \in I_0$ with i read modulo k , let

$$D_i^0 = (p_i \hookrightarrow p_{i+1}) \setminus f_0(p_i, p_{i+1}). \quad (5.6)$$

Then, the set $\{D_i^0 : i \in I_0\}$ contains the largest feasible segment which has no cut vertices.

Proof. We will first show that if $\{p_i, p_{i+1}\} \subset A(C)$, $S := (p_i \hookrightarrow p_{i+1}) \setminus f_0(p_i, p_{i+1})$ is a maximal segment containing no cut vertices for which $R := \mathcal{M} \cup C \setminus S$ is a forcing set of G . First, by a similar argument as in Lemma 5.5, all pendant paths of G — except any pendant paths attached to p_i and p_{i+1} whose bases are not in \mathcal{M} — can get forced by their bases or by the vertices to which they are attached. If at least one of p_i and p_{i+1} is not attached to a pendant path, then the vertex not attached to a pendant path can initiate a forcing chain which colors $(p_i \hookrightarrow p_{i+1})$, and then the other vertex will be able to force its pendant path whose base is not in \mathcal{M} , if it exists. Similarly, if both p_i and p_{i+1} are attached to pendant paths, then f_0 adds $(p_i \hookrightarrow)$ to the forcing set, which is able to initiate a forcing chain to color S . Note that if $p_i \sim p_{i+1}$, $R = \mathcal{M} \cup C$ regardless of whether or not p_i and p_{i+1} are attached to pendant paths. Thus, R is a forcing set of G . R is also maximal, since if either end of S is removed from R , the resulting set would either contain a cut vertex, or would not be forcing, or would not be connected.

In the special case of $p_i = p_{i+1}$, which happens when G has a single cut vertex p_1 , regardless of whether or not p_1 is attached to a pendant path, one of its neighbors in C — say, the counterclockwise one — must be added to R in order to initiate a forcing chain. This is reflected in the first case of the definition of f_0 .

Thus, every segment of $\mathcal{D}^0 := \{D_i^0 : i \in I_0\}$ is a maximal segment containing no cut vertices whose exclusion from $\mathcal{M} \cup C$ yields a forcing set of G . Moreover, every feasible segment containing no cut vertices must either have two ends which are cut vertices, at least one of which is not attached to a pendant path, or have one end which is a cut vertex attached to a pendant path, and another end which is a neighbor of a cut vertex attached to a pendant path — otherwise the segment would contain a cut vertex, or would not be forcing, or would not be maximal. Thus, by construction, \mathcal{D}^0 contains every feasible segment which has no cut vertices, up to the arbitrary choice of whether $f_0(u, v)$ subtracts $(u \hookrightarrow)$ or $(\hookleftarrow v)$, which does not affect the size of the segment. In particular, \mathcal{D}^0 contains a feasible segment of maximum size among all feasible segments with no cut vertices. \square

For an illustration of the constructions in Lemmas 5.5, 5.6, and 5.7, see Figure 5.3 which shows a unicyclic graph with feasible segments of maximum size containing zero, one, and two cut vertices.

Theorem 5.4. *Let G be a unicyclic graph and C be the vertex set of the cycle of G . For $0 \leq j \leq 2$, if $|A(C)| > j$, let $D_{max}^j = \max_{i \in I_j} \{|D_i^j|\}$, where I_j and D_i^j are defined as in (1)–(6); if $|A(C)| \leq j$, let $D_{max}^j = 0$. Let i^* and j^* be such that $|D_{i^*}^{j^*}| = \max\{D_{max}^0, D_{max}^1, D_{max}^2\}$. Then,*

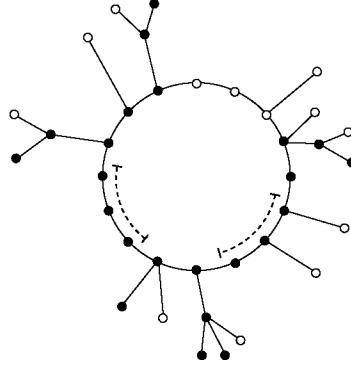


Figure 5.3 : A unicyclic graph and a minimum connected forcing set. Two other minimum connected forcing sets can be obtained by coloring the uncolored segment of C and removing one of the segments indicated by the dashed lines.

$$Z_c(G) = \begin{cases} 2 & \text{if } |A(C)| = 0 \\ |\mathcal{M} \cup C \setminus D_{i*}^{j*}| & \text{if } |A(C)| \geq 1, \end{cases}$$

and a minimum connected forcing set of G can be found in $O(n)$ time.

Proof. If $|A(C)| = 0$, then G is a cycle, and two adjacent vertices of G clearly form a minimum connected forcing set. Thus, we will henceforth assume $|A(C)| \geq 1$. By Lemma 5.3, a feasible segment can have at most two cut vertices. By Lemmas 5.5, 5.6, and 5.7, D_{i*}^{j*} is the largest feasible segment of C , and by Lemma 5.4, $|\mathcal{M} \cup C \setminus D_{i*}^{j*}|$ is a minimum connected forcing set of G .

To verify that the time needed to find D_{i*}^{j*} is linear in the order of the graph, first note that the set of cut vertices in G , and hence the vertices in \mathcal{M} , C , and $A(C)$, can be found in linear time (cf. [192]). Then, the sets $(p_i \hookrightarrow p_{i+1})$, $1 \leq i \leq k$, can also be found in linear time. These sets of cut vertices and other vertices can be stored (with linear space), and each of the functions f_0 , f_1 , and f_2 and D_i^j can be computed in constant time for $0 \leq j \leq 2$ and $1 \leq i \leq k$. Since each of the index sets I_0 , I_1 , and I_2

has at most $|A(C)| = O(n)$ elements, D_i^{j*} can be found by computing the maximum of $O(n)$ terms. \square

The zero forcing number and path cover number of unicyclic graphs have been investigated in [125, 105, 198] and have been shown to coincide. I conclude this section by characterizing the unicyclic graphs for which $Z(G) = Z_c(G)$, and thus describing the connectivity of the minimum zero forcing sets of unicyclic graphs.

Proposition 5.5. *For a unicyclic graph G , $Z_c(G) = Z(G)$ if and only if G is in the family of graphs depicted in Figure 5.4.*

Proof. Let G be a unicyclic graph satisfying $Z_c(G) = Z(G)$. Let C be the vertex set of the cycle of G , and let R be an arbitrary minimum connected forcing set of G . By Lemma 5.1, no vertex in G can have more than one pendant path attached to it. Thus, G cannot have any vertices in R_2 and R_3 , so G consists of the cycle C with at most one pendant path attached to each vertex of C .

Now, for any $v \in C$, define $\ell(v)$ to be v if v is not a cut vertex, and to be the leaf of the pendant path attached to v if v is a cut vertex. Suppose for contradiction that $|R \cap C| \geq 3$. As shown in Lemma 5.2, R can exclude at most one segment of C , which implies that $R \cap C$ also forms a segment; thus, there are vertices $\{u, v, w\} \subset R \cap C$ such that $u \sim v$ and $v \sim w$. We claim $Z := R \setminus \{u, v, w\} \cup \{\ell(u), \ell(v)\}$ is a zero forcing set of G . To see why, note that whether or not u and v are attached to pendant paths, at the first stage of the forcing process, $\ell(u)$ and $\ell(v)$ can initiate forcing chains which color u and v ; next, w will be the only uncolored neighbor of v , and v can force w . At this point, the set of colored vertices contains R , and can therefore color all of G ; this means $Z(G) \leq |Z| < |R| = Z_c(G)$ — a contradiction. Thus, $|R \cap C| < 3$; on the other hand, by Proposition 4.6, $|R \cap C| \geq 2$, so $|R \cap C| = 2$. By inspection, only the

unicyclic graphs in Figure 5.4 satisfy this condition and the condition that each cut vertex of C is in R_1 .

Remark 5.1. Note that Figure 5.4 does not include all unicyclic graphs with zero forcing number 2; one can easily find a unicyclic graph G with $Z(G) = 2$ and $Z_c(G) > 2$. □

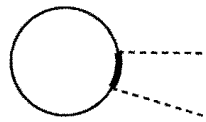


Figure 5.4 : Unicyclic graphs for which the zero forcing number equals the connected forcing number. The solid line represents a cycle of arbitrary size; the bold line represents a single edge; the dotted lines represent paths of arbitrary (possibly zero) length.

Proposition 5.5 allows us to make the following characterization of the minimum zero forcing sets of unicyclic graphs.

Corollary 5.3. *For all unicyclic graphs except the ones in Figure 5.4, any minimum zero forcing set is disconnected.*

5.4 Cactus and block graphs with no pendant paths

I will now characterize the connected forcing numbers of cactus and block graphs which have no pendant paths. The following definition will be used in the sequel.

Definition 5.1. Let G be a graph, let $G_0 = G$, and for $i \geq 1$, let $G_i = G_{i-1} - \{\text{all non-cut vertices of the outer blocks of } G_{i-1}\}$. We will say a block of G has *depth* i if it is an outer block of G_i .

Proposition 5.6. *Let $G = (V, E)$ be a block graph with no pendant paths and b be the number of blocks of G which have at least one non-cut vertex. Then $Z_c(G) = n - b$.*

Proof. Let Q be a set containing one non-cut vertex from each block of G which has non-cut vertices. We claim that $R := V \setminus Q$ is a minimum connected forcing set of G . $G[R]$ is clearly connected, since deleting one non-cut vertex from each block by definition does not disconnect G . Next, since G has no pendant paths, each outer block of G has size at least 3; thus, each outer block has at least two non-cut vertices, one of which is in Q and the other of which is in R and can force the first. Thus, each block at depth 0 can be forced. Now suppose every block at depth at most $i \geq 0$ has been colored and let B be a block at depth $i + 1$. By definition, B must have a cut vertex p adjacent only to blocks at depth less than $i + 1$ (besides B), since otherwise B would not be an outer block when all blocks of smaller depth are deleted. Since all blocks adjacent to p besides B have been colored by assumption, p can force an uncolored non-cut vertex in B , if such a vertex exists. Thus, each block at depth $i + 1$ will get colored as well. By induction, every block in the graph can get forced by R , so R is a connected forcing set.

Now, let S be an arbitrary minimum connected forcing set of G . By Proposition 4.6, S must contain at least $\delta(G[B]) = |B| - 1$ vertices from each block B of G . Moreover, since G has no pendant paths, all cut vertices of G are in R_2 or R_3 , so by Lemma 4.3, all cut vertices of G must be in S . Thus, S can exclude at most one vertex from each of the b blocks that have non-cut vertices, so $|S| \geq n - b$. Thus, R is a minimum connected forcing set. \square

Proposition 5.7. *Let $G = (V, E)$ be a cactus graph with no pendant paths. Let \mathcal{C} be the collection of vertex sets of cycles of G and b be the number of outer blocks of G .*

For $C \in \mathcal{C}$, let D_C be the largest segment of C which does not contain cut vertices of C . Then, $Z_c(G) = n - \sum_{C \in \mathcal{C}} |D_C| + b$, if G is not a cycle, and $Z_c(G) = 2$ if G is a cycle.

Proof. Clearly $Z_c(G) = 2$ if G is a cycle, so suppose henceforth that G is not a cycle. Let Q be a set containing one vertex from each outer block of G which is adjacent to the cut vertex of the outer block. Let $\mathcal{D} := \bigcup_{C \in \mathcal{C}} D_C$ and $R := (V \setminus \mathcal{D}) \cup Q$. We claim that R is a minimum connected forcing set of G . R is connected, since deleting one segment containing no cut vertices from each cycle block does not disconnect G . Next, since G has no pendant paths, each outer block of G is a cycle in which the cut vertex and one of its neighbors (the one in Q) are in R . In each outer block, the colored neighbor of the cut vertex will initiate a forcing chain around the cycle; thus, each block at depth 0 can be forced. Now suppose every block at depth at most $i \geq 0$ has been colored and let B be a block at depth $i + 1$. If B is a cut edge block, then both vertices of B are already in R . If B is a cycle block, then let $(u \hookrightarrow v)$ be the segment missing from B . By definition, one of u and v — say, u — must be adjacent only to blocks at depth less than $i + 1$ (besides B), since otherwise B would not be an outer block when all blocks of smaller depth are deleted. Since all blocks adjacent to u besides B have been colored by assumption, u can initiate a forcing chain which colors the segment $(u \hookrightarrow v)$. Thus, each block at depth $i + 1$ will get colored as well. By induction, every block in the graph can get forced by R , so R is a connected forcing set.

Finally, suppose there is a connected forcing set R' with $|R'| < |R|$. Let B be the vertex set of some block of G which contains fewer vertices of R' than of R . B cannot be the vertex set of a cut edge block of G , since both vertices in each cut edge block are in R_2 and hence belong to R' by Lemma 4.3. B also cannot be an outer

block, since every outer block is a cycle and by Proposition 4.6, R' contains at least $\delta(G[B]) = 2$ vertices from each such block. Thus, B must be a non-outer cycle block of G . However, by Lemma 5.2, the vertices of B which R' excludes form a segment of B , but by construction, this segment cannot be bigger than the segment excluded from R . This is a contradiction, so R is a minimum connected forcing set. Since the segments $\{D_C : C \in \mathcal{C}\}$ are disjoint, $Z_c(G) = n - \sum_{C \in \mathcal{C}} |D_C| + b$. \square

Remark 5.2. The characterizations of Propositions 5.6 and 5.7 are constructive, and minimum connected forcing sets of cactus and block graphs with no pendant paths can be found in linear time, by a similar analysis as in Theorem 5.4.

Chapter 6

Graphs with extremal connected forcing numbers

One way to address the complexity of connected forcing is to derive closed-form expressions and polynomial time algorithms for computing the connected forcing numbers of special classes of graphs, as was done in Chapter 5. Conversely, a complete characterization of graphs having a particular connected forcing number can be obtained through a combinatorial case analysis; such is the content of this chapter. In particular, I characterize the graphs with connected forcing numbers 1, 2, $n - 1$, and $n - 2$. Graphs with zero forcing numbers 1, 2, and $n - 1$ have been characterized in [207], and graphs with zero forcing number $n - 2$ have been characterized in [89]. Other related characterizations have been derived for graphs whose minimal rank is two [126, 208] and three [209], graphs whose positive semi-definite matrices have nullity at most two [210], three-connected graphs whose maximum nullity is at most three [211], and graphs for which the maximum multiplicity of an eigenvalue is two [206]. Many of these characterizations have been obtained using linear algebraic approaches; in contrast, I employ novel combinatorial and graph theoretic techniques which make use of the vertex connectivity of a graph and the connectedness of its forcing set.

6.1 Graphs with $Z_c(G) = 1$ and $Z_c(G) = n - 1$

The following characterizations easily follow from the definition of zero forcing and the properties of forcing chains.

Observation 6.1. *Let G be a graph. Then,*

- $Z(G) = 1$ if and only if $G \simeq P_n$
- $Z_c(G) = 1$ if and only if $G \simeq P_n$
- $Z(G) = n$ if and only if $G \simeq \dot{\bigcup}_{i=1}^n K_1$
- $Z_c(G) = n$ if and only if $G \simeq K_1$
- $Z(G) = n - 1$ if and only if $G \simeq \left(\dot{\bigcup}_{i=1}^t K_1\right) \dot{\cup} K_{n-t}$ for $n \geq 2$ and $0 \leq t \leq n - 2$.

Below I characterize the graphs which have $Z_c(G) = n - 1$.

Theorem 6.1. *Let $G = (V, E)$ be a connected graph. Then, $Z_c(G) = n - 1$ if and only if $G \simeq K_n$, $n \geq 2$, or $G \simeq K_{1,n-1}$, $n \geq 4$.*

Proof. If $G \simeq K_n$ or $G \simeq K_{1,n-1}$, it is easy to verify that $Z_c(G) = n - 1$. Now suppose $Z_c(G) = n - 1$. If G has no separating set of vertices, then G is a complete graph K_n .

We will show that if G has a separating set $S \subset V$ then G must be a star $K_{1,n-1}$.

Let S be a separating set of minimum cardinality, and suppose first that $|S| > 1$. Let V_1 be the vertex set of a component of $G - S$; pick an edge sw , $s \in S$, $w \in V_1$, and pick a vertex $v \in V \setminus (S \cup V_1)$ such that $\{v, s\}$ is not a separating set. If $|S| > 2$, it is clear that such a v exists; if $|S| = 2$, $G - s$ must be connected, so simply pick v to be a non-cut vertex of $G - s$ which is not in S or V_1 . Then, $V \setminus \{v, s\}$ is a connected forcing set of G , since $N(w) \subset V_1 \cup S$, so all of the neighbors of w except s are colored;

thus, w will force s and then any neighbor of v will force v . Thus we have found a connected forcing set of size $n - 2$, a contradiction.

Suppose now that $|S| = 1$, and suppose first that there are multiple cut vertices in G . Let A and B be two blocks of G which are leaves of the block tree of G and which are not incident to the same cut vertex. Color every vertex except one non-cut vertex in A and one non-cut vertex in B . This is clearly a connected forcing set of size $n - 2$ – a contradiction. Now suppose there is only one cut vertex p . If any block B incident to p is nontrivial, color everything except one (non-cut) vertex in B and one (non-cut) vertex in a different block. Then, the uncolored vertex in B will be forced by one of its neighbors in B , and then the uncolored vertex in the other block will be forced by any of its neighbors – a contradiction. Thus, every block incident to p must be a trivial block, so $G \simeq K_{1,n-1}$. \square

6.2 Graphs with $Z_c(G) = 2$

In this section, I characterize all graphs with connected forcing number 2. I first recall some definitions and previous results.

Definition 6.1. A graph $G = (V, E)$ is a *graph of two parallel paths* specified by V_1 and V_2 if $G \not\cong P_n$, and if V can be partitioned into nonempty sets V_1 and V_2 such that $P_1 := G[V_1]$ and $P_2 := G[V_2]$ are paths, and such that G can be drawn in the plane in such a way that P_1 and P_2 are parallel line segments, and the edges between P_1 and P_2 (drawn as straight line segments) do not cross; such a drawing of G is called a *standard drawing*. In a standard drawing of G , fix an ordering of the vertices of P_1 and P_2 that is increasing in the same direction for both paths. In this ordering, let $first(P_i)$ and $last(P_i)$ respectively denote the first and last vertices of P_i for $i = 1, 2$.

The sets $\{first(P_1), first(P_2)\}$ and $\{last(P_1), last(P_2)\}$ will be referred to as *ends* of G .

Note that if G is a graph of two parallel paths, there may be several different partitions of V into V_1 and V_2 which satisfy the conditions above. For example, let $G = (\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 1\}\})$ be a cycle on 5 vertices. Then G is a graph of two parallel paths that can be specified by $V_1 = \{1\}$ and $V_2 = \{2, 3, 4, 5\}$, as well as by $V_1 = \{1, 2, 3\}$ and $V_2 = \{4, 5\}$.

Graphs of two parallel paths were introduced by Johnson et al. [206] in relation to graphs with maximum nullity 2. They were also used by Row [207] in the following characterization.

Theorem 6.2 ([207]). *$Z(G) = 2$ if and only if G is a graph of two parallel paths.*

The following observation regarding the result of Theorem 6.2 is readily verifiable (and has been noted in [207]).

Observation 6.2. *Either end of a graph on two parallel paths is a zero forcing set. Conversely, if $Z(G) = 2$, the two forcing chains associated with a minimum zero forcing set induce a specification of G as a graph on two parallel paths.*

The following observation follows from the definition of forcing vertices.

Observation 6.3. *Every minimum zero forcing set and every minimum connected forcing set contains a vertex together with all-but-one of its neighbors.*

I now prove the main result of this section.

Theorem 6.3. *$Z_c(G) = 2$ if and only if G belongs to the family of graphs described in Figures 6.1 and 6.2.*

Proof. Let $G = (V, E)$ be a graph with $Z_c(G) = 2$. Since $Z(G) = 1$ if and only if $Z_c(G) = 1$, and since $2 = Z_c(G) \geq Z(G)$, it follows that $Z(G) = 2$. Thus, by

Theorem 6.2, G is a graph of two parallel paths. Fix some partition of V into V_1 and V_2 which satisfies Definition 6.1, fix a standard drawing of G based on that partition, and fix a vertex ordering as specified in Definition 6.1. From Proposition 4.1, it follows that G has 0, 1, or 2 leaves.

Claim 6.1. *Let G be a graph of two parallel paths with $Z_c(G) = 2$. Then, there are at least two edges between the two parallel paths of G .*

Proof. If there are no edges between the two parallel paths of G , then G is disconnected, and cannot have a connected forcing set. If there is one edge between the two parallel paths, then G is either isomorphic to a path (and is hence not a graph of two parallel paths), or has more than two leaves (and hence $Z_c(G) > 2$ by Proposition 4.1). Thus, there must be at least two edges with one endpoint in V_1 and the other in V_2 . \square

We will now consider several cases based on the number and position of the leaves in G . Let L be the set of leaves of G .

Claim 6.2. *Let G be a graph of two parallel paths which has 0 leaves, 1 leaf, or 2 leaves which belong to the same end of G . Then, $Z_c(G) = 2$, and G belongs to the family of graphs described in Figure 6.1.*



Figure 6.1 : A graph of two parallel paths with 0 leaves, 1 leaf, or 2 leaves which belong to the same end of the graph; solid lines represent paths of arbitrary (possibly zero) length; bold lines represent mandatory single edges; dashed lines represent any configuration of non-intersecting edges between the parallel paths.

Proof. Without loss of generality, suppose $L \subseteq \{last(V_1), last(V_2)\}$. Let $V'_1 \subseteq V_1$ and $V'_2 \subseteq V_2$ be maximal sets of vertices which do not belong to pendant paths of G . By Claim 6.1, there are at least two distinct edges with one endpoint in V'_1 and the other in V'_2 ; thus, it follows that at least one of the paths $G[V'_1]$ and $G[V'_2]$ must have length greater than zero. Moreover, by Observation 6.2, and since $first(V_1)$ and $first(V_2)$ are adjacent, it follows that $\{first(V_1), first(V_2)\}$ is a connected forcing set; thus, $Z_c(G) = 2$. This is the family of graphs illustrated in Figure 6.1. \square

Claim 6.3. *Let G be a graph of two parallel paths that has 2 leaves which belong to the same path and different ends of G . Then $Z_c(G) = 2$ if and only if G belongs to the family of graphs described in Figure 6.2.*



Figure 6.2 : A graph of two parallel paths with 2 leaves which belong to different ends of the graph; solid lines represent paths of arbitrary (possibly zero) length; bold lines represent mandatory single edges; dashed lines represent a configuration of non-intersecting edges between the parallel paths, all of which are incident to the same vertex in the path containing the mandatory single edge.

Proof. Without loss of generality, suppose $L = \{first(V_1), last(V_1)\}$. Let H_1 be the pendant path containing $first(V_1)$, and u_1 be the vertex to which H_1 is attached; let H_2 be the pendant path containing $last(V_1)$, and u_2 be the vertex to which H_2 is attached. Let $V'_1 = V_1 \setminus (H_1 \cup H_2)$.

Suppose first that $|V'_1| = 1$. Then $u_1 = u_2$, and H_1 and H_2 are both attached to u_1 ; thus, by Lemma 4.3, u_1 and some neighbor z of u_1 in H_1 or H_2 must be contained in every minimum connected forcing set of G . However, a set containing only u_1 and

z is not forcing, since u_1 has at least two uncolored neighbors outside $H_1 \cup H_2$; this is a contradiction.

Suppose next that $|V'_1| \geq 3$, and let $R = \{r_1, r_2\}$ be a connected forcing set of G . By Observation 6.3, and since neither leaf of G together with its neighbor forms a forcing set, it follows that at least one of r_1 and r_2 has degree 2. Without loss of generality, let r_1 be a vertex of degree 2. If r_1 is contained in H_i , for $i \in \{1, 2\}$, then no vertex outside $H_i \cup \{u_i\}$ can be forced by R . Similarly, if r_1 is contained in V'_1 , then no vertex outside V'_1 can be forced by R , and if r_1 is contained in V'_2 , then no vertex outside $V'_2 \cup \{u_1, u_2\}$ can be forced by R . Thus, the assumption that $|V'_1| \geq 3$ leads to a contradiction, so it follows that $|V'_1| = 2$, i.e., $V'_1 = \{u_1, u_2\}$. Recall that by Claim 6.1, each of u_1 and u_2 must be adjacent to at least one vertex of V_2 – namely, $first(V_2)$ or $last(V_2)$, respectively.

Suppose first that both u_1 and u_2 are adjacent to two or more vertices of V_2 . Let v_1 and v_2 respectively be the neighbors of u_1 and u_2 in V_2 which are respectively closest to $first(V_2)$ and $last(V_2)$ in $G[V_2]$; v_1 and v_2 could possibly be the same vertex. Let S_1 , S_2 , and S_3 respectively be the sets of vertices between $first(V_2)$ and v_1 , v_1 and v_2 , and v_2 and $last(V_2)$, inclusively (where “between” refers to the vertex ordering of G , i.e., to the position of the vertices in the path $G[V_2]$). As shown in the case where $|V'_1| \geq 3$, the degree 2 vertex r_1 cannot be contained in H_i , $i \in \{1, 2\}$. Similarly, if r_1 is contained in S_1 , S_2 , or S_3 , then, respectively, no vertex outside $S_1 \cup \{u_1\}$, S_2 , and $S_3 \cup \{u_2\}$ can be forced by R . Once again, it follows that no set consisting of a degree 2 vertex and one of its neighbors can force all of G , a contradiction.

Thus, one of u_1 and u_2 , say u_1 , must be adjacent to a single vertex of V_2 , namely $first(V_2)$. Then $\{u_2, last(V_2)\}$ is a connected forcing set, since $last(V_2)$ can initiate a forcing chain passing through all vertices in V_2 and eventually forcing u_1 ; then u_1

and u_2 will be able to force H_1 and H_2 , respectively. This is the family of graphs illustrated in Figure 6.2. \square

Claim 6.4. *Let G be a graph of two parallel paths which has 2 leaves which belong to different paths and different ends of G . Then $Z_c(G) = 2$ if and only if G (can be respecified as a graph which) belongs to the family of graphs described in Figure 6.2.*

Proof. Without loss of generality, suppose $L = \{\text{first}(V_1), \text{last}(V_2)\}$. Let H_1 be the pendant path containing $\text{first}(V_1)$, and u_1 be the vertex to which H_1 is attached; let H_2 be the pendant path containing $\text{last}(V_2)$, and u_2 be the vertex to which H_2 is attached. Let $V'_1 = V_1 \setminus H_1$ and $V'_2 = V_2 \setminus H_2$. Since G is different from a single path, it cannot be the case that $|V'_1| = 1$ and $|V'_2| = 1$.

Suppose $|V'_1| \geq 2$ and $|V'_2| \geq 2$, and let $R = \{r_1, r_2\}$ be a connected forcing set of G . By the same argument as in Claim 6.3, one of r_1 and r_2 , say r_1 , must have degree 2; moreover, r_1 cannot be contained in H_1 or H_2 . If r_1 is contained in V'_1 , then no vertex outside $V'_1 \cup \{u_2\}$ can be forced by R , a contradiction. By symmetry, r_1 also cannot be in V'_2 .

Thus, exactly one of V'_1 and V'_2 consists of a single vertex; without loss of generality, suppose $|V'_1| = 1$ and $|V'_2| \geq 2$. Note then, that $u_1 = \text{last}(V_1)$, and all edges between V'_1 and V'_2 are incident to u_1 . Let w be the neighbor of u_2 in V'_2 , which exists by the assumption that $|V'_2| \geq 2$. Then, the vertex partition $\widehat{V}_1 = H_1 \cup \{u_1, u_2\} \cup H_2$, $\widehat{V}_2 = V \setminus \widehat{V}_1$ gives an alternate specification of G as a graph of two parallel paths. In this specification, the two leaves of G belong to the same path and different ends of G . Thus, by Claim 6.3, $Z_c(G) = 2$ if and only if G belongs to the family of graphs described in Figure 6.2. \square

Since there are no other possible positions for the leaves of G , this concludes the proof

of Theorem 6.3. □

6.3 Graphs with $Z_c(G) = n - 2$

In this section, I will characterize all graphs with connected forcing number $n - 2$. I begin by recalling a result regarding graphs with zero forcing number $n - 2$.

Theorem 6.4 ([89]). *$Z(G) \geq n - 2$ if and only if G does not contain an induced subgraph isomorphic to any of the graphs in Figure 6.3.*

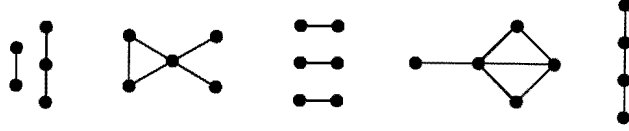


Figure 6.3 : Forbidden induced subgraphs for $Z(G) \geq n - 2$; from left to right: $P_2 \dot{\cup} P_3$, “fish”, $P_2 \dot{\cup} P_2 \dot{\cup} P_2$, “dart”, P_4 .

Theorem 6.4 is a consequence of the following characterization of the graphs with minimum Hermitian rank at most 2, due to Barrett, van der Holst, and Loewy [126].

Theorem 6.5 ([126]). *Given a graph $G = (V, E)$ with vertex set $V = \{1, \dots, n\}$, let $H(G)$ be the set of all Hermitian $n \times n$ matrices $A = [a_{ij}]$ such that $a_{ij} \neq 0$ for $i \neq j$, if and only if $\{i, j\} \in E$ (and no restriction on a_{ii}). Let $\text{hmr}(G) = \min\{\text{rank}(A) : A \in H(G)\}$. Then, the following are equivalent:*

1. $\text{hmr}(G) \leq 2$.
2. G^c has the form $(K_{s_1} \dot{\cup} \dots \dot{\cup} K_{s_t} \dot{\cup} K_{p_1, q_1} \dot{\cup} \dots \dot{\cup} K_{p_k, q_k}) \vee K_r$ where $t, s_1, \dots, s_t, k, p_1, q_1, \dots, p_k, q_k, r$ are nonnegative integers and $p_i + q_i > 0$ for $1 \leq i \leq k$.
3. G is $(P_2 \dot{\cup} P_3, \text{“fish”}, P_2 \dot{\cup} P_2 \dot{\cup} P_2, \text{“dart”}, P_4)$ -free.

The proof of Theorem 6.5, and the relation between $\text{hmr}(G)$ and $Z(G)$ used in the proof of Theorem 6.4, are obtained primarily through linear algebraic techniques. In contrast, in this section, I will develop and use predominantly combinatorial and graph theoretic techniques to derive a characterization of graphs satisfying $Z_c(G) = n - 2$. The following characterization of graphs whose zero forcing number equals $n - 2$ easily follows from Theorem 6.4; this characterization will be used in the sequel.

Corollary 6.1. *$Z(G) = n - 2$ if and only if G satisfies the following conditions:*

1. *G does not contain any of the graphs in Figure 6.3 as induced subgraphs,*
2. *$G \not\cong \dot{\bigcup}_{i=1}^n K_1$,*
3. *$G \not\cong \left(\dot{\bigcup}_{i=1}^t K_1\right) \dot{\cup} K_{n-t}$ for $n \geq 2$ and $0 \leq t \leq n - 2$.*

Proof. By Observation 6.1, the second condition in the statement of Corollary 6.1 is satisfied if and only if $Z(G) = n$, and the third condition is satisfied if and only if $Z(G) = n - 1$. □

The following is a novel concept in the study of forcing sets, and will be useful in proving a technical lemma. Further study of this restriction of connected forcing (and analogously of zero forcing) would be interesting in its own right.

Definition 6.2. For any $S \subset V$, let $Z_c(G; S)$ be the cardinality of the minimum connected forcing set of G which contains S .

For example, let $G = (\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\})$ be a path on 5 vertices. Then $Z_c(G; \{1\}) = 1$, $Z_c(G; \{2, 3\}) = 2$, and $Z_c(G; \{1, 5\}) = 5$.

Lemma 6.1. *Let G be a biconnected graph different from K_n . Then for any $v \in V$, $Z_c(G; \{v\}) \leq n - 2$.*

Proof. Note that since G is biconnected and not complete, it must have at least 4 vertices. Let v^* be an arbitrary vertex of G and suppose for contradiction that $Z_c(G; \{v^*\}) = n - 1$.

Suppose first that some $\{v_1, v_2\} \subset V \setminus \{v^*\}$ forms a separating set of G . Let u be a vertex which is not a cut vertex of $G - v_1$ and which belongs to a component of $G - \{v_1, v_2\}$ that does not contain v^* (it is easy to see that such a vertex exists). We claim that $R = V \setminus \{v_1, u\}$ is a connected forcing set of G . To see why, note first that by construction R is connected. Moreover, some colored neighbor of v_1 in the component of $G - \{v_1, v_2\}$ containing v^* can force v_1 in the first timestep; then, any neighbor of u can force u . Thus, G cannot have a separating set of size 2.

Let v be any vertex in $V \setminus \{v^*\}$ and suppose there is a vertex $u \in V$ which is not adjacent to v ; let w be a neighbor of u different from v^* (which exists since G is biconnected). Then, $V \setminus \{v, w\}$ is a connected forcing set of G , since u can force w in the first timestep, and then v can be forced by any of its neighbors; moreover, since G has no separating sets of size 2, this set is connected. However, since we assumed that $Z_c(G; \{v^*\}) = n - 1$, it follows that every $v \in V \setminus \{v^*\}$ is adjacent to every vertex in V . This implies that G is a complete graph, a contradiction. \square

The following definition is a generalization of Definition 4.1.

Definition 6.3. A *pendant tree attached to vertex v* in graph $G = (V, E)$ is a maximal set $T \subset V$ composed of the vertices of the connected components of $G - v$ which are trees and which have a single vertex adjacent to v in G .

I now prove the main result of this section.

Theorem 6.6. $Z_c(G) = n - 2$ if and only if G belongs to the family of graphs described in Figures 6.4–6.9.

Proof. Let $G = (V, E)$ be a graph with $Z_c(G) = n - 2$. If G does not have a separating set, then G is a complete graph, and $Z_c(G) = n - 1$. Note also that G is connected; thus, $\kappa(G) \geq 1$. We will consider several cases based on the vertex connectivity of G , starting with $\kappa(G) = 1$. We will say v is a *feasible* vertex if v is part of exactly one nontrivial block of G and if every trivial block adjacent to v is part of a pendant tree. If v is a feasible vertex, define $\ell(v)$ to be v if v is not a cut vertex, and otherwise to be some leaf of G in the pendant tree attached to v . Note that for any feasible vertex v , deleting $\ell(v)$ does not disconnect G .

Claim 6.5. *If G is a graph with $\kappa(G) = 1$ and if G has three or more nontrivial blocks, then $Z_c(G) \leq n - 3$.*

Proof. From the structure of G it follows that G has two nontrivial blocks B_1 and B_2 with feasible vertices u_1, v_1 in B_1 and feasible vertices u_2, v_2 in B_2 , and a nontrivial block B_3 with a feasible vertex v_3 . We claim that $V \setminus \{\ell(v_1), \ell(v_2), \ell(v_3)\}$ is a connected forcing set. To see why, note that $\ell(v_1)$ and $\ell(v_2)$ each have a neighbor which is not adjacent to another vertex in $\{\ell(v_1), \ell(v_2), \ell(v_3)\}$; therefore, $\ell(v_1)$ and $\ell(v_2)$ can be forced in the first timestep, and then any neighbor of $\ell(v_3)$ can force $\ell(v_3)$. Thus, $Z_c(G) \leq n - 3$. \square

Claim 6.6. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) = 1$, and two nontrivial blocks. Then, G belongs to the family of graphs described in Figure 6.4.*



Figure 6.4 : Shaded ovals represent cliques, each of size at least 3; dotted line represents a path of possibly zero length.

Proof. Let B_1 and B_2 be the nontrivial blocks of G . Suppose first that G also has at least one trivial outer block. It is easy to see that there are at least two feasible vertices u_1, v_1 in B_1 and at least two feasible vertices u_2, v_2 in B_2 . Let v_3 be a leaf vertex of some pendant tree of G , which, without loss of generality, does not coincide with $\ell(v_1)$ and $\ell(v_2)$ (although it may coincide with $\ell(u_1)$ or $\ell(u_2)$). We claim that $V \setminus \{\ell(v_1), \ell(v_2), v_3\}$ is a connected forcing set. To see why, note that at least one of $\ell(v_1)$ and $\ell(v_2)$, say $\ell(v_1)$, has a neighbor which is not adjacent to another vertex in $\{\ell(v_1), \ell(v_2), v_3\}$. Therefore $\ell(v_1)$ can be forced in the first timestep; then, any neighbor of $\ell(v_2)$ can force $\ell(v_2)$, and then the neighbor of v_3 can force v_3 . Thus, $Z_c(G) \leq n - 3$, a contradiction.

Now suppose G has no trivial outer blocks, and that at least one of B_1 and B_2 , say B_1 , is not a clique. Let v be the cut vertex of B_1 and x be a non-cut vertex in B_2 . By Lemma 6.1, $Z_c(G[B_1]; \{v\}) \leq |B_1| - 2$, so there are two vertices u and w in B_1 such that $V \setminus \{u, w\}$ is a connected forcing set of G . Moreover, some non-cut neighbor of x in B_2 can force x in the first timestep; thus $V \setminus \{u, w, x\}$ is a connected forcing set of G , a contradiction.

Finally, suppose G has no trivial outer blocks, and that both B_1 and B_2 are cliques. By Proposition 4.6 and Lemma 4.3, the set excluding one non-cut vertex from each of B_1 and B_2 is a minimum connected forcing set of G . This is the case illustrated in Figure 6.4. □

Claim 6.7. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) = 1$, a single nontrivial block B , and non-cut vertex $x \in B$. Then any pendant tree T of G is either composed of one or more leaves attached to a vertex of B , or of two or more leaves joined to a vertex of B by a path.*

Proof. Let T be a pendant tree of G attached to some vertex $v \in B$. If T has two leaves ℓ_1 and ℓ_2 which are not adjacent to the same vertex, then $V \setminus \{\ell_1, \ell_2, x\}$ is a connected forcing set; thus, all leaves of T are adjacent to the same vertex. T also cannot be a pendant path of length more than 1, since then $V \setminus \{\ell, w, x\}$ is a connected forcing set, where ℓ is the leaf of the pendant path and w is the neighbor of ℓ . Thus, T is composed of one or more leaves attached to v , or of two or more leaves joined to v by a path. \square

Claim 6.8. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) = 1$, and a single nontrivial block, which is either an inner block or an outer block and a clique. Then, G belongs to the family of graphs described in Figure 6.5.*

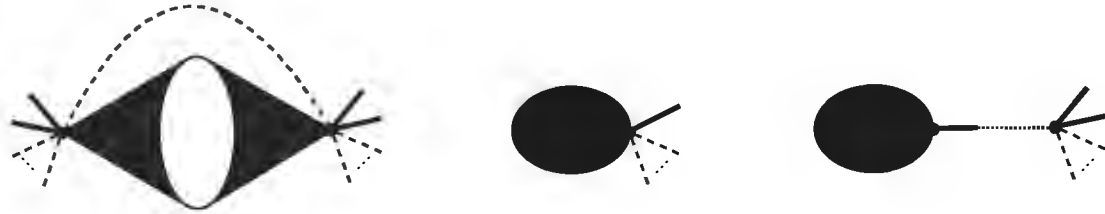


Figure 6.5 : White oval represents an independent set of size at least 1, shaded ovals represent cliques of size at least 3. Shaded regions represent all possible edges being present. Dotted line represents a path of possibly zero length, thick lines represent mandatory single edges, dashed straight lines represent an arbitrary number (possibly zero) of single edges. If white oval consists of a single vertex, dashed curved line represents a mandatory single edge; otherwise, it represents a possibly non-existent single edge.

Proof. Let B be the nontrivial block of G and suppose first that B is an inner block. If B has at least 3 cut vertices v_1 , v_2 , and v_3 (which are by definition feasible vertices), then $V \setminus \{\ell(v_1), \ell(v_2), \ell(v_3)\}$ is a connected forcing set, a contradiction. Thus B has 2 cut vertices v_1 and v_2 . Let T_1 and T_2 be the pendant trees attached to v_1 and v_2 ,

respectively, and let x be some non-cut vertex of B . By Claim 6.7, for $i \in \{1, 2\}$, T_i is composed of one or more leaves attached to v_i , or of two or more leaves joined to v_i by a path.

If at least one of T_1 and T_2 , say T_1 , is composed of two or more leaves joined to v_1 by a path, let ℓ_1 be a leaf in T_1 , ℓ_2 be a leaf in T_2 , and x be a non-cut vertex of B . Then $V \setminus \{\ell_1, \ell_2, x\}$ is a connected forcing set, since ℓ_1 can be forced in the first timestep by its neighbor, then any neighbor of x (possibly except v_2) can force x , and then ℓ_2 can be forced by its neighbor; this is a contradiction.

If at least one of T_1 and T_2 , say T_1 , consists of a single leaf ℓ_1 , and ℓ_2 is a leaf in T_2 , then $V \setminus \{\ell_1, v_1, \ell_2\}$ is a connected forcing set since some non-cut neighbor of v_1 in B (possibly except v_2) can force v_1 in the first timestep, and then ℓ_1 and ℓ_2 can be forced by their neighbors; this is a contradiction.

Thus, T_1 and T_2 each consist of two or more leaves. Let ℓ_1 and ℓ_2 be leaves in T_1 and T_2 , respectively. If $G[B \setminus \{v_1, v_2\}]$ is not an empty graph, then there is an edge between two vertices x and y in $B \setminus \{v_1, v_2\}$. Then, $V \setminus \{\ell_1, \ell_2, x\}$ is a connected forcing set since y can force x in the first timestep, and then ℓ_1 and ℓ_2 can be forced by v_1 and v_2 (note that this set is connected since x is not a cut vertex of G); this is a contradiction.

Now, suppose $G[B \setminus \{v_1, v_2\}]$ is an empty graph. Then both v_1 and v_2 must be adjacent to every vertex in $B \setminus \{v_1, v_2\}$, since otherwise $G[B]$ would not be biconnected; v_1 and v_2 could also possibly be adjacent to each other, and if $B \setminus \{v_1, v_2\}$ consists of a single vertex, then v_1 and v_2 must necessarily be adjacent in order for $G[B]$ to be biconnected. Moreover, it is easy to see that $V \setminus \{\ell_1, \ell_2\}$ is a connected forcing set of G . This set is also minimum, since by Lemma 4.3, v_1 and v_2 are contained in every connected forcing set of G , and a set excluding 3 or more vertices of $V \setminus \{v_1, v_2\}$ will

exclude at least two neighbors of at least one of v_1 and v_2 , and will therefore not be forcing. This family of graphs is illustrated in Figure 6.5, left.

Now suppose that B is an outer block and a clique. Let v be the cut vertex of B , T be the pendant tree attached to v , ℓ be a leaf in T , and x be some non-cut vertex of B . By Claim 6.7, T is either composed of one or more leaves attached to v , or of two or more leaves joined to v by a path. In both cases, by Proposition 4.6 and Lemma 4.3, $V \setminus \{x, \ell\}$ is a minimum connected forcing set of G . These two cases are illustrated in Figure 6.5 middle and right, respectively. \square

Claim 6.9. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) = 1$, and a single nontrivial block, which is an outer block and not a clique. Then, G belongs to the family of graphs described in Figure 6.6.*



Figure 6.6 : Shaded region represents a clique of size at least 3; thick lines represent mandatory single edges, dashed lines represent an arbitrary number (possibly zero) of single edges, such that the vertex outside the shaded oval is not adjacent to every vertex inside the shaded oval.

Proof. Let B be the nontrivial block of G and v be the cut vertex of B . By Claim 6.7, the pendant tree T attached to v must either be composed of one or more leaves attached to v , or two or more leaves joined to v by a path. If T consists of two or more leaves joined to v by a path, then by Lemma 6.1, $Z_c(G[B]; \{v\}) \leq |B| - 2$, so there are two vertices x and y in B such that $V \setminus \{x, y\}$ is a connected forcing set.

Moreover, a leaf ℓ in T can be forced by its neighbor in the first timestep; it follows that $V \setminus \{x, y, \ell\}$ is a connected forcing set of G , a contradiction.

Thus, T consists of one or more leaves attached to v . Let ℓ be one of these leaves. Suppose $G[B] - v$ has no separating set; then $G[B] - v$ is a clique. Since $G[B]$ is not a clique, there must be some vertex $x \in B \setminus \{v\}$ which is not adjacent to v . Let $y \neq x$ be another vertex in $B \setminus \{v\}$. If T consists of a single leaf, then $V \setminus \{\ell, v, y\}$ is a connected forcing set of G , since x can force y in the first timestep, then any neighbor of v in $B \setminus \{v\}$ can force v , and then v can force ℓ . If T contains two or more leaves, then $V \setminus \{\ell, x\}$ is a connected forcing set. Moreover, this set is minimum, since by Lemma 4.3, every connected forcing set contains v and all-but-one leaves attached to v , and if a set excludes two or more vertices from $B \setminus \{v\}$, then any colored neighbor of these vertices would always have at least two uncolored neighbors. This family of graphs is illustrated in Figure 6.6.

Now suppose $G[B] - v$ does have a separating set. Note that since $G[B]$ is bi-connected, $\kappa(G[B] - v) \geq 1$. If $\kappa(G[B] - v) = 1$, let u be a vertex such that $\{u, v\}$ is a separating set of G and let $x \in B$ be a non-cut vertex of $G - u$; let C be the component of $G - \{u, v\}$ containing x . Then $V \setminus \{\ell, u, x\}$ is a connected forcing set, since u can be forced by some neighbor of u in a component of $G - \{u, v\}$ other than C in the first timestep, then x can be forced by any of its neighbors except v , and then v can force ℓ . Thus, $Z_c(G) \leq n - 3$, a contradiction.

If $\kappa(G[B] - v) \geq 2$ and $d_{G[B]}(v) = 2$, let u and w be the neighbors of v in B . Suppose first that there is some vertex $x \in B \setminus \{u, v, w\}$ such that at least one of $\{u, x\}$ and $\{w, x\}$, say $\{u, x\}$, is a separating set of G . Let C be a component of $G - \{u, x\}$ which does not contain v . Let y be a non-cut vertex of $G - x$ in C . Then $V \setminus \{x, y, \ell\}$ is a forcing set of G , since some neighbor of x (except v) in a component

of $G - \{u, x\}$ other than C can force x in the first timestep, then any neighbor of y can force y , and then v can force ℓ . This set is also connected since y is a non-cut vertex of $G[B] - x$, which is connected. Thus, $Z_c(G) \leq n - 3$, a contradiction.

Now suppose that for any $x \in B \setminus \{u, v, w\}$, neither $\{u, x\}$ nor $\{w, x\}$ is a separating set of G . If u is not adjacent to w , let y be any neighbor of u in $B \setminus \{u, v, w\}$. Then $V \setminus \{\ell, w, y\}$ is a connected forcing set, since u can force y in the first timestep, then any neighbor of w (except v) can force w , and then v can force ℓ . Note that this set is connected, since by assumption, $\{w, y\}$ is not a separating set of G . If u is adjacent to v , suppose there is a vertex $y \in B \setminus \{u, v, w\}$ which is not adjacent to at least one of u and w , say $y \not\sim u$. Then $V \setminus \{\ell, w, y\}$ is a connected forcing set, since u can force w in the first timestep, then any neighbor of y can force y , and then v can force ℓ . Now suppose every vertex in $B \setminus \{u, v, w\}$ is adjacent to both u and w . Since $G[B] - v$ is not a clique, there must be some vertices x and y in $B \setminus \{u, v, w\}$ which are not adjacent to each other. Then $V \setminus \{\ell, w, y\}$ is a connected forcing set, since x can force w in the first timestep, then any neighbor of y can force y , and then v can force ℓ . In all these cases, it follows that $Z_c(G) \leq n - 3$, a contradiction.

If $\kappa(G[B] - v) = 2$ and $d_{G[B]}(v) \geq 3$, let $\{u, w\}$ be a separating set of $G[B] - v$, and let x be a non-cut vertex of $G[B] - \{u, v\}$ in some component C of $G[B] - \{u, v, w\}$. Then $V \setminus \{u, x, \ell\}$ is a forcing set of G , since any neighbor of u in a component of $G[B] - \{u, v, w\}$ different from C can force u in the first timestep, then any neighbor of x (except v) can force x , and then v can force ℓ . This set is also connected: $G[B] - \{u, v\}$ is connected since $\kappa(G[B] - v) = 2$, $G[B] - \{u, v, x\}$ is connected since x is a non-cut vertex of $G[B] - \{u, v\}$, $G[B] - \{u, x\}$ is connected since $d_{G[B]}(v) \geq 3$ and hence v 's neighbors in B cannot be only u and x , and $G - \{u, x, \ell\}$ is connected since ℓ is a leaf. Thus, $Z_c(G) \leq n - 3$, a contradiction.

If $\kappa(G[B] - v) \geq 3$ and $d_{G[B]}(v) \geq 3$, let x and y be two non-adjacent vertices in $B \setminus \{v\}$, and let $z \in B \setminus \{v\}$ be a neighbor of y . Then $V \setminus \{\ell, x, z\}$ is a forcing set, since y can force z in the first timestep, then any neighbor of x except v can force x , and then v can force ℓ . This set is also connected, since $\kappa(G[B] - v) \geq 3$ and since $d_{G[B]}(v) \geq 3$. Thus, $Z_c(G) \leq n - 3$, a contradiction. \square

Claim 6.10. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) = 1$, and no nontrivial blocks. Then, G is one of the graphs described in Figure 6.7.*



Figure 6.7 : Left: two stars each with at least 2 leaves joined by a path of length at least 1. Middle: Pendant attached to a leaf of a star with at least 3 leaves. Right: P_3 .

Proof. Since G has only trivial blocks, G is a tree; thus, by Theorem 5.2, $Z_c(G) = R_2 \cup R_3 \cup \mathcal{L}$. If G has three or more vertices in R_3 , then there are at least 3 vertices not in \mathcal{L} , so $Z_c(G) \leq n - 3$, a contradiction.

If G has two vertices u and v in R_3 , then all other vertices must belong to a path connecting u and v , or to pendant paths attached to u or v . By a similar argument as in Claim 6.7, the length of any pendant path attached to u or v must be 1. By Theorem 5.2, this condition is also sufficient to guarantee that $Z_c(G) = n - 2$. This is the family of graphs illustrated in Figure 6.7, left.

If G has one vertex v in R_3 , then all other vertices must belong to pendant paths attached to v . If all pendant paths have length 1, then G is a star and $Z_c(G) = n - 1$. If more than one pendant path has length greater than 1, or if any pendant path

has length greater than 2, by a similar argument as in Claim 6.7, it follows that $Z_c(G) \leq n - 3$. Thus, one pendant path must have length 2, and all other pendant paths must have length 1. This is the family of graphs illustrated in Figure 6.7, middle.

If G has no vertices in R_3 , then G is a path, and $Z_c(G) = n - 2$ if and only if $G \simeq P_3$. This is the graph illustrated in Figure 6.7, right. \square

Claim 6.11. *If G is a graph with $\kappa(G) \geq 2$ and S is a minimum separating set of G such that $G - S$ has three or more components, at least one of which is nontrivial, then $Z_c(G) \leq n - 3$.*

Proof. Let B_1, \dots, B_k be the components of $G - S$, $k \geq 3$; let s_1 and s_2 be vertices in S . Since S is minimum, each vertex of S is connected to at least one vertex of every component of $G - S$. Without loss of generality, let B_1 be a nontrivial component of $G - S$. Note that since $G[B_1]$ is connected and nontrivial, it has at least two non-cut vertices. If s_1 is adjacent to exactly one vertex of B_1 , let x_1 be a non-cut vertex of $G[B_1]$ different from the neighbor of s_1 in B_1 ; otherwise, if s_1 is adjacent to two or more vertices of B_1 , let x_1 be an arbitrary non-cut vertex of $G[B_1]$. If B_2 is a trivial block, let x_2 be the vertex of B_2 ; if B_2 is nontrivial and if s_1 is adjacent to exactly one vertex of B_2 , let x_2 be a non-cut vertex of $G[B_2]$ different from the neighbor of s_1 in B_2 , and if s_1 is adjacent to two or more vertices of B_2 , let x_2 be an arbitrary non-cut vertex of $G[B_2]$. In every case, $V \setminus \{s_2, x_1, x_2\}$ is a forcing set, since any neighbor of s_2 in B_3 can force s_2 in the first timestep, then any neighbor of x_1 in B_1 can force x_1 , and then any neighbor of x_2 can force x_2 . This set is also connected, since each of the graphs $G[B_1] - x_1, G[B_2] - x_2, G[B_3], \dots, G[B_k]$ is connected, s_1 is connected to each of these graphs, and all other vertices of $S \setminus \{s_2\}$ are connected to some vertices

in $G[B_3], \dots, G[B_k]$. Thus, $Z_c(G) \leq n - 3$. \square

Claim 6.12. *Let G be a graph with $Z_c(G) = n - 2$ and let S be a minimum separating set of G such that $G - S$ has only trivial components. Then every trivial component of $G - S$ must be adjacent to every vertex in S ; moreover, any connected forcing set of G excludes at most one trivial component of $G - S$.*

Proof. Let v be a vertex that is a trivial component of $G - S$. If v is not adjacent to some vertex $u \in S$, then $S \setminus \{u\}$ would be a smaller separating set of G than S . Let R be a connected forcing set of G and suppose R excludes two vertices v_1 and v_2 which are trivial components of $G - S$. Since v_1 and v_2 are only adjacent to vertices in S , and since every vertex in S has at least two uncolored neighbors (namely v_1 and v_2), no vertex in S would be able to force v_1 and v_2 . Thus, any connected forcing set can exclude at most one trivial component of $G - S$. \square

Claim 6.13. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) \geq 2$, and let S be a minimum separating set of G such that $G - S$ has only trivial components. Then G is one of the graphs described in Figure 6.8.*

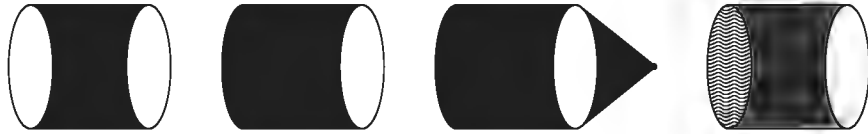


Figure 6.8 : Ovals represent sets of vertices, each of size at least two. Shaded regions represent all possible edges being present within a set of vertices or between sets of vertices; white regions represent no edges being present. Region with wave pattern represents a set of vertices which induces a graph H which has no isolated vertices and has zero forcing number $|V(H)| - 2$.

Proof. Let C be the set of vertices which are trivial components of $G - S$. By Claim 6.12, every vertex in C is adjacent to every vertex in S . Suppose for contradiction that $Z(G[S]) < |S| - 2$, and let Z be a minimum zero forcing set of $G[S]$. Then $Z \cup C$ is a connected forcing set of G , since any vertex in Z which forces a vertex in $G[S]$ can also force the same vertex in G . Thus, $Z_c(G) \leq |Z| + |C| < |S| - 2 + |C| = n - 2$, a contradiction. Thus, $Z(G[S]) \geq |S| - 2$.

If $Z(G[S]) = |S|$, then $G[S]$ is an empty graph, and hence G is a complete bipartite graph with parts C and S . Then, any set containing all-but-one vertices of C and all-but-one vertices of S is connected and forcing (note that $|C| \geq 2$ and $|S| \geq 2$). This set is also minimum, since a set excluding more than one vertex from one (or both) of C and S is not forcing. This family of graphs is illustrated in Figure 6.8, left.

If $Z(G[S]) = |S| - 1$, then $G[S]$ is the disjoint union of a nontrivial clique and zero or more isolated vertices. If $G[S]$ has at most one isolated vertex, then any set containing all-but-one vertices of C and all-but-one vertices of S is connected and forcing. This set is also minimum, since by Claim 6.12, a connected forcing set R can exclude at most one vertex of C ; if R excludes one vertex of C , then it cannot exclude two or more vertices of S , since then no vertex will be able to force them. Similarly, if R contains all vertices of C , then it cannot exclude three or more vertices of S , since then at least two of them will belong to the nontrivial clique in $G[S]$, and no vertex will be able to force them. Thus $Z_c(G) = n - 2$; this family of graphs is illustrated in Figure 6.8, middle-left and middle-right.

If $G[S]$ is the disjoint union of a nontrivial clique and two or more isolated vertices, then let u and x be isolated vertices in $G[S]$, v and y be vertices in the nontrivial clique of $G[S]$, and w be a vertex in C . Then $V \setminus \{u, v, w\}$ is a connected forcing set,

since x can force w in the first timestep, then y can force v , and then w can force u . Thus $Z_c(G) \leq n - 3$, a contradiction.

Finally, if $Z(G[S]) = |S| - 2$, then $G[S]$ is one of the graphs in Corollary 6.1. Let Z be an arbitrary minimum zero forcing set of $G[S]$, let $\{z_1, z_2\} = S \setminus Z$, and let x be a vertex in C . If $G[S]$ has an isolated vertex v , then v must be contained in Z . Then $V \setminus \{z_1, z_2, x\}$ is a connected forcing set, since v can force x in the first timestep, and then z_1 and z_2 can be forced by the same vertices which force them in $G[S]$. Thus, $G[S]$ does not have isolated vertices. Moreover, $V \setminus \{z_1, z_2\}$ is a connected forcing set of G , since z_1 and z_2 can be forced in G by the same vertices which force them in $G[S]$; we claim that this set is also minimum. To see why, suppose there is a connected forcing set R which excludes three or more vertices of G . By Claim 6.12, R can exclude at most one vertex of C . If R excludes one vertex x of C and two or more vertices of S , then no vertex in C can force another vertex until all-but-one vertices in S are forced (because until then, all vertices in C are adjacent to two or more uncolored vertices in S). Thus, the first force must be performed by a vertex y in S . This means y has a single uncolored neighbor, which must be x . Then, all neighbors of y in S are contained in R . Let R' be the set obtained by adding x and all-but-two vertices in $S \setminus R$ to R . R' is also connected and forcing, and there is a chronological list of forces where both vertices not in R' are forced by vertices of S . Thus $R' \cap S$ is a zero forcing set of $G[S]$ of size $|S| - 2$. However, y is a non-isolated vertex in $G[S]$, which is in $R' \cap S$ and all of whose neighbors are in $R' \cap S$. Therefore, $R' \cap S \setminus \{z\}$ is also a zero forcing set of $G[S]$, where z is a neighbor of y in S ; this contradicts the assumption that $Z(G[S]) = |S| - 2$. Similarly, if R excludes no vertices of C , then it cannot exclude three or more vertices of S , since then $S \cap R$ would be a zero forcing set of $G[S]$ of size at most $|S| - 3$, a contradiction. Thus, $Z_c(G) = n - 2$; this family

of graphs is illustrated in Figure 6.8, right. \square

Claim 6.14. *Let G be a graph with $Z_c(G) = n - 2$ and $\kappa(G) = 2$; let S be a minimum separating set of G such that $G - S$ has exactly two components, at least one of which is nontrivial. Then each component of $G - S$ is a clique, and each vertex from each component of $G - S$ is adjacent to every vertex in S .*

Proof. Let s_1 and s_2 be the vertices of S , and let B_1 and B_2 be the components of $G - S$. Let $\mathcal{I} = \{(1, 1), (2, 1)\}, \{(1, 1), (2, 2)\}, \{(1, 2), (2, 1)\}, \{(1, 2), (2, 2)\}$ and $\mathcal{J} = \{(1, 1), (1, 2)\}, \{(2, 1), (2, 2)\}$.

Suppose first that there exists a set $I \in \mathcal{I}$ such that for each $(i, j) \in I$, $G[B_i \cup \{s_j\}]$ has no cut vertices. Without loss of generality, let $I = \{(1, 1), (2, 1)\}$, i.e., suppose $G[B_1 \cup \{s_1\}]$ and $G[B_2 \cup \{s_1\}]$ have no cut vertices. Suppose also that s_2 is not adjacent to some vertex of $B_1 \cup B_2$, say $x \in B_2$; then, B_2 must be a nontrivial component. Let y be a neighbor of x in B_2 and let $v \in B_1$ be a non-cut vertex of $G - s_2$. Then, $V \setminus \{v, s_2, y\}$ is a forcing set, since x can force y in the first timestep, then any neighbor of s_2 in B_2 can force s_2 , and then any neighbor of v can force v . This set is also connected, since $G - s_2$ is connected, y is not a cut vertex of $G[B_2 \cup \{s_1\}]$, and v is not a cut vertex of $G[B_1 \cup \{s_1\}]$. This contradicts $Z_c(G) = n - 2$, so s_2 must be adjacent to every vertex in B_2 . Hence, $G[B_2 \cup \{s_2\}]$ has no cut vertices (since $G[B_2]$ is connected), and so by the same argument as above, it follows that s_1 is also adjacent to every vertex in B_2 . Similarly, s_1 and s_2 are adjacent to every vertex in B_1 . Now suppose B_2 is not a clique; then, B_2 must have at least three vertices. Let x and y be two non-adjacent vertices in B_2 ; let z be a neighbor of x in B_2 , and let v be any vertex in B_1 . Then, $V \setminus \{y, z, v\}$ is a connected forcing set, since x can force z in the first timestep, then any neighbor of y in B_2 can force y , and then any neighbor of

v can force v . This set is also connected, since every vertex in B_1 and B_2 is adjacent to S , and every vertex in S is adjacent to x . This is a contradiction, so B_2 is a clique; similarly, B_1 is a clique.

Now suppose that there does not exist a set $I \in \mathcal{I}$ such that for each $(i, j) \in I$, $G[B_i \cup \{s_j\}]$ has no cut vertices. Equivalently, there exists a set $J \in \mathcal{J}$ such that for each $(i, j) \in J$, $G[B_i \cup \{s_j\}]$ has cut vertices. Without loss of generality, let $J = \{(2, 1), (2, 2)\}$, i.e., $G[B_2 \cup \{s_1\}]$ and $G[B_2 \cup \{s_2\}]$ have cut vertices. Hence, $G[B_2]$ has cut vertices, since $G[B_2]$ is connected. At least one of s_1 and s_2 must be adjacent to a non-cut vertex of every outer block of $G[B_2]$, since otherwise the cut vertex of such a block would be a cut vertex of G . Note also that if s_1 or s_2 , say s_1 , is adjacent to a non-cut vertex of every outer block of $G[B_2]$, then $G[B_2 \cup \{s_1\}]$ would not have any cut vertices. Thus, there is an outer block D_1 of $G[B_2]$ such that s_1 is adjacent to a non-cut vertex d_1 of D_1 and s_2 is not adjacent to any non-cut vertex of D_1 , and there is an outer block D_2 of $G[B_2]$ such that s_2 is adjacent to a non-cut vertex d_2 of D_2 and s_1 is not adjacent to any non-cut vertex of D_2 .

Suppose s_1 is adjacent to a single vertex of B_2 ; this must be the vertex d_1 defined above. Let $v \in B_1$ be a non-cut vertex of $G - s_1$. Then, $V \setminus \{v, s_1, d_1\}$ is a forcing set, since any neighbor of d_1 in B_2 can force d_1 in the first timestep, then d_1 can force s_1 , and then any neighbor of v can force v . This set is also connected, since $G - s_1$ is connected, and v and d_1 are non-cut vertices of $G - s_1$. Thus, $Z_c(G) \leq n - 3$, a contradiction.

Now suppose s_1 is adjacent to two or more vertices of B_2 . Let $v \in B_1$ be a non-cut vertex of $G - s_2$. Then, $V \setminus \{v, s_2, d_1\}$ is a forcing set, since d_2 can force s_2 in the first timestep, then d_1 can be forced by any of its neighbors in B_2 , and then any neighbor of v can force v . This set is also connected, since $G[B_1] - v$ is connected, $G[B_2] - d_1$

is connected, s_1 is adjacent to some vertex in B_1 other than v , and s_1 is adjacent to some vertex in B_2 other than d_1 . Thus, $Z_c(G) \leq n - 3$, a contradiction. \square

Claim 6.15. *Let G be a graph with $Z_c(G) = n - 2$ and $\kappa(G) = 3$; let S be a minimum separating set of G such that $G - S$ has exactly two components, at least one of which is nontrivial. Then each component of $G - S$ is a clique, and each vertex from each component of $G - S$ is adjacent to every vertex in S .*

Proof. Suppose first that at least one of B_1 and B_2 , say B_2 , is not a clique; then B_2 is a nontrivial component. Suppose also that no two vertices of B_2 form a separating set of $G - s_1$. Let x and y be two nonadjacent vertices in B_2 , and let z be a neighbor of x in B_2 . Then, $V \setminus \{s_1, y, z\}$ is a forcing set, since any neighbor of s_1 in B_1 can force s_1 in the first timestep, then x can force z , and then any neighbor of y can force y . This set is also connected, since $G - s_1$ is connected, and by assumption $\{y, z\}$ is not a separating set of $G - s_1$.

Now suppose that two vertices t_1 and t_2 in B_2 form a separating set of $G - s_1$. Let D be a component of $G - \{s_1, t_1, t_2\}$ which does not contain s_2 and s_3 . Note that s_1 must be adjacent to some vertex d in D , since otherwise $\{t_1, t_2\}$ would be a separating set of G . Let $v \in B_1$ be a non-cut vertex of $G - \{s_1, s_2\}$. Then, $V \setminus \{s_1, s_2, v\}$ is a forcing set, since d can force s_1 in the first timestep, then any neighbor of s_2 in B_2 can force s_2 , and then any neighbor of v can force v . This set is also connected, since $G - \{s_1, s_2\}$ is connected, and v is a non-cut vertex of $G - \{s_1, s_2\}$.

In both cases, it follows that $Z_c(G) \leq n - 3$, a contradiction; thus, B_2 is a clique, and similarly, B_1 is a clique. Now suppose that some vertex in S , say s_1 , is not adjacent to some vertex in B_1 or B_2 , say $x \in B_2$; note that B_2 must then be a nontrivial component. Let v be any vertex in B_1 .

If $|B_2| = 2$, let $B_2 = \{x, y\}$. Then, s_1 is adjacent only to y , so both s_2 and s_3 must be adjacent to x , since otherwise x will have fewer than three neighbors (contradicting $\kappa(G) = 3$). Then, $V \setminus \{s_1, y, v\}$ is a forcing set, since x can force y in the first timestep, then any neighbor of s_1 in B_2 can force s_1 , and then any neighbor of v can force v . This set is also connected, since s_2 and s_3 are both adjacent to x , and if B_1 is not a trivial component, then at least one of s_2 and s_3 is adjacent to a vertex of B_1 other than v .

If $|B_2| = 3$, let $B_2 = \{x, y_1, y_2\}$. Note that any pair of vertices in B_2 must collectively have at least two neighbors in S , since otherwise their single neighbor and the other vertex in B_2 form a separating set of G . If s_1 is adjacent to a single vertex in B_2 , let y be that vertex. If s_1 is adjacent to both y_1 and y_2 , and if x is adjacent to both s_2 and s_3 , let y be y_1 . If s_1 is adjacent to both y_1 and y_2 , and if x is adjacent to a single vertex $s \in \{s_1, s_2\}$, and if $S \setminus \{s_1, s\}$ has a single neighbor $z \in B_2$, let y be $B_2 \setminus \{x, z\}$; if $S \setminus \{s_1, s\}$ has multiple neighbors in B_2 , let y be y_1 . In each of these cases, $V \setminus \{s_1, y, v\}$ is a forcing set, since x can force y in the first timestep, then any neighbor of s_1 in B_2 can force s_1 , and then any neighbor of v can force v . This set is also connected, since $G[B_2] - y$ is connected, $G[B_1] - v$ is connected, s_2 and s_3 are each adjacent to at least one vertex in $G[B_2] - y$ (for each choice of y above), and at least one of s_2 and s_3 is adjacent to a vertex of $G[B_1] - v$ (if B_1 is not a trivial component).

If $|B_2| \geq 4$, let y be a vertex in B_2 which is different from x , and — if one or both of s_2 or s_3 have a single neighbor in B_2 — is different from those neighbors. Then, $V \setminus \{s_1, y, v\}$ is a connected forcing set by the same reasoning as above.

In all cases, we reach a contradiction, so it follows that each vertex of B_2 is adjacent to each vertex of S . Similarly, we conclude that each vertex of B_1 is adjacent to each

vertex of S . □

Claim 6.16. *Let G be a graph with $Z_c(G) = n - 2$ and $\kappa(G) \geq 4$; let S be a minimum separating set of G such that $G - S$ has exactly two components, at least one of which is nontrivial. Then each component of $G - S$ is a clique, and each vertex from each component of $G - S$ is adjacent to every vertex in S .*

Proof. Let B_1 and B_2 be the components of $G - S$, and suppose for contradiction that some vertex in B_1 or B_2 , say $x \in B_1$ is not adjacent to some vertex in S , say s_1 ; note that B_1 must then be a nontrivial component. Let y be a neighbor of x in B_1 and let z be a vertex in B_2 . Then, $V \setminus \{y, z, s_1\}$ is a forcing set of G , since x can force y in the first timestep, then some neighbor of s_1 in B_1 can force s_1 , and then any neighbor of z can force z . This set is also connected since $\kappa(G) \geq 4$; thus, $Z_c(G) \leq n - 3$, a contradiction. Therefore, each vertex from each component of $G - S$ is adjacent to every vertex in S .

Now suppose for contradiction that some component of $G - S$, say B_1 , is not a clique. Note that B_1 must then have at least 3 vertices, since if B_1 is a trivial component or has two vertices which are connected, then B_1 is a clique. Let x and y be vertices in B_1 which are not adjacent, and let z be a neighbor of x in B_1 ; let w be a vertex in B_2 . Then, $V \setminus \{y, z, w\}$ is a forcing set of G , since x can force z in the first timestep, then some neighbor of y in B_1 can force y , and then any neighbor of w can force w . This set is also connected since $\kappa(G) \geq 4$; thus, $Z_c(G) \leq n - 3$, a contradiction. Therefore, each component of $G - S$ is a clique. □

Claim 6.17. *Let G be a graph with $Z_c(G) = n - 2$, $\kappa(G) \geq 2$ and let S be a minimum separating set of G such that $G - S$ has exactly two components, at least one of which is nontrivial. Then G is one of the graphs described in Figure 6.9.*



Figure 6.9 : Ovals represent sets of vertices. Shaded regions represent all possible edges being present within a set of vertices or between sets of vertices; white regions represent no edges being present. Region with wave pattern represents a set of vertices which induces a graph H which has no isolated vertices and has zero forcing number $|V(H)| - 2$. Smaller ovals have at least one vertex, larger ovals have at least two vertices.

Proof. Let B_1 and B_2 be the components of $G - S$, where B_1 is a nontrivial component. By Claims 6.14, 6.15, and 6.16, B_1 and B_2 are cliques, and every vertex in B_1 and B_2 is adjacent to every vertex in S . By the same argument as in Claim 6.13, $Z(G[S]) \geq |S| - 2$.

If $Z(G[S]) = |S|$, then $G[S]$ is an empty graph, and any set excluding a single vertex from S and a single vertex from B_1 is connected and forcing (note that $|S| \geq 2$). This set is also minimum, since if R is a set which excludes two or more vertices from B_1 , S , or B_2 , or excludes one vertex from each of B_1 , S , and B_2 , then every vertex in R will have at least two neighbors not in R , and hence R will not be forcing. This family of graphs is illustrated in Figure 6.9, left.

If $Z(G[S]) = |S| - 1$, then $G[S]$ is the disjoint union of a clique and zero or more isolated vertices. If $G[S]$ has at most one isolated vertex, then any set excluding a single vertex from S and a single vertex from B_1 is connected and forcing. This set is also minimum since if R is a set which excludes two or more vertices from B_1 , S ,

or B_2 , or excludes one vertex from each of B_1 , S , and B_2 , then every vertex in R will have at least two neighbors not in R , and hence R will not be forcing. This family of graphs is illustrated in Figure 6.9, middle-left and middle-right.

If $G[S]$ is the disjoint union of a clique and two or more isolated vertices, then let x_1 and x_2 be isolated vertices in $G[S]$, v_1 and v_2 be vertices in the nontrivial clique of $G[S]$, u_1 be a vertex in B_1 . Then $V \setminus \{u_1, v_1, w_1\}$ is a connected forcing set, since x_2 can force u_1 in the first timestep, then v_2 can force v_1 , and then any neighbor of x_1 can force x_1 . Thus $Z_c(G) \leq n - 3$, a contradiction.

Finally, if $Z(G[S]) = |S| - 2$, then $G[S]$ is one of the graphs in Corollary 6.1. Let Z be an arbitrary minimum zero forcing set of $G[S]$, and let $\{z_1, z_2\} = S \setminus Z$. By a similar argument as in Claim 6.13, $G[S]$ does not have isolated vertices; moreover, $V \setminus \{z_1, z_2\}$ is a connected forcing set of G . We claim that this set is also minimum; to see why, suppose there is a connected forcing set R which excludes three or more vertices of G . If R excludes three or more vertices of $B_1 \cup B_2$, then two of them are in the same clique component of $G - S$, and can therefore not be forced by any of their neighbors. For the same reason, if R excludes two vertices of $B_1 \cup B_2$, then one of these vertices must be in B_1 and the other must be in B_2 ; however, if R also excludes one or more vertex of S , then every vertex of G will have at least two uncolored neighbors, and no forcing will be possible. By a similar argument as in Claim 6.13, we also reach a contradiction if R excludes one vertex of $B_1 \cup B_2$ and two or more vertices of S , or if R excludes no vertices of $B_1 \cup B_2$ and three or more vertices of S . Thus, $Z_c(G) = n - 2$; this family of graphs is illustrated in Figure 6.9, right. \square

Since each of the graphs described in Figures 6.4–6.9 has connected forcing number $n - 2$, this concludes the proof of Theorem 6.6. \square

The statement of Theorem 6.6 can be rewritten in a similar format as the statement of Theorem 6.5; however, I chose to express my results using explicit diagrams in order to make it easier to visualize the structure of the graphs in question. Due to the constant number of equivalence classes of vertices in each of the graphs in Figures 6.4–6.9 (or in their complements, according to Theorem 6.5), it is readily verifiable that a graph in this family is efficiently recognizable; this is stated formally below.

Observation 6.4. *It can be recognized whether a graph G belongs to the family of graphs given in Theorem 6.6 in $O(n^2)$ time.*

Chapter 7

Enumeration problems related to zero forcing

In this chapter, I focus on several enumeration problems associated with zero forcing and connected forcing. I investigate the cardinality and other properties of the set of minimum connected forcing sets of a graph G , the set of connected forcing sets of G which have a given size, and the set of all connected forcing sets of G ; analogous results for zero forcing are presented as well. In particular, I identify when some of these sets grow exponentially with the order of the graph, use the set of all forcing sets of certain graphs to define greedoids and matroids, and define zero forcing and connected forcing polynomials which count the number of distinct zero forcing and connected forcing sets of a given size.

7.1 Exponential quantities associated with zero forcing

In this section, I show that several parameters related to zero forcing can grow exponentially with the size of a graph. These results imply that in general, any computation based on enumeration of these parameters will not be efficient. For example, one way to determine the minimum and maximum propagation times of a graph G is to find the propagation times of each of its minimum zero forcing sets. The following result implies that this approach could take exponential time.

Proposition 7.1. *A graph can have exponentially-many distinct minimum zero forcing sets and exponentially-many distinct minimum connected forcing sets.*

Proof. Let P_k be a path with vertex set $\{v_1, \dots, v_k\}$. Let G be the graph obtained by appending two pendants, a_i and b_i , to v_i , $1 \leq i \leq k$. Since $Z(G) \geq L(G)/2$ and since $\{a_1, \dots, a_k\}$ is a zero forcing set, $Z(G) = k$. Since $Z_c(G) \geq L(G)$ and since $\{a_1, \dots, a_k, v_1, \dots, v_k\}$ is a connected forcing set, $Z_c(G) = 2k$. Each set of the form $\{c_1, \dots, c_k\}$ is a zero forcing set, where $c_i \in \{a_i, b_i\}$ for $1 \leq i \leq k$; likewise, each set of the form $\{c_1, \dots, c_k, v_1, \dots, v_k\}$ is a connected forcing set. Thus, G has $\Omega(2^k)$ distinct zero forcing sets and connected forcing sets. \square

Let R be a minimum zero forcing set of a graph $G = (V, E)$, and let G_S be the graph obtained by adding a vertex v^* to G and connecting it to all vertices in a set $S \subset V$ (i.e., G_S is a generalized vertex join of G). A question of interest in this scenario is: when does R remain a zero forcing set of G_S ? A sufficient condition to assure that R is a zero forcing set of G_S is for S to be a set of terminals of forcing chains associated with R (since then v^* would not interfere with any forces performed between vertices of G , and could be forced in the last timestep). The next result shows that in general, one could not efficiently enumerate all sets $S \subset V$ which are terminals of forcing chains associated with R .

Proposition 7.2. *A graph with a fixed minimum zero forcing set or minimum connected forcing set R can have exponentially-many sets of vertices which are terminals of forcing chains associated with R .*

Proof. Let $G' = (V, E)$ be the disjoint union of k copies of C_5 , where the i^{th} copy of C_5 has vertex set $\{a_i, b_i, c_i, u_i, v_i\}$ and edge set $\{a_i b_i, b_i c_i, c_i v_i, v_i u_i, u_i a_i\}$. Let $G = (V \cup \{x\}, E \cup \{xu_i : 1 \leq i \leq k\})$, and $R = \{x, u_1, \dots, u_k, v_1, \dots, v_k\}$. It is easy to see that R is a connected forcing set of G ; by Lemma 4.3 and Proposition 4.6, R is a minimum connected forcing set of G . In each copy of C_5 , possible forcing chains

initiated by u_i and v_i include $\{u_i \rightarrow a_i, v_i \rightarrow c_i \rightarrow b_i\}$ and $\{u_i \rightarrow a_i \rightarrow b_i, v_i \rightarrow c_i\}$. Thus, there are $\Omega(2^k)$ distinct sets of vertices which are terminals of forcing chains associated with R . Similarly, $R' = R \setminus \{x\}$ is a minimum zero forcing set of G' and there are $\Omega(2^k)$ distinct sets of vertices which are terminals of forcing chains associated with R' . \square

Finally, as mentioned in Chapter 5, if the zero forcing number of a graph is known, and if one of the minimum zero forcing sets of a graph is found to be connected, then the connected forcing number of the graph will immediately be determined as well. The following result shows that this approach is not efficient in general, even if distinct minimum zero forcing sets could be generated efficiently.

Proposition 7.3. *A graph could have exponentially-many minimum zero forcing sets which are not connected, and also have minimum zero forcing sets which are connected.*

Proof. Let $G = (V, E)$ be the sun graph on $n = 2k \geq 12$ vertices. By Theorem 5.1, $Z(G) = Z_c(G) = k - 1$; thus, G has minimum zero forcing sets which are connected. We will now show that G has exponentially-many minimum zero forcing sets which are not connected. Let $A = \{a_1, \dots, a_k\}$ be the set of vertices of the k -clique of G and let $B = \{b_1, \dots, b_k\}$ be the set of vertices of G with degree 2, where b_i is adjacent to a_{i-1} and a_i for $1 \leq i \leq k$ (with indices read modulo k). Let \mathcal{R} be the family of sets of the form $\{a_1, b_1, b_2, c_3, \dots, c_{k-2}\}$, where $c_i \in \{a_i, b_i\}$ for $i \in \{3, \dots, k-2\}$; we claim that each $R \in \mathcal{R}$ is a minimum zero forcing set. To see why, note that b_1 and b_2 can force a_k and a_2 in the first timestep, since they each only have one uncolored neighbor. Then, starting at $i = 3$ and incrementing i , either $c_i = a_i$, or $c_i = b_i$ and b_i can force a_i since a_{i-1} is already colored. In any case, eventually a_k, a_1, \dots, a_{k-2} will

be colored, and then a_1 can force a_k since both b_1 and b_2 are already to be colored. Then, starting at $i = 2$ and incrementing i , a_i can force b_{i+1} since b_i (and every vertex in A) is already colored. Thus, R is a zero forcing set, and since it has cardinality $k - 1$, it is a minimum zero forcing set.

Now, we will show that exponentially-many of the sets in \mathcal{R} are disconnected. Note that a sufficient condition for $R \in \mathcal{R}$ to be disconnected is that $c_i = b_i$ and $c_{i+1} = b_{i+1}$ for some $i \in \{3, \dots, k - 3\}$ (since then neither neighbor of b_{i+1} will be in R). Thus, to show that exponentially-many elements of \mathcal{R} are disconnected, it suffices to show that exponentially-many binary strings of length $k - 4$ contain at least two consecutive 0's (where a 0 corresponds to a choice of $c_i = b_i$ in the zero forcing set, and 1 corresponds to $c_i = a_i$). The number of binary strings of length p with at least two consecutive zeros is equal to $2^p - |J_p|$, where J_p is the set of binary strings of length p with no consecutive zeros. The strings in J_p that end in 0 can be obtained by appending '10' to the strings in J_{p-2} , and the strings in J_p that end in 1 can be obtained by appending '1' to the strings in J_{p-1} . Therefore, $|J_p| = |J_{p-1}| + |J_{p-2}|$. Moreover, $|J_1| = 2$ and $|J_2| = 3$, so $|J_p| = F_{p+2}$, where F_i is the i^{th} Fibonacci number. Since $F_p = O(\phi^p)$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio, it follows that $2^p - F_{p+2} = 2^p - O(\phi^{p+2}) = \Omega(2^p)$. Thus, G has $\Omega(2^k)$ disconnected minimum zero forcing sets. \square

I conclude this section with a brief discussion of when the quantities characterized in Propositions 7.1 and 7.2 are only polynomial in the order of the graph. Trivially, if $Z(G) = O(1)$ or $Z(G) = n - O(1)$, there are polynomially-many distinct minimum zero forcing sets, since G has polynomially-many subsets of vertices of size $O(1)$. For the same reason, if $Z(G) = O(1)$ or $Z(G) = n - O(1)$, then there are polynomially-many sets of vertices which are terminals of forcing chains associated with some fixed

zero forcing set R . Analogous conclusions hold true for the connected forcing number. It could also happen that these cardinalities are bounded by a polynomial even when $Z(G) = \Omega(n)$ and $Z_c(G) = \Omega(n)$; for example, this is the case when G is a sunlet graph.

7.2 Connected forcing and matroids

In this section, I investigate graphs for which the set of all connected forcing sets can be used to define greedoids or matroids. A *matroid* is an ordered pair (S, \mathcal{I}) where S is a finite set and \mathcal{I} is a subset of $\mathcal{P}(S)$ (the power set of S) satisfying

$$(M1) \emptyset \in \mathcal{I}$$

$$(M2) \text{ If } J' \subset J \in \mathcal{I} \text{ then } J' \in \mathcal{I}$$

$$(M3) \text{ For every } A \subset S, \text{ every maximal subset of } A \text{ in } \mathcal{I} \text{ has the same cardinality.}$$

An ordered pair (S, \mathcal{I}) which satisfies only (M1) and (M3) is called a *greedoid*. Matroids and greedoids have been studied extensively; see, e.g., [212, 213] for some of their fundamental properties, and in particular their connection to the greedy algorithm. We can define a *greedy algorithm* for finding a connected forcing set of a graph $G = (V, E)$ as follows:

Set $R = V$;

While there exists $v \in R$ with $R \setminus \{v\}$ being a connected forcing set,

Replace R by $R \setminus \{v\}$.

Clearly, this algorithm always produces a connected forcing set. The next results show that in some graphs, the greedy algorithm produces a minimum connected forcing set,

and that the collection of all connected forcing sets can be used to define greedoids and matroids.

Theorem 7.1. *Let \mathcal{T} be the family of trees, \mathcal{T}' be the family of trees whose pendant paths have length one, and \mathcal{B} be the family of block graphs with no pendant paths.*

1. *Let $G = (V, E) \in \mathcal{T} \cup \mathcal{B}$, $G \not\cong P_n$, and let \mathcal{I} be the set of all connected forcing sets of G . Then $(V, \mathcal{P}(V) \setminus \mathcal{I})$ is a greedoid.*
2. *Let $G = (V, E) \in \mathcal{T}' \cup \mathcal{B}$, $G \not\cong P_n$, and let \mathcal{I} be the set of all connected forcing sets of G . Then $(V, \mathcal{P}(V) \setminus \mathcal{I})$ is a matroid.*

Proof. Suppose G is a tree different from a path and let $A \subset V$. If $a \in A \setminus (R_2 \cup R_3)$, then a belongs to a pendant path of G . Let $X_1(A)$ be the set containing, for all $a \in A \setminus (R_2 \cup R_3)$, the vertices of the pendant path containing a which lie between a and the base of that pendant path, including a and the base of the path. Let $X_2(A)$ be the set containing, for $v \in R_3$, all-but-one bases of pendant paths attached to v which do not belong to pendant paths containing vertices of A . We claim that a minimal superset S of A which is a connected forcing set of G is the union of R_2 , R_3 , $X_1(A)$ and $X_2(A)$. First, note that S is clearly a superset of A since $A \subset R_2 \cup R_3 \cup X_1(A)$; S is also connected, since the only vertices of G which are not in S are connected parts of some pendant paths which contain the leaves of those pendant paths, and deleting those does not disconnect G . S is also forcing, since it contains \mathcal{M} , which by Theorem 5.2 is a minimum connected forcing set of G . Now suppose for contradiction that for some $s \in S$, $S \setminus \{s\}$ is also a connected forcing set. By Lemma 4.3, $R_2 \cup R_3 \subset S$, so $s \notin R_2 \cup R_3$. Since each $a \in A \setminus R_2 \cup R_3$ is in S , since the vertex attached to the pendant path containing a is in S , and since S is connected, S must also contain all vertices in that pendant path which lie between a and the base of the pendant

path; thus $s \notin X_1(A)$. Finally, by Lemma 4.3, S must contain all-but-one bases of pendant paths attached to each $v \in V$; thus, $s \notin X_2(A)$. Therefore, S is minimal. Since R_2 , R_3 and $X_1(A)$ are determined by the structure of G and the given set A , and the arbitrary choice of bases in $X_2(A)$ does not affect the cardinality of $X_2(A)$, every minimal superset S of A which is a connected forcing set of G has the same cardinality.

Next, suppose G is a block graph which has no pendant paths and is different from a path and let $A \subset V$. Let $X(A)$ be the set containing, for each block B of G , one non-cut vertex in B which is not in A , if such a vertex exists. We claim that a minimal superset S of A which is a connected forcing set of G equals $V \setminus X(A)$. First, note that by construction, S is a superset of A . S is also connected, since it excludes only non-cut vertices of G , and S is forcing, since it contains a minimum connected forcing set of G , namely, R as defined in the proof of Proposition 5.6. Suppose there is some $s \in S$ such that $S \setminus \{s\}$ is also a connected forcing set of G . By Lemma 4.3, s is a non-cut vertex of some block B , and $s \notin A$. However, since $X(A)$ contains a vertex from each block which has a non-cut vertex which is not in A , $X(A)$ must already include a vertex from B . However, by Proposition 4.6, S cannot exclude two vertices from B . Thus S is minimal, and since the arbitrary choice of vertices in $X(A)$ does not affect the cardinality of $X(A)$, every minimal superset S of A which is a connected forcing set of G has the same cardinality.

For any $G = (V, E) \in \mathcal{T} \cup \mathcal{B}$, V is clearly a connected forcing set of G . Thus, the ordered pair (V, \mathcal{I}) satisfies

$$(M1') \ V \in \mathcal{I}$$

$$(M3') \ \text{For every } A \subset V, \text{ every minimal superset of } A \text{ in } \mathcal{I} \text{ has the same cardi-}$$

nality.

Now, it can be verified that $(V, \mathcal{P}(V) \setminus \mathcal{I})$ satisfies properties (M1) and (M3) and is therefore a greedoid.

Suppose G is a tree whose pendant paths have length one, and let J be an arbitrary connected forcing set of G . Since by Lemma 4.3, $\mathcal{M} \subset J$, the only vertices of G not in J are some of the leaves of G . Let J' be a superset of J . Since each leaf of G is adjacent to a vertex in J and since $J \subset J'$ is a forcing set of G , J' is also a connected forcing set of G .

Suppose G is a block graph with no pendant paths and let J be an arbitrary connected forcing set of G . By Proposition 5.6, the only vertices of G not in J are up to one non-cut vertex in each block of G . Let J' be a superset of J . Since each non-cut vertex of G is adjacent to a vertex in J and since $J \subset J'$ is a forcing set of G , J' is also a connected forcing set of G .

Thus, for any $G = (V, E) \in \mathcal{T}' \cup \mathcal{B}$, the ordered pair (V, \mathcal{I}) satisfies

(M2') If $J' \supset J \in \mathcal{I}$ then $J' \in \mathcal{I}$.

Moreover, since $\mathcal{T}' \cup \mathcal{B} \subset \mathcal{T} \cup \mathcal{B}$, (V, \mathcal{I}) satisfies properties (M1') and (M3') as well. Thus, it can be verified that $(V, \mathcal{P}(V) \setminus \mathcal{I})$ satisfies properties (M1), (M2), and (M3) and is therefore a matroid. \square

I will now briefly address several (negative) results related to Theorem 7.1.

1. If $G = (V, E)$ is an arbitrary cactus or block graph (or a cactus graph with no pendant paths) and \mathcal{I} is the collection of connected forcing sets of G , then $(V, \mathcal{P}(V) \setminus \mathcal{I})$ is not necessarily a greedoid. As a simple counterexample, let G be the graph obtained by attaching two pendants to a triangle, each to a different

vertex. Let S_1 be the set containing both vertices of G of degree 3, and one vertex of degree 1, and S_2 be the set containing one vertex of degree 3 and one vertex of degree 2. S_1 and S_2 are minimal connected forcing sets, but do not have the same cardinality.

2. If $G = (V, E)$ is an arbitrary tree and \mathcal{I} is the collection of connected forcing sets of G , then $(V, \mathcal{P}(V) \setminus \mathcal{I})$ is not necessarily a matroid, since a superset of a connected forcing set of G could be disconnected.
3. If $G = (V, E)$ is a graph in \mathcal{T} , \mathcal{T}' , or \mathcal{B} (defined as in Theorem 7.1) and \mathcal{I} is the collection of zero forcing sets of G , then $(V, \mathcal{P}(V) \setminus \mathcal{I})$ is not necessarily a greedoid or a matroid, since, for example, not every minimal zero forcing set of a graph in these families is minimum. This is only true for restricted subfamilies like star graphs, complete graphs, and cycles. In general, when defining matroids as in Theorem 7.1, it appears that axiom (M2)' is harder to satisfy for the collection of connected forcing sets, since a superset of a connected forcing is always forcing but not always connected, and (M3)' is harder to satisfy for the collection of zero forcing sets, since there are no vertices which are part of every minimum zero forcing set of a graph.

Even if the collection of connected forcing sets of a graph does not define a greedoid or a matroid, the greedy algorithm may nevertheless produce a minimum connected forcing set. I give an example of a family of graphs for which this is the case.

Proposition 7.4. *Let G be a cactus graph different from a path, all of whose cycles are outer blocks. Then, the greedy algorithm produces a minimum connected forcing set of G .*

Proof. Let Q be defined as in the proof of Proposition 5.7; by a similar argument as in Proposition 5.7, $\mathcal{M} \cup Q$ is a minimum connected forcing set of G . Let S be a minimal connected forcing set of G . By Lemma 4.3, $\mathcal{M} \subset S$, and by Proposition 4.6, S contains at least two vertices of each cycle. Since the cut vertex of each cycle is in S and S is connected, at least one neighbor of the cut vertex of each cycle must be in S . However, a single colored neighbor of the cut vertex of each cycle is sufficient to initiate a forcing chain around the cycle; thus, for each cycle of G , S contains exactly one neighbor of the cut vertex of the cycle. Moreover, if S contains a vertex v which does not belong to \mathcal{M} or to any cycle of G , then v must belong to a pendant path of G ; however, v and all other vertices from that pendant path (except one which is in \mathcal{M}) can be removed from S , and the resulting set is still connected and forcing. Thus, S does not contain any vertices outside $\mathcal{M} \cup Q$. Therefore, every minimal connected forcing set of G is also minimum. By definition, the greedy algorithm produces a minimal connected forcing set of G ; thus, in this case it also produces a minimum connected forcing set. \square

The previous results use the fact that in the considered families of graphs, a minimal connected forcing set is also minimum. Similar properties of graphs have been investigated in relation to other parameters. For example, *well-covered graphs* are graphs in which every minimal vertex cover is also minimum; this property has many desirable consequences and has been investigated in [214, 215, 216].

Remark 7.1. The greedy algorithm has run time $O(n \cdot F(n))$, where $F(n)$ is the time required for checking whether a vertex set of size n is forcing. In general, such an approach would take superlinear time, whereas the constructions for finding minimum connected forcing sets of trees and the graphs described in Propositions 5.6 and 5.7

can be realized in linear time.

7.3 Zero forcing polynomial

In this section, I introduce the zero forcing and connected forcing polynomials of a graph, which count the number of distinct zero forcing and connected forcing sets of a given size. I give closed form expressions for the zero forcing and connected forcing polynomials of several families of graphs, and present results about the coefficients and values of these polynomials for general graphs.

Definition 7.1. Let G be a graph. Let $z(G; i)$ and $z_c(G; i)$ respectively be the number of zero forcing sets and the number of connected forcing sets of G with cardinality i .

The *zero forcing polynomial* of G is defined as

$$\mathcal{Z}(G; x) = \sum_{i=Z(G)}^n z(G; i)x^i.$$

The *connected forcing polynomial* of G is defined as

$$\mathcal{Z}_c(G; x) = \sum_{i=Z_c(G)}^n z_c(G; i)x^i.$$

Below are some basic properties of the coefficients of the zero forcing and connected forcing polynomials.

Theorem 7.2. Let $G = (V, E)$ be a graph. Then,

1. If G is connected, $z_c(G; n) = z(G; n) = 1$
2. $z(G; n - 1) = |\{v \in V : d(v) \neq 0\}|$
3. If G is connected, $z_c(G; n - 1) = |\{v \in V : \text{comp}(G \setminus \{v\}) = \text{comp}(G)\}|$
4. $z(G; n - 2) = |\{\{u, v\} \subset V : u \neq v, d(u) \neq 0, d(v) \neq 0, N(u) \setminus \{v\} \neq N(v) \setminus \{u\}\}|$

5. If G is connected, $z_c(G; n-2) = |\{\{u, v\} \subset V : u \neq v, \text{comp}(G \setminus \{u, v\}) = \text{comp}(G), N(u) \setminus \{v\} \neq N(v) \setminus \{u\}\}|$
6. $z(G; 1) = z_c(G; 1) = \begin{cases} 2 & \text{if } G \simeq P_n, n \geq 2 \\ 1 & \text{if } G \simeq P_1 \\ 0 & \text{otherwise} \end{cases}$
7. If $G \simeq G_1 \dot{\cup} G_2$, $\mathcal{Z}(G; x) = \mathcal{Z}(G_1; x)\mathcal{Z}(G_2; x)$
8. $z(G; i) = 0$ if and only if $i < Z(G)$; $z_c(G; i) = 0$ if and only if $i < Z_c(G)$
9. If G is connected, $z_c(G; i) \leq z(G; i)$ for $1 \leq i \leq n$
10. Zero is a root of $\mathcal{Z}(G; x)$ of multiplicity $Z(G)$ and a root of $\mathcal{Z}_c(G; x)$ of multiplicity $Z_c(G)$
11. $\mathcal{Z}(G; x)$ and $\mathcal{Z}_c(G; x)$ are strictly increasing in $[0, \infty)$

Proof. The numbers of the proofs below correspond to the numbers in the statement of the theorem. Statements 1., 8., 10., and 11. above follow directly from the definitions of the zero forcing and connected forcing polynomials.

2. Any non-isolated vertex v has a neighbor which can force v . Thus, each set which excludes one non-isolated vertex of G is a zero forcing set of size $n-1$; moreover, no set which excludes an isolated vertex is a zero forcing set.
3. If G is connected, any non-cut vertex v has a neighbor which can force v . Thus, each set which excludes one non-cut vertex of G is a connected forcing set of size $n-1$; moreover, no set which excludes a cut vertex is a connected forcing set.

4. Let u, v be two non-isolated vertices of G ; if $N(u) \setminus \{v\} \neq N(v) \setminus \{u\}$, there is a vertex w adjacent to one of u and v , but not the other. Suppose $u \sim w$; then, w can force u and any neighbor of v can force v (since $d(v) \neq 0$). Thus, any pair of vertices u, v satisfying these conditions can be excluded from a zero forcing set of size $n - 2$. On the other hand, a pair of vertices u, v which does not satisfy these conditions cannot be excluded from a zero forcing set, since every vertex which is adjacent to one will be adjacent to the other, and hence no vertex will be able to force u or v .
5. The proof is similar to the proof of 4. Note that since G is connected, $d(u) \neq 0$, $d(v) \neq 0$; moreover, any pair of vertices which form a separating set cannot be excluded from any connected forcing set of size $n - 2$.
6. The only graph with zero forcing number 1 is P_n ; thus if $G \not\simeq P_n$, $z(G; 1) = z_c(G; 1) = 0$. If $G \simeq P_n$ and $n \geq 2$, either end of the path is a zero forcing set; if $n = 1$, there is a single zero forcing set.
7. A zero forcing set of size i in G consists of a zero forcing set of size i_1 in G_1 and a zero forcing set of size $i_2 = i - i_1$ in G_2 . Since zero forcing sets of size i_1 and i_2 can be chosen independently in G_1 and G_2 for each $i_1 \geq Z(G_1)$, $i_2 \geq Z(G_2)$, and since $z(G_1; i_1)z(G_2; i_2) = 0$ for each $i_1 < Z(G_1)$ or $i_2 < Z(G_2)$, it follows that $z(G; i) = \sum_{i_1+i_2=i} z(G_1; i_1)z(G_2; i_2)$. The left-hand-side of this equation is the coefficient of x^i in $\mathcal{Z}(G)$, and since $\mathcal{Z}(G_1; x) = \sum_{i=Z(G_1)}^{|V(G_1)|} z(G_1; i)x^i$ and $\mathcal{Z}(G_2; x) = \sum_{i=Z(G_2)}^{|V(G_2)|} z(G_2; i)x^i$, the right-hand-side of the equation is the coefficient of x^i in $\mathcal{Z}(G_1; x)\mathcal{Z}(G_2; x)$. Thus, $\mathcal{Z}(G_1; x)\mathcal{Z}(G_2; x)$ and $\mathcal{Z}(G; x)$ have the same coefficients and the same degree, so they are identical.
9. Since every connected forcing set is a zero forcing set, the number of connected

forcing sets of size i is at most the number of zero forcing sets of size i . \square

I now give closed form expressions for the zero forcing and connected forcing polynomials of certain families of graphs.

Proposition 7.5. *If $n \geq 2$, $\mathcal{Z}(K_n; x) = \mathcal{Z}_c(K_n; x) = x^n + nx^{n-1}$.*

Proof. $Z(K_n) = Z_c(K_n) = n - 1$, so $z(K_n; i) = z_c(K_n; i) = 0$ for $i < n - 1$. By Theorem 7.2, $z(K_n; n - 1) = z_c(K_n; n - 1) = n$ and $z(K_n; n) = z_c(K_n; n) = 1$, so $\mathcal{Z}(K_n; x) = \mathcal{Z}_c(K_n; x) = x^n + nx^{n-1}$. \square

Proposition 7.6. *If $a_1 \dots, a_k \geq 2$, $\mathcal{Z}(K_{a_1, \dots, a_k}; x) = \mathcal{Z}_c(K_{a_1, \dots, a_k}; x) = x^n + nx^{n-1} + (\sum_{1 \leq i < j \leq k} a_i a_j) x^{n-2}$.*

Proof. $Z(K_{a_1, \dots, a_k}) = Z_c(K_{a_1, \dots, a_k}) = n - 2$, so $z(K_{a_1, \dots, a_k}; i) = z_c(K_{a_1, \dots, a_k}; i) = 0$ for $i < n - 2$. Each minimum zero forcing set and each minimum connected forcing set of K_{a_1, \dots, a_k} excludes a vertex from two of the parts of K_{a_1, \dots, a_k} ; there are $\sum_{1 \leq i < j \leq k} a_i a_j$ ways to pick such a pair of vertices, so $z(K_{a_1, \dots, a_k}; n - 2) = z_c(K_{a_1, \dots, a_k}; n - 2) = \sum_{1 \leq i < j \leq k} a_i a_j$. By Theorem 7.2, $z(K_{a_1, \dots, a_k}; n - 1) = z_c(K_{a_1, \dots, a_k}; n - 1) = n$ and $z(K_{a_1, \dots, a_k}; n) = z_c(K_{a_1, \dots, a_k}; n) = 1$, so $\mathcal{Z}(K_{a_1, \dots, a_k}; x) = \mathcal{Z}_c(K_{a_1, \dots, a_k}; x) = x^n + nx^{n-1} + (\sum_{1 \leq i < j \leq k} a_i a_j) x^{n-2}$. \square

Proposition 7.7. *If $n \geq 2$, $\mathcal{Z}_c(P_n; x) = 2x + \sum_{i=2}^n (n - i + 1)x^i$.*

Proof. Either endpoint of P_n is a connected forcing set of size 1, so $z_c(G; 1) = 2$. Label the vertices of P_n according to the order they are visited in depth-first-search starting from one of the endpoints. Let R be a connected forcing set of size i ; there are $n - (i - 1)$ ways to choose the vertex in R with the smallest label j . This choice uniquely determines R , since the other $i - 1$ vertices in R must be the vertices

with labels $j + 1, \dots, j + (i - 1)$. Thus, $z_c(P_n; i) = n - i + 1$ for $2 \leq i \leq n$, so $\mathcal{Z}_c(P_n; x) = 2x + \sum_{i=2}^n (n - i + 1)x^i$. \square

Proposition 7.8. *If $n \geq 3$, $\mathcal{Z}_c(C_n; x) = x^n + \sum_{i=2}^{n-1} nx^i$.*

Proof. $Z_c(C_n) = 2$, so $z_c(C_n; 1) = 0$; also, clearly, $z_c(C_n; n) = 1$. Label the vertices of C_n according to the order they are visited in depth-first-search starting from some arbitrary vertex. Let R be a connected forcing set of size i ; there are n ways to choose the vertex in R with label j , such that all remaining $i - 1$ vertices in R have labels $j + 1 \pmod n, \dots, j + (i - 1) \pmod n$. This choice of the vertex with label j uniquely determines R , so $z_c(C_n; i) = n$ for $2 \leq i \leq n - 1$. Thus, $\mathcal{Z}_c(C_n; x) = x^n + \sum_{i=2}^{n-1} nx^i$. \square

Proposition 7.9. *For $n \geq 1$, $\mathcal{Z}(P_n, x) = \sum_{i=1}^n \left(\binom{n}{i} - \binom{n-i-1}{i} \right) x^i$.*

Proof. The sets of size i which are not forcing are those which do not contain an end-vertex of the path and do not contain adjacent vertices. To count the number of non-forcing sets of size i , we can use the following argument: there are $n - i$ indistinguishable uncolored vertices to be placed in the $i + 1$ positions around the colored vertices, where each position must receive at least one uncolored vertex (in order for there not to be any adjacent colored vertices and for the end-vertices to be uncolored). There are $\binom{(n-i)-1}{(i+1)-1}$ ways to choose the positions of the uncolored vertices. Thus, there are $\binom{n}{i} - \binom{n-i-1}{i}$ zero forcing sets of size i . Note that when $i \geq \lceil \frac{n}{2} \rceil$, by the Pigeonhole Principle, one of the end-vertices or two adjacent vertices must be colored; this is resolved by the convention that $\binom{a}{b} = 0$ when $a < b$, and the fact that $n - i - 1 < i$ if and only if $i \geq \lceil \frac{n}{2} \rceil$. Thus, $\mathcal{Z}(P_n, x) = \sum_{i=1}^n \left(\binom{n}{i} - \binom{n-i-1}{i} \right) x^i$. \square

Proposition 7.10. *For $n \geq 3$, $\mathcal{Z}(C_n, x) = \sum_{i=2}^n \left(\binom{n}{i} - \frac{n}{i} \binom{n-i-1}{i-1} \right) x^i$.*

Proof. The sets of size i which are not forcing are those which do not contain adjacent vertices. To count the number of non-forcing sets of size i , we can use the following argument: first, select a representative vertex v of C_n and color it; this can be done in n ways. There are $n - i$ indistinguishable uncolored vertices to be placed in the i positions around the remaining $i - 1$ colored vertices, where each position must receive at least one uncolored vertex (in order for there not to be any adjacent colored vertices). There are $\binom{n-i-1}{i-1}$ ways to choose the positions of the uncolored vertices. This can be done for each of the n choices of a representative vertex v ; however, since we are interested in sets without a representative vertex, and since each set has been counted i times with a different representative vertex, we must divide this quantity by i . Thus, there are $\frac{n}{i} \binom{n-i-1}{i-1}$ ways to choose i vertices from C_n so that no two are adjacent. It follows that there are $\binom{n}{i} - \frac{n}{i} \binom{n-i-1}{i-1}$ zero forcing sets of size i . Note that when $i \geq \lfloor \frac{n}{2} \rfloor + 1$, by the Pigeonhole Principle, two adjacent vertices must be colored; this is resolved by the convention that $\binom{a}{b} = 0$ when $a < b$, and the fact that $n-i-1 < i-1$ if and only if $i \geq \lfloor \frac{n}{2} \rfloor + 1$. Thus, $\mathcal{Z}(C_n, x) = \sum_{i=2}^n (\binom{n}{i} - \frac{n}{i} \binom{n-i-1}{i-1}) x^i$. \square

The following notation will be used to characterize the connected forcing polynomials of trees.

Definition 7.2. Let $\mathcal{S} = \mathcal{S}(a; b_1, \dots, b_k)$ be the set of k -tuples of positive integers whose sum is a and whose i^{th} element is at most b_i . The cardinality of \mathcal{S} will be denoted by $s(a; b_1, \dots, b_k)$, and the elements of \mathcal{S} will be denoted by $S(a; b_1, \dots, b_k; 1), \dots, S(a; b_1, \dots, b_k; s)$, where $s = s(a; b_1, \dots, b_k)$. For $1 \leq j \leq k$, $S(a; b_1, \dots, b_k; i; j)$ will denote the j^{th} element of the k -tuple $S(a; b_1, \dots, b_k; i)$. When a and b_1, \dots, b_k are clear from the context, we will write for short $S_i = S(a; b_1, \dots, b_k; i)$ and $S_{i,j} = S(a; b_1, \dots, b_k; i; j)$ for $1 \leq i \leq s$, $1 \leq j \leq k$. Define $\mathcal{S}' = \mathcal{S}'(a; b_1, \dots, b_k)$ to be the

set of k -tuples of nonnegative integers whose sum is a and whose i^{th} element is at most b_i , and define s' , S' , and $S'_{i,j}$ analogously to s , S , and $S_{i,j}$.

As an example of Definition 7.2, consider the set $\mathcal{S}(5; 2, 2, 6)$. The cardinality of this set is 4, and its elements are $(1, 1, 3), (1, 2, 2), (2, 1, 2), (2, 2, 1)$. If $S_1 = (1, 1, 3)$, then $S_{11} = 1$, $S_{12} = 1$, and $S_{11} = 3$. Similarly, $\mathcal{S}'(5; 2, 2, 6) = \{(0, 0, 5), (0, 1, 4), (1, 0, 4), (1, 1, 3), (2, 0, 3), (0, 2, 3), (1, 2, 2), (2, 1, 2), (2, 2, 1)\}$.

In addition to their integer partition definitions, there are several ways to interpret the sets $\mathcal{S}(a; b_1, \dots, b_k)$ and $\mathcal{S}'(a; b_1, \dots, b_k)$. For example, \mathcal{S} also represents the ways to place a identical balls into k distinct boxes, so that the i^{th} box contains at least one and at most b_i balls; \mathcal{S} is also the set of feasible solutions to the integer program $\{\min 0 : \sum_{i=1}^k x_i = a, x_1 \leq b_1, \dots, x_k \leq b_k, x \in \mathbb{N}\}$. The latter also gives a way to obtain all the elements of \mathcal{S} , i.e., solving the integer program while adding previous solutions as constraints. The elements of \mathcal{S} and \mathcal{S}' can also be found by a recursive combinatorial algorithm, e.g., using dynamic programming. Note that the sets $\mathcal{S}(a + k; b_1 + 1, \dots, b_k + 1)$ and $\mathcal{S}'(a; b_1, \dots, b_k)$ are equivalent, in the sense that $s' = s$ and $S'_{i,j} = S_{i,j} - 1$. This can easily be seen from the context of distributing balls to boxes: in $\mathcal{S}(a + k; b_1 + 1, \dots, b_k + 1)$ every box must contain a ball, so k of the balls can be placed in the boxes right away, whereupon the capacity of box i becomes b_i ; then, the remaining a balls can be distributed to the boxes without a lower bound, which is exactly the process described by $\mathcal{S}'(a; b_1, \dots, b_k)$.

I now use Definition 7.2 to give characterization of the connected forcing polynomial of a special class of trees (called *spiders* or *generalized stars*), which will then be used to characterize the connected forcing polynomial of trees.

Proposition 7.11. *Let G be a graph composed of k pendant paths attached to a vertex*

v , where the i^{th} pendant path has b_i vertices, $1 \leq i \leq k$. Then,

$$\mathcal{Z}_c(G, x) = \sum_{j=1}^n \left(s(j-1; b_1, \dots, b_k) + \sum_{\ell=1}^k s(j-1; b_1, \dots, b_{\ell-1}, b_{\ell+1}, \dots, b_k) \right) x^j.$$

Proof. By Lemma 4.3 and Theorem 5.2, every connected set of vertices which contains v and all-but-one neighbors of v is a connected forcing set of G . Let p_1, \dots, p_k be the vertex sets of the pendant paths of G , where $|p_i| = b_i$. Then, the set of connected forcing sets of G of size j can be partitioned into the connected forcing sets of G of size j where every neighbor of v is colored, and the connected forcing sets of G of size j where the neighbor of v in p_i is not colored (and hence also all other vertices in p_i are not colored), $1 \leq i \leq k$.

The connected forcing sets of G of size j where every neighbor of v is colored can be counted by $s(j-1; b_1, \dots, b_k)$, since v must be colored, which leaves $j-1$ of the remaining vertices of G to be chosen, where at least one and at most b_i vertices are chosen from p_i , $1 \leq i \leq k$. The connected forcing sets of G of size j where no vertices of p_ℓ are chosen for some $\ell \in \{1, \dots, k\}$ can be counted by $s(j-1; b_1, \dots, b_{\ell-1}, b_{\ell+1}, \dots, b_k)$, since v must be colored, which leaves $j-1$ of the remaining vertices of G to be chosen, where at least one and at most b_i vertices are chosen from p_i , $1 \leq i \leq k$, $i \neq \ell$. Thus, the total number of connected forcing sets of G of size j is

$$z_c(G; j) = s(j-1; b_1, \dots, b_k) + \sum_{\ell=1}^k s(j-1; b_1, \dots, b_{\ell-1}, b_{\ell+1}, \dots, b_k).$$

By the convention that $s(a; b_1, \dots, b_k) = 0$ whenever $a < k$ or $a > \sum_{i=1}^k b_i$, we conclude that the connected forcing polynomial of G is as claimed. \square

Using the previous results, I now characterize the connected forcing polynomials of trees.

Theorem 7.3. *Let T be a tree different from a path. Let p_1, \dots, p_k be the vertex sets of the pendant paths in T . Let R_3^1 be the set of vertices to which a single pendant path is attached, and $R_3^2 = \{v_1, \dots, v_q\}$ be the set of vertices to which two or more pendant paths are attached. For $1 \leq i \leq q$, let $I_j = \{i : p_i \text{ is a pendant path attached to } v_j \in R_3^2\}$; also, let $I_{q+1} = \{i : p_i \text{ is a pendant path attached to some vertex in } R_3^1\}$. For $1 \leq j \leq q+1$, let $P_j = \bigcup_{i \in I_j} p_i$. Then, $\mathcal{Z}_c(T; x)$ is equal to*

$$\sum_{d=1}^n \left(\sum_{i=1}^{s'} \left(\left(\prod_{j=1}^q s(|I_j| - 1 + S'_{i,j}; [|p_\ell| : \ell \in I_j]) \right) s'(S'_{i,q+1}; [|p_\ell| : \ell \in I_{q+1}]) \right) \right) x^d,$$

where in the second sum, $s' = s'(d - |\mathcal{M}|; |P_1| - |I_1| + 1, \dots, |P_q| - |I_q| + 1, |P_{q+1}|)$, and for $1 \leq i \leq q+1$, $S'_{i,j} = S'(d - |\mathcal{M}|; |P_1| - |I_1| + 1, \dots, |P_q| - |I_q| + 1, |P_{q+1}|; i; j)$, and where $[|p_\ell| : \ell \in I_j]$ stands for the sequence of elements $|p_\ell|$ for all $\ell \in I_j$.

Proof. By Lemma 4.3 and Theorem 5.2, every connected set of vertices which contains $R_2 \cup R_3$ and all-but-one bases of pendant paths indexed by I_j , $1 \leq j \leq q$, is a connected forcing set of T . Let R be a connected forcing set of T of size d . Then, $|R_2 \cup R_3|$ of the vertices of R are taken up by the vertices in $R_2 \cup R_3$, and at least $|I_j| - 1$ of the vertices of R are taken up by vertices in P_j for $1 \leq j \leq q$. Thus, the remaining $d - |R_2 \cup R_3| - (|I_1| - 1) - \dots - (|I_q| - 1) = d - |R_2 \cup R_3| - |\mathcal{L}| = d - |\mathcal{M}|$ vertices of R can be taken up by any of the sets P_j , $1 \leq j \leq q+1$, as long as the number of vertices added to P_j does not exceed $|P_j| - (|I_j| - 1)$ for $1 \leq j \leq q$ and does not exceed $|P_j|$ for $j = q+1$. By Definition 7.2, the possible assignments of vertices according to these conditions are described by

$$\mathcal{S}'(d - |\mathcal{M}|; |P_1| - |I_1| + 1, \dots, |P_q| - |I_q| + 1, |P_{q+1}|) = \{S'_1, \dots, S'_{s'}\}.$$

Thus, the set of connected forcing sets of T of size d can be partitioned into s' parts, where the i^{th} part consists of the connected forcing sets of G of size d where

$|I_j| - 1 + S'_{i,j}$ of the colored vertices are in P_j , $1 \leq j \leq q$, and $S'_{i,q+1}$ of the colored vertices are in P_{q+1} . Moreover, when $|I_j| - 1 + S'_{i,j}$ vertices are allotted to P_j , $1 \leq j \leq q$, they must be distributed among the pendant paths indexed by I_j in such a way that all or all-but-one of them have at least one colored vertex, and so that the pendant path with vertex set p_ℓ does not receive more than $|p_\ell|$ vertices, for any $\ell \in I_j$. Similarly, when $S'_{i,q+1}$ vertices are allotted to P_{q+1} , they can be distributed among the pendant paths indexed by I_{q+1} without a lower bound, as long as path p_ℓ does not receive more than $|p_\ell|$ vertices, for any $\ell \in I_{q+1}$.

Finally, when some number of colored vertices is assigned to the sets P_1, \dots, P_{q+1} , one can distribute the colored vertices allotted to P_j to the specific pendant paths in P_j independently for each P_j . Thus, the number of connected forcing sets of G of size d where $|I_j| - 1 + S_{i,j}$ of the colored vertices are in P_j , $1 \leq j \leq q$, and $S'_{i,q+1}$ of the colored vertices are in P_{q+1} , is equal to the product of the number of ways to assign $|I_j| - 1 + S_{i,j}$ and $S'_{i,q+1}$ colored vertices, respectively, to the sets P_j , $1 \leq j \leq q$ and P_{q+1} . In turn, by a similar argument as in Proposition 7.11, the number of ways to assign $|I_j| - 1 + S'_{i,j}$ vertices to P_j , $1 \leq j \leq q$, is $s(|I_j| - 1 + S'_{i,j}; [|p_\ell| : \ell \in I_j])$, where $[|p_\ell| : \ell \in I_j]$ stands for the sequence of elements $|p_\ell|$ for all $\ell \in I_j$. Similarly, the number of ways to assign $S'_{i,q+1}$ vertices to P_{q+1} is $s'(S'_{i,q+1}; [|p_\ell| : \ell \in I_{q+1}])$. Thus, multiplying for $1 \leq j \leq q+1$ and summing over $1 \leq i \leq s' = s'(d - |\mathcal{M}|; |P_1| - |I_1| + 1, \dots, |P_q| - |I_q| + 1, |P_{q+1}|)$, we conclude that the number of connected forcing sets of G of size d is

$$z_c(T; d) = \sum_{i=1}^{s'} \left(\left(\prod_{j=1}^q s(|I_j| - 1 + S'_{i,j}; [|p_\ell| : \ell \in I_j]) \right) s'(S'_{i,q+1}; [|p_\ell| : \ell \in I_{q+1}]) \right).$$

By the conventions that $s'(0; b_1, \dots, b_k) = 1$, $s'(a; b_1, \dots, b_k) = 0$ whenever $a > \sum_{i=1}^k b_i$, and $s(a; b_1, \dots, b_k) = 0$ whenever $a < k$ or $a > \sum_{i=1}^k b_i$, we conclude that the

connected forcing polynomial of T is as claimed. \square

Example 7.1. In this example, I will apply Theorem 7.3 to enumerate the connected forcing sets of size $d = 8$ for the tree T shown in Figure 7.1. The pendant paths of T

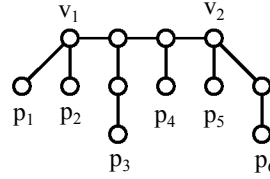


Figure 7.1 : A tree T for which $z_c(T; 8)$ is computed in Example 7.1.

are labeled p_1, \dots, p_6 . First, note that $Z_c(T) = |\mathcal{M}(T)| = 6$ and $R_3^2 = \{v_1, v_2\}$. Thus, $I_1 = \{1, 2\}$, $I_2 = \{5, 6\}$, $I_3 = \{3, 4\}$, and $P_1 = p_1 \cup p_2$, $P_2 = p_5 \cup p_6$, $P_3 = p_3 \cup p_4$. The total number of vertices in these sets is $|P_1| = 2$, $|P_2| = 3$, $|P_3| = 3$. Using these values, we compute

$$\begin{aligned} \mathcal{S}'(d - |\mathcal{M}|; |P_1| - |I_1| + 1, |P_2| - |I_2| + 1, |P_3|) &= \mathcal{S}'(2; 1, 2, 3) = \\ &= \{(1, 1, 0), (0, 2, 0), (1, 0, 1), (0, 1, 1), (0, 0, 2)\} = \{S'_1, S'_2, S'_3, S'_4, S'_5\} \end{aligned}$$

We now apply the formula for $z_c(T; d)$ given in Theorem 7.3.

$$\begin{aligned}
z_c(T; d) &= \sum_{i=1}^{s'} \left(\left(\prod_{j=1}^q s(|I_j| - 1 + S'_{i,j}; [|p_\ell| : \ell \in I_j]) \right) s'(S'_{i,q+1}; [|p_\ell| : \ell \in I_{q+1}]) \right) \\
z_c(T; 8) &= \sum_{i=1}^5 \left(s(2 - 1 + S'_{i,1}; |p_1|, |p_2|) s(2 - 1 + S'_{i,2}; |p_5|, |p_6|) \right) s'(S'_{i,3}; |p_3|, |p_4|) \\
&= \sum_{i=1}^5 s(1 + S'_{i,1}; 1, 1) s(1 + S'_{i,2}; 1, 2) s'(S'_{i,3}; 2, 1) \\
&= s(2; 1, 1) s(2; 2, 1) s'(0; 1, 2) + s(1; 1, 1) s(3; 2, 1) s'(0; 1, 2) + \\
&\quad s(2; 1, 1) s(1; 2, 1) s'(1; 1, 2) + s(1; 1, 1) s(2; 2, 1) s'(1; 1, 2) + \\
&\quad s(1; 1, 1) s(1; 2, 1) s'(2; 1, 2) \\
&= 1 \cdot 2 \cdot 1 + 2 \cdot 1 \cdot 1 + 1 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 + 2 \cdot 2 \cdot 2 = 24
\end{aligned}$$

Thus, there are 24 different connected forcing sets of size 8. These are shown in Figure 7.2, where the panel showing the connected forcing sets corresponding to S'_1 , S'_2 , S'_3 , S'_4 , and S'_5 , respectively, is the left-top, left-bottom, middle-left, middle-right, and right panel.

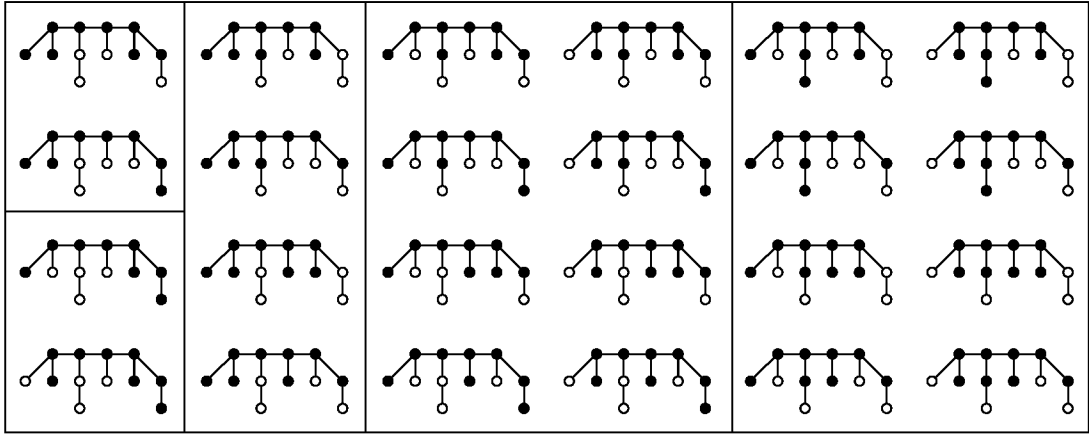


Figure 7.2 : All connected forcing sets of size 8 for the tree T .

The number of connected forcing sets of T of size d , $Z_c(T) = 6 \leq d \leq 12 = n$, (and

hence $\mathcal{Z}_c(T; x)$ can be found in a similar way.

I now present several structural results about the coefficients of the zero forcing and connected forcing polynomials. I begin with the following observation which implies that both bounds in $1 \leq z_c(G; Z_c(G)) \leq \binom{n}{Z_c(G)}$ are tight (the latter is tight, e.g., for complete graphs).

Observation 7.1. *A graph can have a unique minimum connected forcing set.*

Proof. The graph in Figure 7.3 has a unique minimum connected forcing set. □

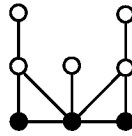


Figure 7.3 : A graph with a unique minimum connected forcing set, indicated by the colored vertices.

The same does not hold for zero forcing, since reversing the forcing chains associated with a zero forcing set produces another zero forcing set. Thus, a (non-empty) graph cannot have a unique minimum zero forcing set. I now strengthen the above observation by showing that only a minimum (and maximum) connected forcing set can be unique.

Theorem 7.4. $z_c(G; i) = 1$ only if $i = Z_c(G)$ or $i = n$.

Proof. Suppose G has a unique connected forcing set R of size i , where $Z_c(G) < i < n$. Let R' be a connected forcing set of size $i - 1$. Then $R' \subset R$, since otherwise R' together with any neighbor of R' forms a connected forcing set of size i different from R . Let $\{v\} = R \setminus R'$. Then $N(R') = \{v\}$, since otherwise R' together with either of its two or more neighbors would form a connected forcing set of size i . Moreover, for

all $u \in N(R)$, $N(u) \cap R = \{v\}$, since if some $u \in N(R)$ was adjacent to a vertex in R other than v , v would not be the only vertex in $N(R')$. If R' consists of a single vertex, then G is a path, and by Proposition 7.7, there is no unique connected forcing set of size $i < n$; thus, R' has at least two vertices. Let w be a non-cut vertex of $G[R]$ different from v , and let u be a vertex in $N(R)$. We claim $R \setminus \{w\} \cup \{u\}$ is a connected forcing set of G . This set is connected by construction, and it is forcing since w can be forced in the first timestep by any of its neighbors in R' . This contradicts R being the unique connected forcing set of size i . \square

The last result of this section concerns the unimodality of the zero forcing polynomial. I first recall a well-known theorem due to Hall [217]. A *matching* of $G = (V, E)$ is a set $M \subset E$ such that no two edges in M have a common endpoint. A matching M *saturates* a vertex v , if v is an endpoint of some edge in M .

Theorem 7.5 (Hall's Theorem [217]). *Let G be a bipartite graph with parts X and Y . G has a matching that saturates every vertex in X if and only if for all $S \subseteq X$, $|S| \leq |N(S)|$.*

Theorem 7.6. *Let $G = (V, E)$ be a graph. Then, $z(G; i) \leq z(G; i + 1)$ for $1 \leq i < \frac{n}{2}$.*

Proof. For every zero forcing set R of size i and every $v \in V \setminus R$, $R \cup \{v\}$ is a zero forcing set of cardinality $i + 1$. We will now show that to each zero forcing set of size i , $1 \leq i < \frac{n}{2}$, there corresponds a unique zero forcing set of size $i + 1$. Let H be a bipartite graph with parts X and Y , where the vertices of X are zero forcing sets of G of size i , and the vertices of Y are all subsets of V of size $i + 1$; a vertex $x \in X$ is adjacent to a vertex $y \in Y$ in H whenever $x \subset y$ in G . For each $x \in X$, there are $n - i$ vertices not in x ; thus, $d(x; H) = n - i$. Since a set of size $i + 1$ has $i + 1$ subsets of size i , it follows that for each $y \in Y$, $d(y; H) \leq i + 1$. Suppose for

contradiction that there exists a set $S \subset X$ such that $|S| > |N(S)|$. Since each vertex in S has $n - i$ neighbors and since $|S| > |N(S)|$, by the Pigeonhole Principle, some vertex $v \in N(S)$ must have more than $n - i$ neighbors. Thus, $i + 1 \geq d(v; H) > n - i$, whence it follows that $i \geq \frac{n}{2}$; this contradicts the assumption that $i < \frac{n}{2}$. Thus, for every $S \subseteq X$, $|S| \leq |N(S)|$; by Theorem 7.6, H has a matching that saturates all vertices of X . Thus, to each zero forcing set of size i , there corresponds a unique zero forcing set of size $i + 1$. We conclude that $z(G; i) \leq z(G; i + 1)$ for $1 \leq i < \frac{n}{2}$. \square

Chapter 8

Conclusion

In the first part of this thesis, I presented new efficient methods to compute the chromatic and flow polynomials of several families of graphs, including outerplanar graphs and generalized vertex joins of trees, cliques, and cycles. I also showed that in these graphs, computation based on the proposed methods strongly outperforms a general-purpose solver. If a graph contains one or more subgraphs whose chromatic or flow polynomials can be found efficiently, locating those subgraphs and modifying the deletion-contraction algorithm so that they appear as components in some step of the recursion can speed up the computation significantly. Thus, the proposed algorithms could be used in this capacity to improve the performance of general-purpose solvers. As discussed in Chapter 1, the chromatic and flow polynomials have applications in statistical physics, combinatorics, and theoretical computer science; thus, my results may also be applicable to the Traveling Salesman Problem and the anti-ferromagnetic Potts model — either directly on the graphs I consider, or on larger graphs which contain them as subgraphs. The novel theoretical results presented in this part of the thesis have been published in [218].

Because of the difficulty in computing the chromatic and flow polynomials of general graphs, research in this area has often focused on developing specialized algorithms for graphs with some exploitable structure. In this line of research, there are a number of simple families of graphs whose chromatic and flow polynomials are still being sought. Of note is the problem of efficiently computing the chromatic poly-

mial of an $a \times b$ grid graph, which is largely still unsolved (see [219, 220] for some results in this direction). Grid graphs have arbitrarily large treewidth, which renders bounded-treewidth algorithms like the ones discussed in Chapter 1 inapplicable. In the words of Read and Tutte, “that is an easy question to ask, but without a doubt a fiendishly difficult one to answer” [32]. The generalized vertex join operation provides a natural extension to all families of graphs; thus, future work could also focus on finding closed formulas for the chromatic polynomials of generalized vertex joins of various families of graphs whose chromatic polynomials are already known.

In the second part of this thesis, I explored several facets of the connected zero forcing problem. First, I presented a variety of structural results about connected forcing, such as the effects of vertex and edge operations on the connected forcing number, the relations between the connected forcing number and other graph parameters, and the computational complexity of connected forcing. I also gave efficient algorithms for computing the connected forcing numbers of different families of graphs, including certain product graphs, trees, unicyclic graphs, and cactus graphs with no pendant paths. In this direction, it would be useful to develop a general framework for computing the connected forcing numbers of graphs with cut vertices in terms of the connected forcing numbers of their blocks. Such a framework has been developed for the zero forcing number (cf. [105]), but the same approach does not carry over to connected forcing due to the unboundedness of the connected forcing spread of vertices and edges. Some of the novel theoretical results presented in this part of the thesis appear in [221, 222].

In the next part of the thesis, I characterized graphs with connected forcing numbers 1, 2, $n - 1$, and $n - 2$. In doing so, I employed novel combinatorial and graph theoretic techniques, which differ from the linear algebraic approaches typically used

in deriving similar characterizations. The results presented in this part of the thesis appear in [223]. A problem of interest is to obtain an analogous classification of graphs with connected forcing number or zero forcing number 3 and $n - 3$; some of the techniques developed in Chapter 6 could be useful toward that end. As part of my proof of Theorem 6.6, I introduced the notion of a connected forcing set which is required to contain a certain subset of the vertices of a graph (Definition 6.2). I term this notion *restrained connected forcing*; the notion of *restrained zero forcing* (and *restrained power domination*) can be defined analogously, i.e., a zero forcing set (power dominating set) of $G = (V, E)$ restrained by $S \subset V$ is a zero forcing set (power dominating set) which contains S . These notions could lead to improved modeling of some of the physical phenomena related to the forcing process. For example, if an electrical power network is expanded by additional substations and transmission lines, the electrical company would likely keep the existing PMUs and the non-portable supporting infrastructure at their former locations. Thus, it would be of interest to find a minimum power dominating set of the expanded network restrained by the set of locations of the existing PMUs.

Finally, I investigated several enumeration problems associated with zero forcing and connected forcing. I presented results regarding the cardinality and other properties of the set of minimum connected forcing sets of a graph G , the set of connected forcing sets of G which have a given size, and the set of all connected forcing sets of G (as well as analogous results for zero forcing). In particular, I identified some families of graphs for which the greedy algorithm produces minimum connected forcing sets, and showed that collections of connected forcing sets can be used to define greedoids and matroids. I also defined the zero forcing and connected forcing polynomials as the generating functions of the zero forcing and connected forcing sets of a graph, and

studied some of their properties like unimodality and multiplicativity with respect to connected components. I characterized the zero forcing and connected forcing polynomials of several families of graphs including cycles, complete graphs, and trees. Future work in this area could focus on studying the roots and coefficients of the zero forcing and connected forcing polynomials, and relating them to other graph polynomials. It would also be useful to derive recursive formulas for the connected forcing and zero forcing polynomials — akin to the deletion-contraction formula for the chromatic polynomial — which would aid in their computation for general graphs.

Bibliography

- [1] N. Biggs, *Algebraic Graph Theory*. Cambridge University Press, 1993.
- [2] C. D. Godsil, G. Royle, and C. Godsil, *Algebraic Graph Theory*, vol. 207. Springer New York, 2001.
- [3] Mishyp, “Map clip art.” Clker.com, 2011.
- [4] G. D. Birkhoff, “A determinant formula for the number of ways of coloring a map,” *The Annals of Mathematics*, vol. 14, no. 1/4, pp. 42–46, 1912.
- [5] H. Whitney, “A logical expansion in mathematics,” *Bulletin of the American Mathematical Society*, vol. 38, no. 8, pp. 572–579, 1932.
- [6] K. I. Appel and W. Haken, *Every planar map is four colorable*, vol. 98. American Mathematical Society, Providence, 1989.
- [7] W. T. Tutte, “A contribution to the theory of chromatic polynomials,” *Canadian Journal of Mathematics*, vol. 6, no. 80-91, pp. 3–4, 1954.
- [8] W. T. Tutte, “On dichromatic polynomials,” *Journal of Combinatorial Theory*, vol. 2, no. 3, pp. 301–320, 1967.
- [9] C. Hoede and X. Li, “Clique polynomials and independent set polynomials of graphs,” *Discrete Mathematics*, vol. 125, no. 1, pp. 219–228, 1994.

- [10] B. McClosky, A. Simms, and I. V. Hicks, “Co-2-plex polynomials,” *Journal of Combinatorial Optimization*, vol. 22, no. 4, pp. 640–650, 2011.
- [11] G. Chia, “A bibliography on chromatic polynomials,” *Discrete Mathematics*, vol. 172, no. 1, pp. 175–191, 1997.
- [12] F. M. Dong, K. M. Koh, and K.-L. Teo, *Chromatic Polynomials and Chromaticity of Graphs*. World Scientific, 2005.
- [13] C.-Q. Zhang, *Integer Flows and Cycle Covers of Graphs*, vol. 205. CRC Press, 1997.
- [14] W. T. Tutte, *Graph Theory, volume 21 of Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1984.
- [15] E. Süli and D. F. Mayers, *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.
- [16] T. Hubai, “The chromatic polynomial,” Master’s thesis, Eötvös Loránd University, 2009.
- [17] R. P. Stanley, “Acyclic orientations of graphs,” in *Classic Papers in Combinatorics*, pp. 453–460, Springer, 1987.
- [18] J. Oxley and D. Welsh, “Chromatic, flow and reliability polynomials: the complexity of their coefficients,” *Combinatorics, Probability & Computing*, vol. 11, no. 04, pp. 403–426, 2002.
- [19] J. Huh, “Milnor numbers of projective hypersurfaces and the chromatic polynomial of graphs,” *Journal of the American Mathematical Society*, vol. 25, no. 3,

pp. 907–927, 2012.

- [20] R. C. Read, “An introduction to chromatic polynomials,” *Journal of Combinatorial Theory*, vol. 4, no. 1, pp. 52–71, 1968.
- [21] W. Weickert, *Untersuchungen zum chromatischen Polynom anhand spezieller Basisdarstellungen*. PhD thesis, Tech Univ. Munchen, 1983.
- [22] H. H. Crapo, “A higher invariant for matroids,” *Journal of Combinatorial Theory*, vol. 2, no. 4, pp. 406–417, 1967.
- [23] T. H. Brylawski, “A decomposition for combinatorial geometries,” *Transactions of the American Mathematical Society*, vol. 171, pp. 235–282, 1972.
- [24] M. Las Vergnas, “The Tutte polynomial of a morphism of matroids II: Activities of orientations,” *Progress in Graph Theory, Proc. Waterloo Silver Jubilee Conf.* Academic Press, 1984.
- [25] J. A. Ellis-Monaghan, “Identities for circuit partition polynomials, with applications to the Tutte polynomial,” *Advances in Applied Mathematics*, vol. 32, no. 1, pp. 188–197, 2004.
- [26] K. Takamizawa, T. Nishizeki, and N. Saito, “Linear-time computability of combinatorial problems on series-parallel graphs,” *Journal of ACM*, vol. 29, no. 3, pp. 623–641, 1982.
- [27] E. G. Whitehead and L.-C. Zhao, “Cutpoints and the chromatic polynomial,” *Journal of Graph Theory*, vol. 8, no. 3, pp. 371–377, 1984.
- [28] C. Thomassen, “Chromatic roots and Hamiltonian paths,” *Journal of Combinatorial Theory, Series B*, vol. 80, no. 2, pp. 218–224, 2000.

- [29] C. Thomassen, “The zero-free intervals for chromatic polynomials of graphs,” *Combinatorics, Probability and Computing*, vol. 6, no. 4, pp. 497–506, 1997.
- [30] A. D. Sokal, “Chromatic roots are dense in the whole complex plane,” *Combinatorics, Probability and Computing*, vol. 13, no. 2, pp. 221–261, 2004.
- [31] B. Jackson, “A zero-free interval for chromatic polynomials of graphs,” *Combinatorics, Probability and Computing*, vol. 2, no. 3, pp. 325–336, 1993.
- [32] R. C. Read and W. T. Tutte, “Chromatic polynomials,” *Selected Topics in Graph Theory*, vol. 3, pp. 15–42, 1988.
- [33] P. J. Heawood, “Map-colour theorem,” *Proceedings of the London Mathematical Society*, vol. 2, no. 1, pp. 161–175, 1949.
- [34] G. D. Birkhoff and D. C. Lewis, “Chromatic polynomials,” *Transactions of the American Mathematical Society*, pp. 355–451, 1946.
- [35] M. Kubale, *Graph Colorings*, vol. 352. American Mathematical Soc., 2004.
- [36] A. D. Sokal, “Chromatic polynomials, Potts models and all that,” *Physica A: Statistical Mechanics and its Applications*, vol. 279, no. 1, pp. 324–332, 2000.
- [37] A. Mohr and T. Porter, “Applications of chromatic polynomials involving stirring numbers,” *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 70, pp. 57–64, 2009.
- [38] E. G. Whitehead, “Stirling number identities from chromatic polynomials,” *Journal of Combinatorial Theory, Series A*, vol. 24, no. 3, pp. 314–317, 1978.

- [39] S. Beraha, J. Kahane, and N. Weiss, “Limits of chromatic zeros of some families of maps,” *Journal of Combinatorial Theory, Series B*, vol. 28, no. 1, pp. 52–65, 1980.
- [40] W. Tutte, “On chromatic polynomials and the golden ratio,” *Journal of Combinatorial Theory*, vol. 9, no. 3, pp. 289–296, 1970.
- [41] S. Pemmaraju and S. Skiena, *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica®*. Cambridge University Press, 2003.
- [42] H. S. Wilf, *Algorithms and complexity*, vol. 986. Prentice-Hall Englewood Cliffs, NJ, 1986.
- [43] A. Bjorklund, T. Husfeldt, P. Kaski, and M. Koivisto, “Computing the Tutte polynomial in vertex-exponential time,” in *Foundations of Computer Science, 49th Annual IEEE Symposium*, pp. 677–686, 2008.
- [44] J. A. Makowsky, U. Rotics, I. Averbouch, and B. Godlin, “Computing graph polynomials on graphs of bounded clique-width,” in *Graph-theoretic Concepts in Computer Science*, pp. 191–204, Springer, 2006.
- [45] A. Andrzejak, “An algorithm for the Tutte polynomials of graphs of bounded treewidth,” *Discrete Mathematics*, vol. 190, no. 1, pp. 39–54, 1998.
- [46] G. Szekeres, “Polyhedral decompositions of cubic graphs,” *Bulletin of the Australian Mathematical Society*, vol. 8, no. 3, pp. 367–387, 1973.
- [47] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, vol. 290. Macmillan London, 1976.

- [48] P. D. Seymour, “Nowhere-zero 6-flows,” *Journal of Combinatorial Theory, Series B*, vol. 30, no. 2, pp. 130–135, 1981.
- [49] D. P. Dailey, “Uniqueness of colorability and colorability of planar 4-regular graphs are np-complete,” *Discrete Mathematics*, vol. 30, no. 3, pp. 289–293, 1980.
- [50] M. Kochol, “Polynomials associated with nowhere-zero flows,” *Journal of Combinatorial Theory, Series B*, vol. 84, no. 2, pp. 260–269, 2002.
- [51] P. A. Kilpatrick, “Tutte’s first colour-cycle conjecture,” Master’s thesis, University of Cape Town, 1975.
- [52] F. Jaeger, “On nowhere-zero flows in multigraphs,” in *Proceedings of the Fifth British Combinatorial Conference*, vol. 15, pp. 373–378, 1976.
- [53] M. Noy, “Acyclic and totally cyclic orientations in planar graphs,” *American Mathematical Monthly*, pp. 66–68, 2001.
- [54] B. Jackson, “Zeros of chromatic and flow polynomials of graphs,” *Journal of Geometry*, vol. 76, no. 1, pp. 95–109, 2003.
- [55] F. M. Dong and K. M. Koh, “Bounds for the coefficients of flow polynomials,” *Journal of Combinatorial Theory, Series B*, vol. 97, no. 3, pp. 413–420, 2007.
- [56] F. E. Allen, “Control flow analysis,” *ACM Sigplan Notices*, vol. 5, no. 7, pp. 1–19, 1970.
- [57] T. J. McCabe, “A complexity measure,” *IEEE Transactions on Software Engineering*, no. 4, pp. 308–320, 1976.

- [58] L. M. Lovász, C. Thomassen, Y. Wu, and C.-Q. Zhang, “Nowhere-zero 3-flows and modulo k -orientations,” *Journal of Combinatorial Theory, Series B*, vol. 103, no. 5, pp. 587–598, 2013.
- [59] C. Thomassen, “The weak 3-flow conjecture and the weak circular flow conjecture,” *Journal of Combinatorial Theory, Series B*, vol. 102, no. 2, pp. 521–529, 2012.
- [60] F. Jaeger, “Flows and generalized coloring theorems in graphs,” *Journal of Combinatorial Theory, Series B*, vol. 26, no. 2, pp. 205–216, 1979.
- [61] W. Tutte, “On the algebraic theory of graph colorings,” *Journal of Combinatorial Theory*, vol. 1, no. 1, pp. 15–50, 1966.
- [62] C. Thomassen, “The weak 3-flow conjecture and the weak circular flow conjecture,” *Journal of Combinatorial Theory, Series B*, vol. 102, no. 2, pp. 521–529, 2012.
- [63] M. Kochol, “An equivalent version of the 3-flow conjecture,” *Journal of Combinatorial Theory, Series B*, vol. 83, no. 2, pp. 258–261, 2001.
- [64] C. Thomassen, Y. Wu, and C.-Q. Zhang, “The 3-flow conjecture, factors modulo k , and the 1-2-3-conjecture,” *Journal of Combinatorial Theory, Series B*, vol. 121, pp. 308–325, 2016.
- [65] E. H. Lieb, “Residual entropy of square ice,” *Physical Review*, vol. 162, no. 1, p. 162, 1967.
- [66] L. Pauling, “The structure and entropy of ice and of other crystals with some randomness of atomic arrangement,” *Journal of the American Chemical Society*,

- vol. 57, no. 12, pp. 2680–2684, 1935.
- [67] J. C. Slater, “Theory of the transition in KH_2PO_4 ,” *The Journal of Chemical Physics*, vol. 9, no. 1, pp. 16–33, 1941.
- [68] M. Kochol, “Decomposition formulas for the flow polynomial,” *European Journal of Combinatorics*, vol. 26, no. 7, pp. 1086–1093, 2005.
- [69] F. Jaeger, “Even subgraph expansions for the flow polynomial of cubic plane maps,” *Journal of Combinatorial Theory, Series B*, vol. 52, no. 2, pp. 259–273, 1991.
- [70] W. Kook, V. Reiner, and D. Stanton, “A convolution formula for the Tutte polynomial,” *Journal of Combinatorial Theory, Series B*, vol. 76, no. 2, pp. 297–300, 1999.
- [71] A. D. Sokal, “The multivariate Tutte polynomial (alias Potts model) for graphs and matroids,” *Surveys in Combinatorics*, vol. 327, pp. 173–226, 2005.
- [72] J. A. Ellis-Monaghan and C. Merino, “Graph polynomials and their applications I: The Tutte polynomial,” in *Structural Analysis of Complex Networks*, pp. 219–255, Springer, 2011.
- [73] M. Chari and C. J. Colbourn, “Reliability polynomials: a survey,” *Journal of Combinatorics, Information and System Sciences*, vol. 22, no. 3-4, pp. 177–193, 1997.
- [74] V. F. Jones, “A polynomial invariant for knots via von Neumann algebras,” *Bulletin of the American Mathematical Society*, vol. 12, pp. 103–111, 1985.

- [75] L. Lovász, *Large Networks and Graph Limits*, vol. 60. American Mathematical Soc., 2012.
- [76] S. Butenko, *Maximum independent set and related problems, with applications*. PhD thesis, University of Florida, 2003.
- [77] B. Balasundaram, S. Butenko, and I. V. Hicks, “Clique relaxations in social network analysis: The maximum k-plex problem,” *Operations Research*, vol. 59, no. 1, pp. 133–142, 2011.
- [78] S. Alikhani and Y.-h. Peng, “Introduction to domination polynomial of a graph,” *Ars Combinatoria*, vol. 114, pp. 257–266, 2014.
- [79] T. Kotek, J. Preen, F. Simon, P. Tittmann, and M. Trinks, “Recurrence relations and splitting formulas for the domination polynomial,” *Electronic Journal of Combinatorics*, vol. 19, no. 3, 2012.
- [80] J. I. Brown and J. Tufts, “On the roots of domination polynomials,” *Graphs and Combinatorics*, vol. 30, no. 3, pp. 527–547, 2014.
- [81] S. Jahari and S. Alikhani, “Domination polynomials of k-tree related graphs,” *International Journal of Combinatorics*, vol. 2014, 2014.
- [82] S. Akbari, S. Alikhani, and Y.-h. Peng, “Characterization of graphs using domination polynomials,” *European Journal of Combinatorics*, vol. 31, no. 7, pp. 1714–1724, 2010.
- [83] G. Deepak and N. Soner, “Connected domination polynomial of a graph,” *International Journal of Mathematical Archive*, vol. 4, no. 11, pp. 90–96, 2014.

- [84] C. Hoede and X. Li, “Clique polynomials and independent set polynomials of graphs,” *Discrete Mathematics*, vol. 125, no. 1-3, pp. 219–228, 1994.
- [85] H. Hajiabolhassan and M. Mehrabadi, “On clique polynomials,” *Australasian Journal of Combinatorics*, vol. 18, pp. 313–316, 1998.
- [86] F. M. Dong, M. D. Hendy, K. L. Teo, and C. H. Little, “The vertex-cover polynomial of a graph,” *Discrete Mathematics*, vol. 250, no. 1-3, pp. 71–78, 2002.
- [87] S. Akbari and M. R. Oboudi, “On the edge cover polynomial of a graph,” *European Journal of Combinatorics*, vol. 34, no. 2, pp. 297–321, 2013.
- [88] J. A. Ellis-Monaghan and C. Merino, “Graph polynomials and their applications ii: Interrelations and interpretations,” in *Structural Analysis of Complex Networks*, pp. 257–292, Springer, 2011.
- [89] A. S. W. Group, “Zero forcing sets and the minimum rank of graphs,” *Linear Algebra and its Applications*, vol. 428, no. 7, pp. 1628–1648, 2008.
- [90] D. Burgarth and V. Giovannetti, “Full control by locally induced relaxation,” *Physical Review Letters*, vol. 99, no. 10, p. 100501, 2007.
- [91] B. Yang, “Fast-mixed searching and related problems on graphs,” *Theoretical Computer Science*, vol. 507, pp. 100–113, 2013.
- [92] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning, “Domination in graphs applied to electric power networks,” *SIAM Journal on Discrete Mathematics*, vol. 15, no. 4, pp. 519–529, 2002.

- [93] D. Burgarth, D. D'Alessandro, L. Hogben, S. Severini, and M. Young, "Zero forcing, linear and quantum controllability for systems evolving on networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2349–2354, 2013.
- [94] D. Burgarth, V. Giovannetti, L. Hogben, S. Severini, and M. Young, "Logic circuits from zero forcing," *Natural Computing*, vol. 14, no. 3, pp. 485–490, 2015.
- [95] Y. Caro and R. Pepper, "Dynamic approach to k-forcing," *Theory and Applications of Graphs*, vol. 2, no. 2, 2015.
- [96] P. A. Dreyer and F. S. Roberts, "Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion," *Discrete Applied Mathematics*, vol. 157, no. 7, pp. 1615–1627, 2009.
- [97] F. Barioli, W. Barrett, S. M. Fallat, H. T. Hall, L. Hogben, B. Shader, P. van den Driessche, and H. Van Der Holst, "Parameters related to tree-width, zero forcing, and maximum nullity of a graph," *Journal of Graph Theory*, vol. 72, no. 2, pp. 146–177, 2013.
- [98] S. M. Fallat and L. Hogben, "The minimum rank of symmetric matrices described by a graph: a survey," *Linear Algebra and its Applications*, vol. 426, no. 2, pp. 558–582, 2007.
- [99] M. Trefois and J.-C. Delvenne, "Zero forcing number, constrained matchings and strong structural controllability," *Linear Algebra and its Applications*, vol. 484, pp. 199–218, 2015.
- [100] S. Butler, J. Grout, and H. T. Hall, "Using variants of zero forcing to bound

- the inertia set of a graph,” *Electronic Journal of Linear Algebra*, vol. 30, no. 1, p. 1, 2015.
- [101] F. Goldberg and A. Berman, “Zero forcing for sign patterns,” *Linear Algebra and its Applications*, vol. 447, pp. 56–67, 2014.
- [102] M. Allison, E. Bodine, L. M. DeAlba, J. Debnath, L. DeLoss, C. Garnett, J. Grout, L. Hogben, B. Im, H. Kim, *et al.*, “Minimum rank of skew-symmetric matrices described by a graph,” *Linear Algebra and its Applications*, vol. 432, 2010.
- [103] A. Aazami, *Hardness results and approximation algorithms for some problems on graphs*. PhD thesis, PhD thesis. University of Waterloo, 2008.
- [104] C. J. Edholm, L. Hogben, J. LaGrange, and D. D. Row, “Vertex and edge spread of zero forcing number, maximum nullity, and minimum rank of a graph,” *Linear Algebra and its Applications*, vol. 436, no. 12, pp. 4352–4372, 2012.
- [105] D. D. Row, “A technique for computing the zero forcing number of a graph with a cut-vertex,” *Linear Algebra and its Applications*, vol. 436, no. 12, pp. 4423–4432, 2012.
- [106] L.-H. Huang, G. J. Chang, and H.-G. Yeh, “On minimum rank and zero forcing sets of a graph,” *Linear Algebra and its Applications*, vol. 432, no. 11, pp. 2961–2973, 2010.
- [107] R. Davila and F. Kenter, “Bounds for the zero forcing number of a graph with large girth,” *Theory and Applications of Graphs*, vol. 2, 2015.

- [108] M. Gentner, L. D. Penso, D. Rautenbach, and U. S. Souza, “Extremal values and bounds for the zero forcing number,” *Discrete Applied Mathematics*, vol. 214, pp. 196–200, 2016.
- [109] K. F. Benson, D. Ferrero, M. Flagg, V. Furst, L. Hogben, V. Vasilevska, and B. Wissman, “Power domination and zero forcing,” *arXiv:1510.02421*, 2015.
- [110] S. A. Meyer, “Zero forcing sets and bipartite circulants,” *Linear Algebra and its Applications*, vol. 436, no. 4, pp. 888–900, 2012.
- [111] L. Eroh, C. X. Kang, and E. Yi, “Metric dimension and zero forcing number of two families of line graphs,” *Mathematica Bohemica*, vol. 139, no. 3, pp. 467–483, 2014.
- [112] E. Yi, “On zero forcing number of permutation graphs,” in *International Conference on Combinatorial Optimization and Applications*, pp. 61–72, Springer, 2012.
- [113] M. Catral, A. Cepek, L. Hogben, K. Lazebnik, T. Peters, and M. Young, “Zero forcing number, maximum nullity, and path cover number of subdivided graphs,” *Electronic Journal of Linear Algebra*, vol. 23, no. 1, p. 64, 2012.
- [114] S. Butler and M. Young, “Throttling zero forcing propagation speed on graphs,” *Australasian Journal of Combinatorics*, vol. 57, pp. 65–71, 2013.
- [115] K. B. Chilakamarri, N. Dean, C. X. Kang, and E. Yi, “Iteration index of a zero forcing set in a graph,” *arXiv:1105.1492*, 2011.
- [116] L. Hogben, N. Kingsley, S. Meyer, S. Walker, and M. Young, “Propagation time for zero forcing on a graph,” *Discrete Applied Mathematics*, vol. 160, no. 13,

pp. 1994–2005, 2012.

- [117] A. Berliner, C. Bozeman, S. Butler, M. Catral, L. Hogben, B. Kroschel, J. C.-H. Lin, N. Warnberg, and M. Young, “Zero forcing propagation time on oriented graphs,” *Discrete Applied Mathematics*, 2017.
- [118] N. Warnberg, “Positive semidefinite propagation time,” *Discrete Applied Mathematics*, vol. 198, pp. 274–290, 2016.
- [119] S. Butler, L. DeLoss, J. Grout, H. Hall, J. LaGrange, T. McKay, J. Smith, and G. Tims, “Minimum rank library,” 2014. Sage programs for calculating bounds on the minimum rank of a graph, and for computing zero forcing parameters. Available at https://github.com/jasongrout/minimum_rank.
- [120] B. Brimkov, C. C. Fast, and I. V. Hicks, “Computational approaches for zero forcing and related problems,” *arXiv:1704.02065*, 2017.
- [121] P. M. Nylen, “Minimum-rank matrices with prescribed graph,” *Linear Algebra and its Applications*, vol. 248, pp. 303–316, 1996.
- [122] F. Barioli, S. Fallat, and L. Hogben, “A variant on the graph parameters of colin de verdier: Implications to the minimum rank of graphs,” *Electronic Journal of Linear Algebra*, vol. 13, no. 1, p. 24, 2005.
- [123] L. M. DeAlba, T. L. Hardy, I. R. Hentzel, L. Hogben, and A. Wangsness, “Minimum rank and maximum eigenvalue multiplicity of symmetric tree sign patterns,” *Linear Algebra and its Applications*, vol. 418, no. 2-3, pp. 394–415, 2006.

- [124] G. Ding and A. Kotlov, “On minimal rank over finite fields,” *Electronic Journal of Linear Algebra*, vol. 15, no. 1, p. 16, 2006.
- [125] F. Barioli, S. Fallat, and L. Hogben, “On the difference between the maximum multiplicity and path cover number for tree-like graphs,” *Linear Algebra and its Applications*, vol. 409, pp. 13–31, 2005.
- [126] W. Barrett, H. Van Der Holst, and R. Loewy, “Graphs whose minimal rank is two,” *Electronic Journal of Linear Algebra*, vol. 11, no. 258-280, p. 687, 2004.
- [127] F. Barioli, S. Fallat, and L. Hogben, “Computation of minimal rank and path cover number for certain graphs,” *Linear Algebra and its Applications*, vol. 392, pp. 289–303, 2004.
- [128] F. Barioli, W. Barrett, S. M. Fallat, H. T. Hall, L. Hogben, B. Shader, P. Van Den Driessche, and H. Van Der Holst, “Zero forcing parameters and minimum rank problems,” *Linear Algebra and its Applications*, vol. 433, no. 2, pp. 401–411, 2010.
- [129] L. H. Mitchell, S. K. Narayan, and A. M. Zimmer, “Lower bounds in minimum rank problems,” *Linear Algebra and its Applications*, vol. 432, no. 1, pp. 430–440, 2010.
- [130] J. C.-H. Lin, “The sieving process and lower bounds for the minimum rank problem,” *Congressus Numerantium*, vol. 219, pp. 73–88, 2014.
- [131] E. Almodovar, L. DeLoss, L. Hogben, K. Hogenson, K. Murphy, T. Peters, and C. A. Ramírez, “Minimum rank, maximum nullity and zero forcing number for selected graph families,” *Involve, a Journal of Mathematics*, vol. 3, no. 4, pp. 371–392, 2011.

- [132] D. Row, “Zero forcing number, path cover number, and maximum nullity of cacti,” *Involve, a Journal of Mathematics*, vol. 4, no. 3, pp. 277–291, 2012.
- [133] T. Baldwin, L. Mili, M. Boisen, and R. Adapa, “Power system observability with minimal phasor measurement placement,” *IEEE Transactions on Power Systems*, vol. 8, no. 2, pp. 707–715, 1993.
- [134] K. F. Benson, D. Ferrero, M. Flagg, V. Furst, L. Hogben, V. Vasilevska, and B. Wissman, “Zero forcing and power domination for graph products,” 2016. Preprint; Available at http://orion.math.iastate.edu/lhogben/research/powerdom_BFFFHVW.pdf.
- [135] C.-S. Liao and D.-T. Lee, “Power domination problem in graphs,” in *International Computing and Combinatorics Conference*, pp. 818–828, Springer, 2005.
- [136] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith, “Parameterized power domination complexity,” *Information Processing Letters*, vol. 98, no. 4, pp. 145–149, 2006.
- [137] G. Xu, L. Kang, E. Shan, and M. Zhao, “Power domination in block graphs,” *Theoretical Computer Science*, vol. 359, no. 1-3, pp. 299–305, 2006.
- [138] M. Dorfling and M. A. Henning, “A note on power domination in grid graphs,” *Discrete Applied Mathematics*, vol. 154, no. 6, pp. 1023–1027, 2006.
- [139] P. Dorbec, M. Mollard, S. Klavžar, and S. Špacapan, “Power domination in product graphs,” *SIAM Journal on Discrete Mathematics*, vol. 22, no. 2, pp. 554–567, 2008.

- [140] K. Koh and K. Soh, “Power domination of the cartesian product of graphs,” *AKCE International Journal of Graphs and Combinatorics*, vol. 13, no. 1, pp. 22–30, 2016.
- [141] M. Zhao, L. Kang, and G. J. Chang, “Power domination in graphs,” *Discrete Mathematics*, vol. 306, no. 15, pp. 1812–1816, 2006.
- [142] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, “The complexity of searching a graph,” *Journal of the ACM*, vol. 35, no. 1, pp. 18–44, 1988.
- [143] L. M. Kirousis and C. H. Papadimitriou, “Searching and pebbling,” *Theoretical Computer Science*, vol. 47, pp. 205–218, 1986.
- [144] D. Bienstock and P. Seymour, “Monotonicity in graph searching,” *Journal of Algorithms*, vol. 12, no. 2, pp. 239–245, 1991.
- [145] D. Dyer, B. Yang, and Ö. Yaşar, “On the fast searching problem,” in *International Conference on Algorithmic Applications in Management*, pp. 143–154, Springer, 2008.
- [146] B. Yang, “Fast edge searching and fast searching on graphs,” *Theoretical Computer Science*, vol. 412, no. 12–14, pp. 1208–1219, 2011.
- [147] D. Stanley and B. Yang, “Fast searching games on graphs,” *Journal of Combinatorial Optimization*, vol. 22, no. 4, pp. 763–777, 2011.
- [148] S. Fallat, K. Meagher, and B. Yang, “On the complexity of the positive semidefinite zero forcing number,” *Linear Algebra and its Applications*, vol. 491, pp. 101–122, 2016.

- [149] S. Severini, “Nondiscriminatory propagation on trees,” *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 48, p. 482002, 2008.
- [150] D. Burgarth and K. Maruyama, “Indirect hamiltonian identification through a small gateway,” *New Journal of Physics*, vol. 11, no. 10, p. 103019, 2009.
- [151] D. Burgarth, S. Bose, C. Bruder, and V. Giovannetti, “Local controllability of quantum networks,” *Physical Review A*, vol. 79, no. 6, p. 060305, 2009.
- [152] J. Ekstrand, C. Erickson, H. T. Hall, D. Hay, L. Hogben, R. Johnson, N. Kingsley, S. Osborne, T. Peters, J. Roat, *et al.*, “Positive semidefinite zero forcing,” *Linear Algebra and its Applications*, vol. 439, no. 7, pp. 1862–1874, 2013.
- [153] S. Fallat, K. Meagher, A. Soltani, and B. Yang, “Positive zero forcing and edge clique coverings,” in *International Workshop on Frontiers in Algorithmics*, pp. 53–64, Springer, 2016.
- [154] T. Peters, “Positive semidefinite maximum nullity and zero forcing number,” *Electronic Journal of Linear Algebra*, vol. 23, 2012.
- [155] L. Hogben, K. F. Palmowski, D. E. Roberson, and M. Young, “Fractional zero forcing via three-color forcing games,” *Discrete Applied Mathematics*, vol. 213, pp. 114–129, 2016.
- [156] K. Fetcie, B. Jacob, and D. Saavedra, “The failed zero forcing number of a graph,” *Involve, a Journal of Mathematics*, vol. 8, no. 1, pp. 99–117, 2014.
- [157] Y. Shitov, “On the complexity of failed zero forcing,” *Theoretical Computer Science*, vol. 660, pp. 102–104, 2017.

- [158] T. Ansill, B. Jacob, J. Penzellna, and D. Saavedra, “Failed skew zero forcing on a graph,” *Linear Algebra and its Applications*, vol. 509, pp. 40–63, 2016.
- [159] C. X. Kang and E. Yi, “Probabilistic zero forcing in graphs,” *Bulletin of the Institute of Combinatorics and its Applications*, vol. 67, pp. 9–16, 2013.
- [160] D. Amos, Y. Caro, R. Davila, and R. Pepper, “Upper bounds on the k-forcing number of a graph,” *Discrete Applied Mathematics*, vol. 181, pp. 1–10, 2015.
- [161] L. Lu, B. Wu, and Z. Tang, “Proof of a conjecture on the zero forcing number of a graph,” *Discrete Applied Mathematics*, vol. 213, pp. 233–237, 2016.
- [162] R. Davila and M. Henning, “Total forcing sets in trees,” *arXiv:1702.06496*, 2017.
- [163] R. Davila and M. A. Henning, “On the total forcing number of a graph,” *arXiv:1702.06035*, 2017.
- [164] A. Berliner, C. Brown, J. Carlson, N. Cox, L. Hogben, J. Hu, K. Jacobs, K. Manternach, T. Peters, N. Warnberg, *et al.*, “Path cover number, maximum nullity, and zero forcing number of oriented graphs and other simple digraphs,” *Involve, a Journal of Mathematics*, vol. 8, no. 1, pp. 147–167, 2014.
- [165] A. Berliner, M. Catral, L. Hogben, K. Lied, and M. Young, “Minimum rank, maximum nullity, and zero forcing number of simple digraphs,” *Electronic Journal of Linear Algebra*, vol. 26, no. 1, p. 52, 2013.
- [166] L. Hogben, “Minimum rank problems,” *Linear Algebra and its Applications*, vol. 432, no. 8, pp. 1961–1974, 2010.

- [167] E. Camby, J. Cardinal, S. Fiorini, and O. Schaudt, “The price of connectivity for vertex cover,” *Discrete Mathematics and Theoretical Computer Science*, vol. 16, pp. 207–223, 2014.
- [168] F. Delbot, C. Laforest, and S. Rovedakis, “Self-stabilizing algorithms for connected vertex cover and clique decomposition problems,” in *International Conference on Principles of Distributed Systems*, pp. 307–322, Springer, 2014.
- [169] P. A. Golovach, P. Heggernes, and D. Kratsch, “Enumeration and maximum number of minimal connected vertex covers in graphs,” in *International Workshop on Combinatorial Algorithms*, pp. 235–247, Springer, 2015.
- [170] Y. Li, Z. Yang, and W. Wang, “Complexity and algorithms for the connected vertex cover problem in 4-regular graphs,” *Applied Mathematics and Computation*, vol. 301, pp. 107–114, 2017.
- [171] X. Li, Z. Zhang, and X. Huang, “Approximation algorithm for the minimum connected k-path vertex cover problem,” in *International Conference on Combinatorial Optimization and Applications*, pp. 764–771, Springer, 2014.
- [172] H. Fernau, F. V. Fomin, G. Philip, and S. Saurabh, “On the parameterized complexity of vertex cover and edge cover with connectivity constraints,” *Theoretical Computer Science*, vol. 565, pp. 1–15, 2015.
- [173] E. Sampathkumar and H. B. Walikar, “The connected domination number of a graph,” *Journal of Mathematical and Physical Sciences*, vol. 13, no. 6, pp. 607–613, 1979.
- [174] Y. Caro, D. B. West, and R. Yuster, “Connected domination and spanning

- trees with many leaves,” *SIAM Journal on Discrete Mathematics*, vol. 13, no. 2, pp. 202–211, 2000.
- [175] W. J. Desormeaux, T. W. Haynes, and M. A. Henning, “Bounds on the connected domination number of a graph,” *Discrete Applied Mathematics*, vol. 161, no. 18, pp. 2925–2931, 2013.
- [176] F. V. Fomin, F. Grandoni, and D. Kratsch, “Solving connected dominating set faster than 2^n ,” *Algorithmica*, vol. 52, no. 2, pp. 153–166, 2008.
- [177] N. Fan and J.-P. Watson, “Solving the connected dominating set problem and power dominating set problem by integer programming,” in *International Conference on Combinatorial Optimization and Applications*, pp. 371–383, Springer, 2012.
- [178] M. R. Garey and D. S. Johnson, *Computers and intractability*. W.H. Freeman & Co., San Francisco, 1979.
- [179] J. v. Rooij, J. Nederlof, and T. v. Dijk, “Inclusion/exclusion meets measure and conquer: Exact algorithms for counting dominating set,” in *17th Annual European Symposium on Algorithms*, pp. 554–565, Springer, 2009.
- [180] H. Fernau, J. Kneis, D. Kratsch, A. Langer, M. Liedloff, D. Raible, and P. Rossmanith, “An exact algorithm for the maximum leaf spanning tree problem,” *Theoretical Computer Science*, vol. 412, no. 45, pp. 6290–6302, 2011.
- [181] K. Meeks and A. Scott, “Spanning trees and the complexity of flood-filling games,” *Theory of Computing Systems*, vol. 54, no. 4, pp. 731–753, 2014.

- [182] C. Fast, *Novel Techniques for the Zero-Forcing and p -Median Graph Location Problems*. PhD thesis, Rice University, 2017.
- [183] D. B. West, *Introduction to Graph Theory*, vol. 2. Prentice Hall Upper Saddle River, 2001.
- [184] L. A. Goldberg and M. Jerrum, “Inapproximability of the Tutte polynomial,” in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pp. 459–468, ACM, 2007.
- [185] C. Wakelin and D. R. Woodall, “Chromatic polynomials, polygon trees, and outerplanar graphs,” *Journal of Graph Theory*, vol. 16, no. 5, pp. 459–466, 1992.
- [186] E. G. Whitehead, “Chromatic polynomials of generalized trees,” *Discrete Mathematics*, vol. 72, no. 1, pp. 391–393, 1988.
- [187] E. G. Whitehead, “Chromaticity of two-trees,” *Journal of Graph Theory*, vol. 9, no. 2, pp. 279–284, 1985.
- [188] E. Lazuka, “On chromaticity of graphs,” *Discussiones Mathematicae Graph Theory*, vol. 15, no. 1, pp. 19–31, 1995.
- [189] G. Gordon, “Chromatic and Tutte polynomials for graphs, rooted graphs and trees,” *Graph Theory Notes for New York LIV*, pp. 39–50, 2008.
- [190] E. Mphako-Banda, “Some polynomials of flower graphs,” *International Mathematical Forum*, vol. 2, no. 51, pp. 2511–2118, 2007.
- [191] E. Mphako-Banda, “Tutte polynomials of flower graphs,” *Bulletin of the Iranian Mathematical Society*, vol. 35, no. 2, pp. 179–190, 2009.

- [192] R. E. Tarjan, “A note on finding the bridges of a graph,” *Information Processing Letters*, vol. 2, no. 6, pp. 160–161, 1974.
- [193] K. Koh and C. Teo, “The chromatic uniqueness of certain broken wheels,” *Discrete mathematics*, vol. 96, no. 1, pp. 65–69, 1991.
- [194] H. Shahmohamad, “A survey on flow polynomial,” *Utilitas Mathematica Publishing Incorporated*, vol. 62, pp. 13–32, 2002.
- [195] M. M. Sysło, “Characterizations of outerplanar graphs,” *Discrete Mathematics*, vol. 26, no. 1, pp. 47–53, 1979.
- [196] J. Hopcroft and R. Tarjan, “Algorithm 447: Efficient algorithms for graph manipulation,” *Communications of the ACM*, vol. 16, no. 6, pp. 372–378, 1973.
- [197] J. Hopcroft and R. Tarjan, “Efficient planarity testing,” *Journal of the ACM*, vol. 21, no. 4, pp. 549–568, 1974.
- [198] F. A. Taklimi, “Zero forcing sets for graphs,” *arXiv:1311.7672*, 2013.
- [199] P. M. Nylén, “Minimum-rank matrices with prescribed graph,” *Linear Algebra and its Applications*, vol. 248, pp. 303–316, 1996.
- [200] K. D. Owens, “Properties of the zero forcing number,” Master’s thesis, Brigham Young University, 2009.
- [201] R. Davila, M. Henning, C. Magnant, and R. Pepper, “Bounds on the connected forcing number of a graph,” *arXiv:1605.02124*, 2016.
- [202] L. Lu, B. Wu, and Z. Tang, “Proof of a conjecture on the zero forcing number of a graph,” *Discrete Applied Mathematics*, vol. 213, pp. 233–237, 2016.

- [203] B. Yang, “Fast-mixed searching and related problems on graphs,” *Theoretical Computer Science*, vol. 507, pp. 100–113, 2013.
- [204] F. Boesch, S. Chen, and J. McHugh, “On covering the points of a graph with point disjoint paths,” in *Graphs and Combinatorics*, pp. 201–212, Springer, 1974.
- [205] S. Goodman and S. Hedetniemi, “On the hamiltonian completion problem,” in *Graphs and Combinatorics*, pp. 262–272, Springer, 1974.
- [206] C. R. Johnson and A. L. Duarte, “The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree,” *Linear and Multilinear Algebra*, vol. 46, no. 1-2, pp. 139–144, 1999.
- [207] D. D. Row, *Zero forcing number: Results for computation and comparison with other graph parameters*. PhD thesis, Iowa State University, 2011.
- [208] W. Barrett, H. Van Der Holst, and R. Loewy, “Graphs whose minimal rank is two: the finite fields case,” *Electronic Journal of Linear Algebra*, vol. 14, no. 1, p. 4, 2005.
- [209] W. Barrett, J. Grout, and R. Loewy, “The minimum rank problem over the finite field of order 2: minimum rank 3,” *Linear Algebra and its Applications*, vol. 430, no. 4, pp. 890–923, 2009.
- [210] H. van der Holst, “Graphs whose positive semi-definite matrices have nullity at most two,” *Linear Algebra and its Applications*, vol. 375, pp. 1–11, 2003.
- [211] H. van der Holst, “Three-connected graphs whose maximum nullity is at most three,” *Linear Algebra and its Applications*, vol. 429, no. 2-3, pp. 625–632, 2008.

- [212] B. Korte and L. Lovász, “Mathematical structures underlying greedy algorithms,” in *Fundamentals of Computation Theory*, pp. 205–209, Springer, 1981.
- [213] J. G. Oxley, *Matroid theory*, vol. 3. Oxford University Press, USA, 2006.
- [214] M. D. Plummer, “Some covering concepts in graphs,” *Journal of Combinatorial Theory*, vol. 8, no. 1, pp. 91–98, 1970.
- [215] D. Tankus and M. Tarsi, “The structure of well-covered graphs and the complexity of their recognition problems,” *Journal of Combinatorial Theory, Series B*, vol. 69, no. 2, pp. 230–233, 1997.
- [216] M. D. Plummer, “Well-covered graphs: a survey,” *Quaestiones Mathematicae*, vol. 16, no. 3, pp. 253–287, 1993.
- [217] P. Hall, “On representatives of subsets,” *Journal of the London Mathematical Society*, vol. 10, no. 1, pp. 26–30, 1935.
- [218] B. Brimkov and I. V. Hicks, “Chromatic and flow polynomials of generalized vertex join graphs and outerplanar graphs,” *Discrete Applied Mathematics*, vol. 204, pp. 13–21, 2016.
- [219] P. Berthomé, S. Lebresne, and K. Nguy, “Computation of chromatic polynomials using triangulations and clique trees,” in *International Workshop on Graph-Theoretic Concepts in Computer Science*, pp. 362–373, Springer, 2005.
- [220] S. B. Ekhad, J. Quaintance, and D. Zeilberger, “Automatic generation of generating functions for chromatic polynomials for grid graphs (and more general creatures) of fixed (but arbitrary!) width,” *arXiv:1103.6206*, 2011.

- [221] B. Brimkov and R. Davila, “Characterizations of the connected forcing number of a graph,” *arXiv:1604.00740*, 2016.
- [222] B. Brimkov, “Complexity and computation of connected zero forcing,” *arXiv:1607.00658*, 2016.
- [223] B. Brimkov, C. C. Fast, and I. V. Hicks, “Graphs with extremal connected forcing numbers,” *arXiv:1701.08500*, 2017.