Spring 2010

Graphical Indices and their Applications

Daniel Gray

Recommended Citation

Gray, Daniel, "Graphical Indices and their Applications" (2010). *Electronic Theses and Dissertations*. 659.
https://digitalcommons.georgiasouthern.edu/etd/659

Version: April 21, 2010

GRAPHICAL INDICES AND THEIR APPLICATIONS

by

DANIEL GRAY

(Under the Direction of Dr. Hua Wang)

ABSTRACT

The biochemical community has been using graphical (topological, chemical) indices in the study of Quantitative Structure-Activity Relationships (QSAR) and Quantitative Structure-Property Relationships (QSPR), as they have been shown to have strong correlations with the chemical properties of certain chemical compounds (i.e. boiling point, surface area, etc.). We examine some of these chemical indices and closely related pure graph theoretical indices: the Randić index, the Wiener index, the degree distance, and the number of subtrees. We find which structure will maximize the Randić index of a class of graphs known as *cacti*, and we find a functional relationship between the Wiener index and the degree distance for several types of graphs. We also develop an algorithm to find the structure that maximizes the number of subtrees of trees, a characterization of the second maximal tree may also follow as an immediate result of this algorithm.

Key Words: Trees, subtrees, cacti, Randić index, Wiener index, degree distance

2010 *Mathematics Subject Classification*: 05C05, 05C35, 05C38, 05C90

GRAPHICAL INDICES AND THEIR APPLICATIONS

by

DANIEL GRAY

B.S. in Mathematics at Augusta State

A Thesis Submitted to the Graduate Faculty of Georgia Southern University in Partial Fulfillment of the Requirement for the Degree

MASTER OF SCIENCE
IN APPLIED MATHEMATICS

STATESBORO, GEORGIA

2010

GRAPHICAL INDICES AND THEIR APPLICATIONS

by

DANIEL GRAY

Major Professor:	Dr. Hua Wang
Committee:	Dr. Goran Lesaja
	Dr. Yan Wu

Electronic Version Approved:

May 2010

ACKNOWLEDGMENTS

I wish to acknowledge Dr. Hua Wang for his exceptional aid and skill in helping me to prepare this document, and for the knowledge imparted unto me through his advisement. I also wish to acknowledge the staff and faculty of the math department that I have had the pleasure of meeting, working with, or being a student of.

In particular, I would like to thank Dr. Martha Abell for the financial support I received for travel to many of the conferences I attended during my graduate studies at Georgia Southern University, and the other members of my thesis committee, Dr. Goran Lesaja and Dr. Yan Wu, for helping me to improve this document as much as possible despite their very busy schedules. Of the many talented students in the department, thanks to Chason Smith, James Bland, and Jason Brandies for the tips and tricks of Latex, Beamer, and Tikz.

Finally, I am grateful to my family for supporting me in all of my academic endeavors.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION AND TERMINOLOGY

All graphs in this thesis are finite, simple, and undirected. The *order* of a graph, G, is the number of vertices in its vertex set, $V(G)$. The *size* of G is the number of edges in its edge set, $E(G)$. If two vertices share an edge then they are *adjacent*; we also say that the vertices are *incident* with the edge. The *degree* of a vertex, $d_G(v)$ where $v \in V(G)$, is the number of edges that vertex is incident with (i.e. the number of vertices it is adjacent to). A *tree*, T, is a connected, acyclic (no cycles) graph. A *cactus*, G is a graph wherein no two cycles can share a common edge.

A graphical index is a numerical value which describes the topology of a graph. The numerical value can depend on the degrees of the vertices, distance between vertices, or any combination of characteristics of a graph.

The structure of a chemical compound is usually modeled as a polygonal shape, which is often called the *molecular graph* of this compound. To obtain the molecular graph of a chemical compound, remove all hydrogen atoms and let each of the remaining atoms be a vertex in the graph. We draw an edge between two vertices if their corresponding atoms are bonded in the chemical compound (See Figure 1.1). The biochemical community has been using graphical indices in the study of Quantitative Structure-Activity Relationships (QSAR) and Quantitative Structure-Property Relationships (QSPR), as they have been shown to have strong correlations with the chemical properties of certain chemical compounds (i.e. boiling point, surface area, etc.).

In this thesis, we explore the relationship between some of these graphical indices, as well as characterize the structures which achieve the extremal values with

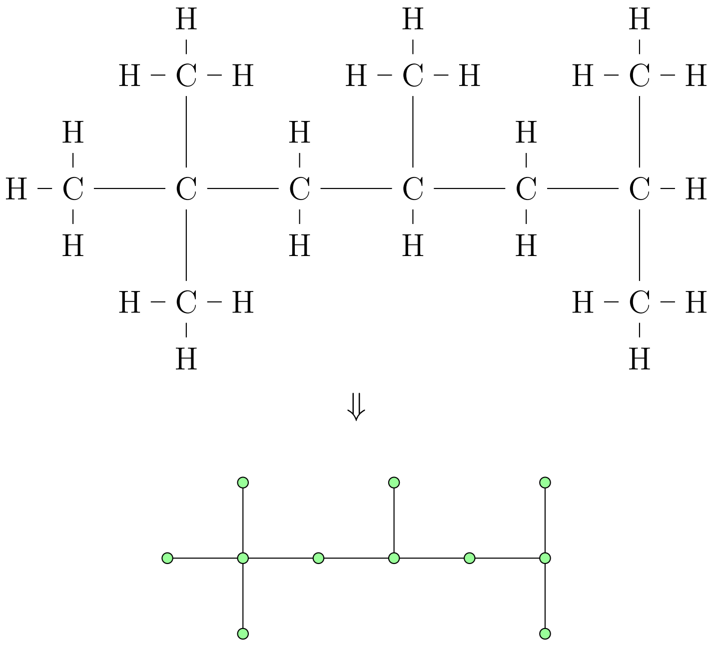


Figure 1.1: Obtaining the molecular graph of a chemical compound.

respect to these graphical indices. The research presented here has led to three seperate manuscripts, [9] has been published, [10] has been submitted, and [11] is to be submitted in the near future.

1.1 Randić index

In a graph G, let $d(v)$ denote the degree of the vertex v in G. The *Randić index* of G is

$$w_\alpha(G) = \sum_{uv \in E(G)} (d(u)d(v))^\alpha$$

where $E(G)$ is the edge set of G and $\alpha \neq 0$. This index is named after Milan Randić, who introduced $w_{-1}(G)$ and $w_{-1/2}(G)$ in [22]. The Randić index, also called the *connectivity index*, has been vigorously studied in recent years (see [4] and the reference there for detail). In pure graph theory, for a tree, T, $w_1(T)$ is also called the weight of T and has been well studied.

Being one of the most important topological indices in bio-chemistry, the bounds of the value of the Randić index are studied for general graphs [1], [18], for trees in general [20], trees with restricted degrees [23] and trees with given degree sequence [3], [28]. See [16] for a survey of the extremal structures for the Randić index for graphs under various restrictions.

In Chapter 2, we study the structure that maximizes the Randić index for $\alpha \in (0, 1]$ and minimizes the Randić index for $\alpha \in [-1, 0)$, which leads to a formula to easily calculate the Randić index of extremal cacti. These results have formed an accepted article [9].

1.2 Wiener index and degree distance

The *Wiener index* of a graph G is the sum of the distances between all pairs of vertices, denoted by

$$W(G) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} L_{i,j}$$

where $L_{i,j}$ is the distance between two vertices $v_i, v_j \in V(G) := \{v_1, v_2, \ldots, v_n\}$. The Wiener index, $W(G)$, was introduced by and named after H. Wiener [30] and is one of the most classic and well-studied concepts in the study of QSAR/QSPR.

Dobrynin and Kochetova [6] introduced the degree distance as a 'degree analogue of the Wiener index', which is defined by

$$D'(G) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} ((d_i + d_j)L_{i,j}) \tag{1.1}$$

where d_i is the degree of the vertex v_i. Upon realizing that the distance matrix L is symmetric, we can rewrite the equation into a more convenient form,

$$D'(G) = \sum_{i=1}^{n} \sum_{j=1}^{n} (d_i L_{i,j}) . \tag{1.2}$$

We will use the second equation for the degree distance throughout this thesis.

In Chapter 3, we derive a functional relationship between $D'(G)$ and $W(G)$ for various kinds of graphs by introducing an error term $e(G)$. We provide exact equations to calculate $e(G)$ for the different graphs and we study the bounds of $e(G)$ to obtain simplified formulas. A manuscript containing these results has been submitted [10].

1.3 Subtrees of trees

Let G be a tree, the number of subtrees of G is denoted by $f(G)$ (i.e. the number of possible induced substructures of G). In pure graph theory, subtrees of trees has been extensively studied.

The relationship between the total number of subtrees and the Wiener Index is of particular importance. It is known that among binary trees of the same size, the extremal one that minimizes the Wiener Index is exactly the one that maximizes the number of subtrees, and vice versa. In [25], it is proven that the same kind of extremal trees, ones that minimize the Wiener Index and maximize the number of subtrees among binary trees of the same size, also maximize the number of leaf-containing subtrees, where a leaf containing subtree is a subtree of a tree T that contains leaf vertices (i.e. vertices of degree 1) of T.

It is also known that among all trees of the same size and same maximum degree, the structure that minimizes the Wiener index will maximize the number of subtrees.

See [8, 13, 14].

The number of subtrees of binary trees and the Wiener Index are studied in [24], but a counterexample shows that no functional relation exists between them.

In Chapter 4, we construct algorithms to find the properties that a maximal tree must possess, and in Chapter 5, we use these algorithms to prove that the tree that has these properties is unique, thus proving that the tree obtained from the algorithm is the maximal tree. We then prove several lemmas in order to characterize the second maximal tree. These results are from a work in progress [11].

CHAPTER 2

RANDIĆ INDEX AND EXTREMAL CACTI

Let us recall the definition of the *Randić index* of a graph G. It is defined as

$$w_\alpha(G) = \sum_{uv \in E(G)} (d(u)d(v))^\alpha,$$

where $d(u)$ denotes the degree of a vertex u, $E(G)$ denotes the edge set of G, and $\alpha \in [-1, 0) \bigcup (0, 1]$.

A *cut vertex* is a vertex whose removal will disconnect the graph (i.e. there exists a pair of vertices with no path between them). A *block* is a maximal biconnected (no cut vertices) subgraph of a graph. A graph is called a *cactus* if each block is either an edge or a cycle. Throughout this thesis, let $\mathcal{G}(n, r, s)$ denote the set of all cacti with n vertices, r 'edge-blocks' and s 'cycle-blocks', let $G(n, r, s)$ denote all cacti in $\mathcal{G}(n, r, s)$ with only one cut vertex (the vertex shared by all blocks). Given a cactus in $G(n, r, s)$, the edges incident to *leaves* (vertices of degree 1) are called *pendant edges*. The vertex shared by all blocks is called the *center* of G and is denoted by v in Fig. 2.1.

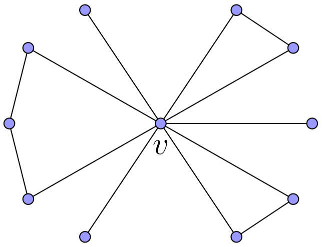


Figure 2.1: An example of $G(11, 3, 3)$

Because of the appearance of cactus graphs in various molecular structures, the Randić index of cacti has been studied recently. In [19], a sharp lower bound on the

Randić index of cacti with given number of cycles is provided. In [17], a sharp lower bound of $w_{-1/2}(G)$ is provided for cacti with given number of vertices and pendant edges. A natural project is to consider the extremal problems for the generalized Randić index, which seems to be much more difficult because of the infinite possibilities of α. In this chapter, we provide, in Section 2.1, a result on the structure of extremal cacti (Theorem 2.1.1). This result is of general interest in addition to its role in this work.

Furthermore, another case is considered in Section 2.2, where we propose the conjecture that an extremal cactus in $\mathcal{G}(n, r, s)$ must be in $G(n, r, s)$. A simple formula of $w_\alpha(G)$ for graphs in $G(n, r, s)$ follows in Section 2.3, which leads to a straightforward characterization of extremal cacti with various restrictions for the weight of a graph.

2.1 The structures of extremal cacti in $\mathcal{G}(n, r, s)$

For convenience we focus on the extremal cactus in $\mathcal{G}(n, r, s)$ that maximizes $w_\alpha(G)$ for positive α; the case for cacti that minimize $w_\alpha(G)$ with negative α is similar. We call this cactus *optimal*.

To show that the optimal cacti must be in $G(n, r, s)$, assume the contrary: the optimal cactus has at least two cut vertices. There are two cases to consider:

Case I: there are two cut vertices connected by a cut edge;

Case II: there are two cut vertices lying on a cycle.

2.1.1 Case I

First we show the following result that prevents 'Case I' from an optimal cacti:

Theorem 2.1.1. *An optimal cactus in $\mathcal{G}(n, r, s)$ contains no cut edge unless it is a pendant edge.*

Proof. Assume otherwise; let uw be such a cut edge. Let U and W be the components that contain u and w respectively after the removal of the edges on all the paths connecting u and w (Fig. 2.2).

Without loss of generality, assume $d(u) \geq d(w)$ and let H be the cactus obtained from G by 'moving' W from w to u (Fig. 2.2). Obviously $\{n, r, s\}$ stays the same before and after this operation. We will show that $w_\alpha(H) > w_\alpha(G)$ by studying the terms $(d(v_1)d(v_2))^\alpha$ for all the edges xy incident with u or w. For other edges $(d(v_1)d(v_2))^\alpha$ will not change.

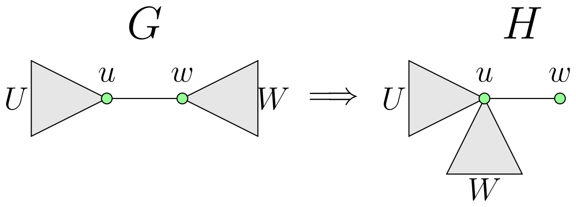


Figure 2.2: The cacti G (on the left) and H (on the right).

For the simplification of notations, we introduce the following:

i) $d'(u)$ $(d'(w))$: the degree of u (w) in U (W), noticing that $d(u) - d'(u) = 1$ $(d(w) - d'(w) = 1)$;

ii) A (B): the sum of the αth powers of the degrees of vertices adjacent to u (w)

in U (W) (notice that $A \geq d'(u) = d(u) - 1$ and $B \geq d'(w) = d(w) - 1$).

We have

$$\begin{aligned} & \omega_\alpha(H) - \omega_\alpha(G) \\ = \; & (A + B + 1)(d(u) + d(w) - 1)^\alpha \\ & -A(d(u))^\alpha - B(d(w))^\alpha - (d(u)d(w))^\alpha \\ = \; & A((d(u) + d(w) - 1)^\alpha - d(u)^\alpha) \\ & +B((d(u) + d(w) - 1)^\alpha - d(w)^\alpha) - (d(u)d(w))^\alpha. \end{aligned}$$

Notice that $(d(u) + d(w) - 1)^\alpha - d(u)^\alpha > 0$ and $(d(u) + d(w) - 1)^\alpha - d(w)^\alpha > 0$, and recall that $A \geq d(u) - 1$ and $B \geq d(w) - 1$. We obtain,

$$\begin{aligned} & \omega_\alpha(H) - \omega_\alpha(G) \\ \geq \; & (d(u) - 1)((d(u) + d(w) - 1)^\alpha - d(u)^\alpha) \\ & +(d(w) - 1)((d(u) + d(w) - 1)^\alpha - d(w)^\alpha) - (d(u)d(w))^\alpha \\ \geq \; & (d(u) + d(w) - 1)(d(u) + d(w) - 1)^\alpha \\ & -(d(u) - 1)(d(u))^\alpha - (d(w) - 1)(d(w))^\alpha - (d(u)d(w))^\alpha. \end{aligned}$$

For simplicity we now let $x = d(u)$ and $y = d(w)$. Thus,

$$\begin{aligned} & \omega_\alpha(H) - \omega_\alpha(G) \\ \geq \; & (x + y - 1)(x + y - 1)^\alpha - (x - 1)x^\alpha - (y - 1)y^\alpha - (xy)^\alpha. \end{aligned}$$

With the restriction $d(u) = x \geq y = d(w)$, we obtain

$$\begin{aligned} & \omega_\alpha(H) - \omega_\alpha(G) \\ \geq \; & (x + y - 1)(x + y - 1)^\alpha - (x + y - 2)x^\alpha - (xy)^\alpha := f(\alpha). \end{aligned}$$

It is easily verified that $f(0) = 0$ and $f(1) = xy - (x + y - 1) > 0$ for x and y greater than 2 (since the degrees of u and w are at least 2). Rewrite $f(\alpha)$ in the following way,

$$f(\alpha) = (x + y - 1)(x + y - 1)^\alpha(1 - C(\alpha) - D(\alpha)), \tag{2.1}$$

where $C(\alpha) = \frac{(x+y-2)x^{\alpha}}{(x+y-1)(x+y-1)^{\alpha}}$ and $D(\alpha) = \frac{(xy)^{\alpha}}{(x+y-1)(x+y-1)^{\alpha}}$.

If we can show that $C(\alpha) + D(\alpha) \leq 1$ on $\alpha \in [0, 1]$ then we are done. Consider the derivative of $C(\alpha) + D(\alpha)$, we have

$$\begin{aligned} & \tfrac{d}{d\alpha}(C(\alpha) + D(\alpha)) \\ = \; & \tfrac{x^{\alpha}(x+y-2)ln(\frac{x}{x+y-1})}{(x+y-1)(x+y-1)^{\alpha}} + \tfrac{(xy)^{\alpha}ln(\frac{xy}{x+y-1})}{(x+y-1)(x+y-1)^{\alpha}} \\ = \; & \tfrac{-x^{\alpha}(x+y-2)ln(\frac{x+y-1}{x})+(xy)^{\alpha}ln(\frac{xy}{x+y-1})}{(x+y-1)(x+y-1)^{\alpha}} \end{aligned} \tag{2.2}$$

Clearly, the denominator in (2.2) is always positive. We seek to find an interval where $\frac{d}{d\alpha}(C(\alpha) + D(\alpha))$ is negative (i.e. when the numerator of (2.2) is negative). This will happen if and only if

$$\frac{(xy)^{\alpha}ln(\frac{xy}{x+y-1})}{(x+y-1)(x+y-1)^{\alpha}} < \frac{x^{\alpha}(x+y-2)ln(\frac{x+y-1}{x})}{(x+y-1)(x+y-1)^{\alpha}} \tag{2.3}$$

which is equivalent to

$$\alpha < \frac{ln(\frac{(x+y-2)ln(\frac{x+y-1}{x})}{ln(\frac{xy}{x+y-1})})}{ln(y)} := \alpha_0.$$

Then, for $\alpha < \alpha_0$ the derivative of $C(\alpha) + D(\alpha)$ is strictly negative, and for $\alpha \geq \alpha_0$ the derivative is nonnegative. Recall that $f(0) = 0$ and $f(1) > 0$, which implies that $1 = C(0) + D(0)$ and $1 > C(1) + D(1)$; thus, $C(0) + D(0) > C(1) + D(1)$. Then the derivative is negative somewhere in the interval from 0 to 1. Therefore, $C(\alpha) + D(\alpha)$ is monotonically decreasing on the interval $\alpha \in [0, \alpha_0)$ and is nondecreasing on the interval $\alpha \in [\alpha_0, 1]$. Hence, there can be no $\alpha \in [0, 1]$ such that $C(\alpha) + D(\alpha) > 1$. This proves that $\omega_{\alpha}(H) - \omega_{\alpha}(G) \geq f(\alpha) > 0$ on $(0, 1]$.

$\square$

2.1.2 Case II

Using similar notations as in Section 2.1, assume now that u and w are cut vertices of G lying on the same cycle. Then, let H (H′) be obtained from G by moving the component W (U)to u (w). See Fig. 2.3 below.

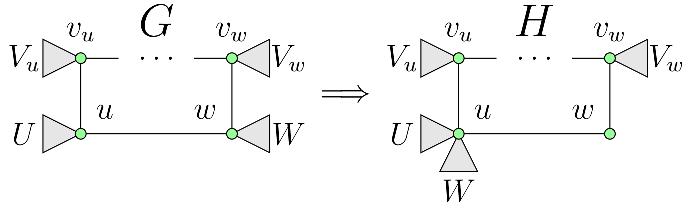


Figure 2.3: The cacti G (on the left) and H (on the right).

We now compare $w_\alpha(G)$ with $w_\alpha(H)$ and $w_\alpha(H')$. Let $x = d(u)$, $y = d(w)$, $v_x = d(v_u)$ and $v_y = d(v_w)$. Then $A \geq x - 2$ and $B \geq y - 2$. Without loss of generality, let u be the vertex with the largest degree. Then

$$\begin{aligned}
& w_\alpha(H) - w_\alpha(G) \\
= \;& (A + B + (v_x)^\alpha)(x + y - 2)^\alpha + (2v_y)^\alpha + (2(x + y - 2))^\alpha \\
& -(A + (v_x)^\alpha)x^\alpha - (B + (v_y)^\alpha)y^\alpha - (xy)^\alpha \\
= \;& A((x + y - 2)^\alpha - x^\alpha) + B((x + y - 2)^\alpha - y^\alpha) - (xy)^\alpha \\
& +(v_x)^\alpha((x + y - 2)^\alpha - x^\alpha) - (v_y)^\alpha(y^\alpha - 2) + (2(x + y - 2))^\alpha.
\end{aligned}$$

Recall the restrictions $A \geq x - 2$ and $B \geq y - 2$, and suppose $v_x \geq v_y$. We obtain

$$\begin{aligned}
& w_\alpha(H) - w_\alpha(G) \\
\geq \;& (x - 2)((x + y - 2)^\alpha - x^\alpha) + (y - 2)((x + y - 2)^\alpha - y^\alpha) - (xy)^\alpha \\
& +v_x^\alpha((x + y - 2)^\alpha - x^\alpha) - v_y^\alpha(y^\alpha - 2^\alpha) + (2(x + y - 2))^\alpha \\
\geq \;& (x + y - 4)(x + y - 2)^\alpha - (x - 2)x^\alpha - (y - 2)y^\alpha - (xy)^\alpha \\
& +(2(x + y - 2))^\alpha + v^\alpha((x + y - 2)^\alpha + 2^\alpha - x^\alpha - y^\alpha),
\end{aligned}$$

where $v = v_x$. Similarly,

$$\begin{aligned}
& \omega_\alpha(H') - \omega_\alpha(G) \\
= \;& (A + B + (v_y)^\alpha)(x + y - 2)^\alpha + (2v_x)^\alpha + (2(x + y - 2))^\alpha \\
& -(A + (v_x)^\alpha)x^\alpha - (B + (v_y)^\alpha)y^\alpha - (xy)^\alpha \\
= \;& A((x + y - 2)^\alpha - x^\alpha) + B((x + y - 2)^\alpha - y^\alpha) - (xy)^\alpha \\
& +(v_y)^\alpha((x + y - 2)^\alpha - y^\alpha) - (v_y)^\alpha(x^\alpha - 2^\alpha) + (2(x + y - 2))^\alpha.
\end{aligned}$$

This time suppose $v_y \geq v_x$. Then,

$$\begin{aligned}
& \omega_\alpha(H') - \omega_\alpha(G) \\
\geq \;& (x - 2)((x + y - 2)^\alpha - x^\alpha) + (y - 2)((x + y - 2)^\alpha - y^\alpha) - (xy)^\alpha \\
& +v_y^\alpha((x + y - 2)^\alpha - y^\alpha) - v_x^\alpha(x^\alpha - 2^\alpha) + (2(x + y - 2))^\alpha \\
\geq \;& (x + y - 4)(x + y - 2)^\alpha - (x - 2)x^\alpha - (y - 2)y^\alpha - (xy)^\alpha \\
& +(2(x + y - 2))^\alpha + v^\alpha((x + y - 2)^\alpha + 2^\alpha - x^\alpha - y^\alpha),
\end{aligned}$$

where $v = v_y$.

Hence, from both cases we have either $\omega_\alpha(H) - \omega_\alpha(G)$ or $\omega_\alpha(H') - \omega_\alpha(G)$ at least as large as

$$\begin{aligned}
& (x + y - 4)(x + y - 2)^\alpha - (x - 2)x^\alpha - (y - 2)y^\alpha - (xy)^\alpha \\
+ \;& (2(x + y - 2))^\alpha + v^\alpha((x + y - 2)^\alpha + 2^\alpha - x^\alpha - y^\alpha).
\end{aligned} \tag{2.4}$$

If one can show that (2.4) is positive for $\alpha \in [0, 1]$, then we will have $\omega_\alpha(H) > \omega_\alpha(G)$ or $\omega_\alpha(H') > \omega_\alpha(G)$. Notice that the last term in (2.4) is always negative for $\alpha \in (0, 1)$. When $v = x$, i.e. the worst case scenario, we can reduce the formula to a function of only three variables x,y, and α.

This leads to the following seemingly simple conjecture, the verification of which is deceptively difficult:

Conjecture 2.1.2. *The function*

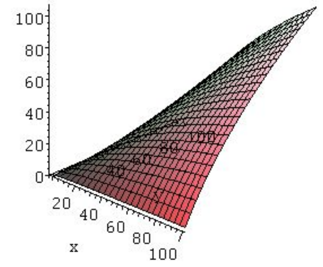
$$\begin{aligned} & (x+y-4)(x+y-2)^\alpha-(x-2)x^\alpha-(y-2)y^\alpha-(xy)^\alpha \\ + \quad & (2(x+y-2))^\alpha+x^\alpha((x+y-2)^\alpha+2^\alpha-x^\alpha-y^\alpha). \end{aligned}$$

is always positive for $\alpha \in (0,1]$ and integers $x \geq y \geq 3$.

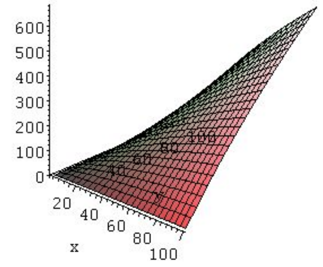
Of course, with Theorem 2.1.1, Conjecture 2.1.2 implies the aforementioned conjecture:

Conjecture 2.1.3. *If a cactus $G \in \mathcal{G}(n,r,s)$ is extremal, i.e. maximizes the Randić index for positive α or minimizes the Randić index for negative α, then $G \in G(n,r,s)$.*

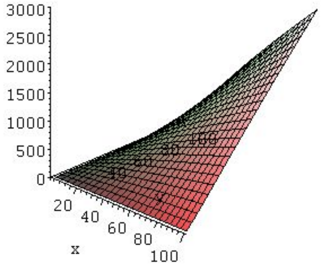
Remark: Notice that the approach introduced in this section and the formula in Conjecture 2.1.2 make it fairly straightforward to prove Conjecture 2.1.3. For instance, below is a list of 3d-plots of (2.4) with $v = x$ and $\alpha = \frac{1}{4}$, $\frac{1}{2}$, $\frac{3}{4}$, and 1, respectively, agreeing with our conjecture.



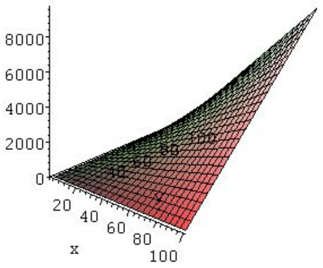
$\alpha = \frac{1}{4}$



$\alpha = \frac{1}{2}$



$\alpha = \frac{3}{4}$



$\alpha = 1$

2.2 Calculating $w_\alpha(G(n,r,s))$ and applications

Recall that $w_\alpha(G) = \sum_{uv \in E(G)} (d(u)d(v))^\alpha$. We consider $(d(u)d(v))^\alpha$ for each edge in a cactus $G(n,r,s)$ (Fig. 2.1). We observe the following:

1. the degree of the center is $d(v) = r + 2s$;
2. for each of the r pendant edges, we have $(r+2s)^\alpha$;
3. for each of the other $2s$ edges incident with the center, we have $(2(r+2s))^\alpha$;
4. there are $n - r - s - 1$ edges left, for each of them we have $((2)(2))^\alpha$.

Combining these observations, we obtain

$$\begin{aligned} & w_\alpha(G(n,r,s)) \\ = \; & r(r+2s)^\alpha + 2s(2(r+2s))^\alpha + (n-r-s-1)4^\alpha \\ = \; & (n-1)4^\alpha + r((r+2s)^\alpha - 4^\alpha) + s(2(2(r+2s))^\alpha - 4^\alpha) \\ = \; & N + R + S \end{aligned}, \tag{2.5}$$

where $N = (n-1)4^\alpha$, $R = r((r+2s)^\alpha - 4^\alpha)$ and $S = s(2(2(r+2s))^\alpha - 4^\alpha)$.

2.2.1 The case $\alpha = 1$

We restrict our attention to the weight in the remainder of this section, where we make the application of the approach discussed in section 2 explicit in the case $\alpha = 1$. Theorem 2.1.1 implies the following corollary.

Corollary 2.2.1. *The cactus that maximizes the weight in $\mathcal{G}(n,r,s)$ has no cut edge unless it is a pendant edge.*

Also, let $\alpha = 1$ and $v = v_x = v_y$ in (2.4), we have

$$\begin{aligned} & (x+y-2)(x+y-2) - (x+y-4)x - xy + 0 \\ = \; & x^2 + 2xy + y^2 - 4x - 4y + 4 - x^2 - xy + 4x - xy \\ = \; & y^2 - 4y + 4 \end{aligned}$$

which is positive for $y \geq 3$. Therefore, at least one of $w_1(H)$ and $w_1(H')$ is larger than $w_1(G)$ in section 2.2, thus by contradiction, we have the following lemma.

Lemma 2.2.2. *The cactus that maximizes the weight in $\mathcal{G}(n,r,s)$ can not have two cut vertices lying on the same cycle.*

Note: The case that two cut vertices lie on the same cycle but are not adjacent can be proved in exactly the same way, and we skip it here.

From Corollary 2.2.1 and Lemma 2.2.2, we get

Theorem 2.2.3. *The cactus that maximizes the weight in* $\mathcal{G}(n, r, s)$ *must be in* $G(n, r, s)$*.*

2.2.2 The extremal cacti

Based on the results in Sections 2.1.1 and 2.1.2, we discuss the extremal cacti under various restrictions. We restrict our attention to cacti with fixed n and $r + 2s \geq 5$. In the case $r + 2s \leq 4$, the cacti are very small and the argument is trivial. We also assume that $\alpha = 1$.

I) Cacti with given r:

In this case we have n vertices and r pendant edges (leaf vertices). From (2.5) we see that the weight is maximized when s is maximized. The number of cycles is maximized if every cycle is of minimum length. There are at least 3 edges, and thus 3 vertices (one of which will always be the center), in every cycle. Thus, we divide all vertices which are not the center and are not leaf vertices into groups of two.

I-a) general case:

If $n - r - 1$ is even, then s is maximized when we have $\frac{n-r-1}{2}$ cycles of length 3. If $n - r - 1$ is odd, then s is maximized when we have $\frac{n-r-4}{2}$ cycles of length 3, and one cycle of length 4.

I-b) with given minimum cycle length d:

In this case s is maximized when we have $\left\lfloor \frac{n-r-d}{d-1} \right\rfloor$ cycles of length d, and one extra cycle with remaining vertices (there are several possibilities here, we present a simple one). Recall that $\lfloor x \rfloor = n \in \mathbb{Z}$ is the the closest integer to $x \in \mathbb{R}$ such that $x > n$.

I-c) with given maximum cycle length d:

In this case, the result stays the same as in the general case.

II) Cacti with given s:

In this case we have n vertices and s cycles. From (2.5) we see that the weight is maximized when r is maximized. The number of pendant edges is maximized when every cycle is of minimum length. It is a well known result that the number of edges in a cactus is $n + s - 1$. The number of pendant edges is the total number of edges minus the number of edges lying in a cycle.

II-a) general case:

In this case, r is maximized when we have all s cycles being of length 3. Thus, $r = n - 2s - 1$.

II-b) with given minimum cycle length d

In this case, r is maximized when we have all s cycles being of minimum length d, i.e. $r = n - s(d-1) - 1$.

II-c) with given maximum cycle length d

In this case, the result stays the same as in the general case.

CHAPTER 3

WIENER INDEX VS. DEGREE DISTANCE

The *Wiener index* of a graph G is the sum of the distances between all pairs of vertices, denoted by

$$W(G) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} L_{i,j}$$

where $L_{i,j}$ is the distance between two vertices $v_i, v_j \in V(G) := \{v_1, v_2, \ldots, v_n\}$. The *degree distance* of a graph G, introduced as a degree analogue of the Wiener index [6], is denoted by

$$D'(G) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}((d_i + d_j)L_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{n}(d_i L_{i,j}), \tag{3.1}$$

where d_i and d_j are the degrees of v_i and v_j.

In the past two decades, the properties of $W(G)$ and $D'(G)$ of various types of graphs have been studied extensively. See for instance [2, 5, 7, 8, 21, 26, 27] and the references therein. In many cases, the extremal values of the Wiener index and teh degree distance are achieved by the same structures. This is not a surprise considering the similarity in their definitions. Hence, it is natural to consider the correlation between $W(G)$ and $D'(G)$. An elegant result for trees was achieved in [12] and independently in [15], and is stated in the theorem below.

Theorem 3.0.4. *$D'(T) = 4W(T) - n(n-1)$ for a tree T with n vertices.*

For completeness we present a short proof in the next section, which also serves as a starting point for similar study of other graphs in this chapter.

3.1 Preliminaries and Trees

We first provide a simple observation.

Lemma 3.1.1. *Given a graph G, the following equation holds:*

$$D'(G) = 4W(G) - 2n|E(G)| + n(n-1) + e(G),$$

where $e(G) = \sum_{i=1}^{n}(\sum_{\substack{j=1\\ j\neq i}}^{n}(d_i-1)(L_{i,j}+1)+d_i) - 2W(G)$ *is considered an 'error term'.*

Proof. From (3.1) we have

$$\begin{aligned} D'(G) &= \sum_{i=1}^{n}\sum_{j=1}^{n}\Big((d_i-1)(L_{i,j}+1)+L_{i,j}-d_i+1\Big) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\Big((d_i-1)(L_{i,j}+1)\Big)+\sum_{i=1}^{n}\sum_{j=1}^{n}L_{i,j}-\sum_{i=1}^{n}\sum_{j=1}^{n}d_i+\sum_{i=1}^{n}\sum_{j=1}^{n}1 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\Big((d_i-1)(L_{i,j}+1)\Big)+2W(G)-2n|E(G)|+n^2 \qquad (3.2) \end{aligned}$$

where $(d_i-1)(L_{i,j}+1)$ counts the distances between all neighbors of v_i with v_j (neglecting the one lying on the shortest path between v_i and v_j), except when $i=j$. In the case when $i=j$, $(d_i-1)(L_{i,i}+1)=d_i-1$ undercounts by 1 the number of vertices distance 1 away from v_i; to compensate for the undercounting, we rewrite (3.2) as

$$\begin{aligned} D'(G) &= \sum_{i=1}^{n}\Big(\sum_{\substack{j=1\\ j\neq i}}^{n}((d_i-1)(L_{i,j}+1))+d_i-1\Big)+2W(G)-2n|E(G)|+n^2 \\ &= \sum_{i=1}^{n}\Big(\sum_{\substack{j=1\\ j\neq i}}^{n}((d_i-1)(L_{i,j}+1))+d_i\Big)+2W(G)-2n|E(G)|+n(n-1). \end{aligned}$$

Then, for any two vertices x and y $(x\neq y)$ in $V(G)$, the sum

$$\sum_{i=1}^{n}\Big(\sum_{\substack{j=1\\ j\neq i}}^{n}(d_i-1)(L_{i,j}+1)+d_i\Big)=\sum_{i=1}^{n}\Big(\sum_{\substack{j=1\\ j\neq i}}^{n}(d_i-1)(L_{i,j}+1)\Big)+\sum_{i=1}^{n}d_i \qquad (3.3)$$

counts the distance between x and y once when v_i is a neighbor of x on the path between x and y and $v_j = y$, and once when v_i is a neighbor of y on the path between x and y and $v_j = x$. When d_i is summed over all i we get twice the number of edges, which double counts all pairs of vertices distance 1 apart. Hence, we can write

$$e(G) := \sum_{i=1}^{n} \Big(\sum_{\substack{j=1 \\ j\neq i}}^{n} (d_i - 1)(L_{i,j} + 1) + d_i \Big) - 2W(G)$$

□

Remark 3.1.2. *Through out this chapter, we will focus on the distances that are counted more than twice in (3.3), the sum of which gives us* $e(G)$.

Remark 3.1.3. *Note that the sum of distances of pairs of adjacent vertices are counted exactly twice by* $\sum_{i=1}^{n} d_i$. *So from now on we only need to consider the distances between non-adjacent vertices.*

In the case of trees, we will show that the distance between every pair of vertices is counted exactly twice in (3.3), or equivalently, $e(G) = 0$.

Proof. (of Theorem 3.0.4)

To see that (3.3) counts the distance between every pair of vertices twice, let x, y, z, and w in $V(T)$ be arranged as in Fig. 3.1. Note that z and x are not adjacent or identical.

Then the distance between z and x is counted once when $v_i = w$ and $v_j = x$, another time when $v_j = z$ and $v_i = y$. Notice that w (y) is the only neighbor of z (x) on the path between z and x.

Since T is a tree, there is a unique path between z and x. Then for any other

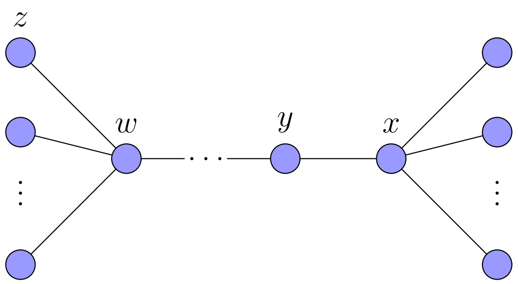


Figure 3.1: Distance between z and x is counted exactly twice.

combination of v_i and v_j, $(d_i - 1)(L_{i,j} + 1)$ does not count the distance between z and x. Hence, the distance between z and x is counted exactly twice.

Thus, (3.3) is exactly $2W(T)$ and

$$\begin{aligned} D'(T) &= \sum_{i=1}^{n}(\sum_{\substack{j=1\\ j\neq i}}^{n}(d_i - 1)(L_{i,j} + 1) + d_i) + 2W(T) - 2n|E(T)| + n(n-1) \\ &= 2W(T) + 2W(T) - 2n(n-1) + n(n-1) \\ &= 4W(T) - n(n-1) \end{aligned}$$

□

3.2 Cycles and Unicyclic graphs

In the following two sections, we extend this result to cycles and unicyclic graphs, and more generally to cacti. The ‘error term’ $e(G)$ enables us to present the general formula in a ‘neat’ manner that leads to its explicit form. Furthermore, we explore the bounds of the error term.

A cactus is a connected graph G where any two cycles share at most one vertex. Trees, cycles, and unicyclic graphs are all special cases of cacti. Let s be the number of cycles in G. Clearly, $s = 1$ in cycles and unicyclic graphs.

In the next section, we discuss the simple cases of cycles and unicyclic graphs.

3.2.1 Cycles

For cycles, the following is easily verified from the definition.

Theorem 3.2.1. *for a cycle C_n on n vertices we have,*

$$D'(C_n) = 4W(C_n) = 4W(C_n) - n(n+1) + e(C_n),$$

where $e(C_n) = n(n+1)$.

Proof. First notice that $|E(C_n)| = n$, from Lemma 3.1.1 we have

$$\begin{aligned} D'(G) &= \sum_{i=1}^{n}(\sum_{\substack{j=1 \\ j\neq i}}^{n}(d_i - 1)(L_{i,j} + 1) + d_i) + 2W(G) - n(n+1) \\ &= 4W(G) - n(n+1) + e(G) \qquad (3.4) \end{aligned}$$

For $e(G)$, we only show the case for even n, the other case is similar.

Suppose $G = C_{2m}$ is a cycle of even length $2m$. Label the vertices on the cycle in a clockwise fashion as in Fig. 3.2, with $1 \leq i < j \leq 2m$ and $v_0 := v_{2m}$, $v_{2m+1} := v_1$ respectively.

The maximum distance between two vertices on a cycle of length $2m$ is m. Note that there are two paths between two vertices in a cycle.

Suppose $2 \leq \ell - k \leq m - 2$, which implies $L_{k+1,\ell} + 2 = L_{k,\ell} + 1 = L_{k-1,\ell} \leq m - 1$. There is a unique shortest path between v_ℓ and any of v_{k-1}, v_k, or v_{k+1}, namely the

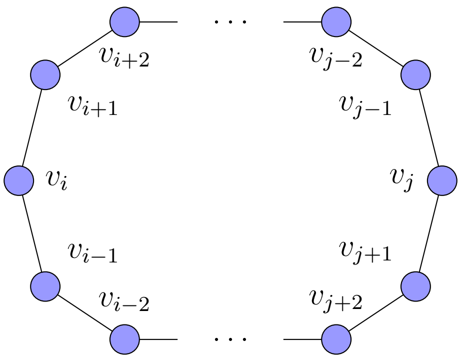


Figure 3.2: Cycle of length $2m$.

path going clockwise along the cycle. Since v_{k+1} ($v_{\ell-1}$) is the only neighbor of v_k (v_ℓ) on the aforementioned path, $(d_i - 1)(L_{i,j} + 1)$ counts the distance between v_k and v_ℓ if and only if $i = k + 1$ and $j = \ell$ or $j = k$ and $i = \ell - 1$. Hence, $L_{k,\ell}$ is counted exactly twice. The case for $\ell - k \geq m + 2$ is similar.

If $\ell - k = m - 1$ (Fig. 3.3), the distance between v_k and v_ℓ is counted once when $i = k+1$ and $j = \ell$ and again when $j = k$ and $i = \ell - 1$. Similarly for $\ell - k = m + 1$. Furthermore, when $\ell - k = m+1$, $(d_{k+1} - 1)(L_{k+1,\ell} + 1)$ incorrectly counts the distance between v_k and v_ℓ as being $L_{k,\ell+1} + 1 = m + 1$. Similarly, $(d_{k-1} - 1)(L_{k-1,\ell} + 1)$ counts the distance between v_k and v_ℓ incorrectly as being $m + 1$ when $\ell - k = m - 1$. Hence each case contributes $m(m + 1)$ as i goes from 1 to m, for a total contribution of $2m(m + 1)$ to $e(G)$.

Similarly, when $\ell - k = m$, there is an extra value of $2m^2$ contributed to $e(G)$.

Altogether, $e(G) = 2m(m + 1) + 2m^2 = n(n + 1)$. From (3.4) we have

$$D'(G) = 4W(G) - n(n + 1) + n(n + 1) = 4W(G).$$

Note that this can be easily verified from the definition of the degree distance. Since

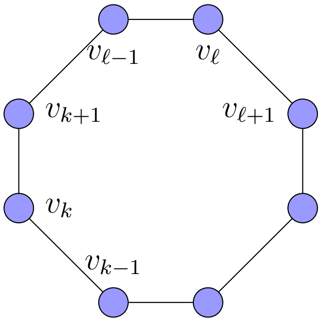


Figure 3.3: Two paths of equal length between v_{k-1} and v_ℓ on C_8.

all vertices on a cycle have degree two, we have

$$D'(G) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(d_i + d_j)L_{i,j} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(2+2)L_{i,j} = 2\sum_{i=1}^{n}\sum_{j=1}^{n}L_{i,j} = 4W(G).$$

□

3.2.2 Unicyclic graphs

For unicyclic graphs, we prove the following theorem.

Theorem 3.2.2. *For a unicyclic graph G on n vertices, we have*

$$D'(G) = 4W(G) - n(n+1) + e(G),$$

with $2(n-\lambda) + n(\lambda+1) \leq e(G) \leq (n-\lambda+1)^2 + n\lambda + \lambda - 1$ *where λ is the length of the unique cycle.*

Proof. The formula for $D'(G)$ follows immediately since $|E(G)| = |V(G)| = n$.

We first establish a formula for $e(G)$. Let $x_1, x_2, \ldots, x_\lambda$ be the vertices on the cycle labeled in a clockwise fashion as in Fig. 3.4, and let $X_1, X_2, \ldots, X_\lambda$ be the

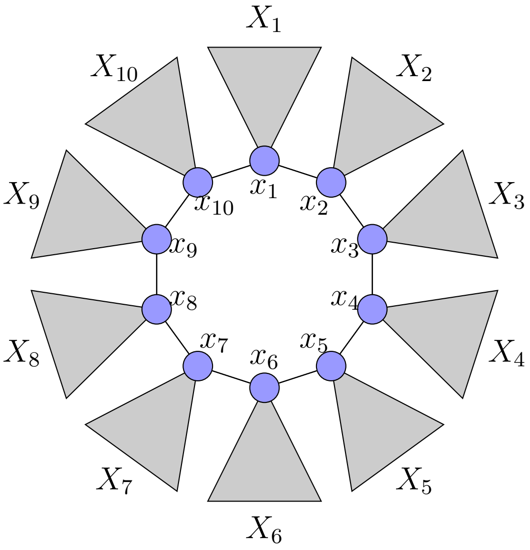


Figure 3.4: Unicyclic graph, $\lambda = 10$.

corresponding components after removing all edges on the cycle. Note that X_k is a tree for $k = 1, 2, \ldots, \lambda$.

First notice that the distances between any two vertices in X_k are counted exactly twice since X_k is a tree, for $k = 1, 2, \ldots, \lambda$. Suppose that $v_i \in X_k$ and $v_j \in X_\ell$, and $1 \leq k < \ell \leq \lambda$. For $v_i \neq x_k$, the shortest paths between v_j and all but one neighbors of v_i must contain the path from v_i to v_j. Then the distance between v_j and all neighbors of v_i not on the path between v_j and v_i is counted exactly twice.

Let λ be even (the case for odd λ is similar). Suppose further that $v_i = x_k$ and $v_j \in X_\ell$. If $\ell - k \leq \frac{\lambda}{2} - 2$ or $\ell - k \geq \frac{\lambda}{2} + 2$, then the shortest paths between v_j and all but one neighbors of x_k must contain the path from x_k to v_j. Then the distance between v_j and any vertex adjacent to x_k not on the path between v_j and x_k is counted exactly twice.

When $\ell - k = \frac{\lambda}{2} - 1$, the distance between v_j and any neighbor of x_k in $V(X_k)$

will be counted twice as discussed above. However, the distance between x_{k-1} and v_j is given by

$$1 + L(x_k, v_j) = 1 + L(x_k, x_\ell) + L(x_\ell, v_j) = \frac{\lambda}{2} + L(x_\ell, v_j)$$

where $L(v, u)$ is the distance between v and u. Note that this distance is also counted for the case $\ell - k' = \frac{\lambda}{2} + 1$. Hence, it is overcounted once for every $v_j \in V(X_\ell)$. Thus, the total contribution to $e(G)$ as ℓ ranges from 1 to λ is

$$\sum_{\ell=1}^{\lambda} \sum_{v \in V(X_\ell)} (\frac{\lambda}{2} + L(x_\ell, v)) = n(\frac{\lambda}{2}) + \sum_{\ell=1}^{\lambda} \sum_{v \in V(X_\ell)} L(x_\ell, v).$$

When $\ell - k = \frac{\lambda}{2}$, the distance between x_{k-1} or x_{k+1} and $v_j \in V(X_\ell)$ is miscounted once as $\frac{\lambda}{2} + 1 + L(x_\ell, v)$. The contribution to $e(G)$ is given by

$$\sum_{\ell=1}^{\lambda} \sum_{v \in V(X_\ell)} (\frac{\lambda}{2} + 1 + L(x_\ell, v)) = n(\frac{\lambda}{2} + 1) + \sum_{\ell=1}^{\lambda} \sum_{v \in V(X_\ell)} L(x_\ell, v).$$

Hence,

$$e(G) = 2\sum_{\ell=1}^{\lambda} \sum_{v \in V(X_\ell)} L(x_\ell, v) + n(\lambda + 1) = 2\sum_{\ell=1}^{\lambda} D_\ell + n(\lambda + 1).$$

Here $D_\ell = \sum\limits_{v \in V(X_\ell)} L(x_\ell, v)$ is often referred to as the *distance function* of x_ℓ in X_ℓ.

Next we analyze the value $e(G)$. It is known that, with given number of vertices, D_ℓ is minimized by the center of a star and maximized by one end of a path. Hence, we have

$$|V(X_\ell)| - 1 \leq D_\ell \leq \frac{1}{2}|V(X_\ell)|(|V(X_\ell)| - 1).$$

One can easily see from (3.2.2) that the value of $e(G)$ is only changed when the values of D_ℓ change, for $\ell = 1$ to λ, since both the number of vertices and the cycle length are constant. Then by minimizing D_ℓ for all ℓ we achieve a lower bound for $e(G)$ by making all vertices which do not lie on a cycle adjacent to x_ℓ for some $1 \leq \ell \leq \lambda$.

Similarly, one can show $e(G)$ is maximized when one component contains a pendant path and the other components are empty. Thus, we have

$$2\sum_{\ell=1}^{\lambda}(|V(X_\ell)|-1)+n(\lambda+1)\le e(G)\le\sum_{\ell=1}^{\lambda}\Big(|V(X_\ell)|(|V(X_l)|-1)\Big)+n(\lambda+1),$$

implying that

$$2(n-\lambda)+n(\lambda+1)\le e(G)\le(n-\lambda+1)^2+n\lambda+\lambda-1.$$

□

3.3 General cacti

Let G be a cactus with s cycles and r edges not on any cycle. Label the cycles c_α for $\alpha=1,2,\ldots,s$, let λ_α be the length of c_α and x_ℓ^α be a vertex on c_α with component X_ℓ^α for $\ell=1,2,\ldots,\lambda_\alpha$. Define the distance function of x_ℓ^α by $D_\ell^\alpha=\sum\limits_{v\in X_\ell^\alpha}L(x_\ell^\alpha,v)$.

Since $|E(G)|=n+s-1$, we immediately have

$$D'(G)=4W(G)-n(n+2s-1)+e(G).$$

Now we start by providing a formula for $e(G)$ for general cacti, then we develop the bounds of $e(G)$.

3.3.1 On $e(G)$

As was the case for the unicyclic graph, for every cycle in G we have a contribution of $2\sum\limits_{\ell=1}^{\lambda_\alpha}\sum\limits_{v\in V(X_\ell^\alpha)}L(x_\ell^\alpha,v_j)+n(\lambda_\alpha+1)$ to $e(G)$. An explicit formula for $e(G)$ is given by

$$
\begin{aligned}
e(G) &= \sum_{\alpha=1}^{s}\Big(2\sum_{\ell=1}^{\lambda_\alpha} D_\ell^\alpha + n(\lambda_\alpha+1)\Big)\\
&= 2\sum_{\alpha=1}^{s}\sum_{\ell=1}^{\lambda_\alpha} D_\ell^\alpha + n\sum_{\alpha=1}^{s}(\lambda_\alpha+1)\\
&= 2\sum_{\alpha=1}^{s}\sum_{\ell=1}^{\lambda_\alpha} D_\ell^\alpha + n(n-r+2s-1). \qquad (3.5)
\end{aligned}
$$

3.3.2 Bounds and extremal cacti

We claim that, with given n, s, r and cycle lengths, the *star-shaped cactus* (a cactus that has only one cut vertex as its *center*, see Fig. 3.5) minimizes $e(G)$.

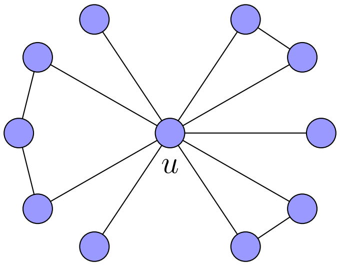


Figure 3.5: A star-shaped cactus with center u.

Proposition 3.3.1. *Given a cactus G with order n, s cycles, r edges and any vertex $w \in V(G)$, there exists a star-shaped cactus with the same parameters and cycle lengths with center u, such that $\sum_{v\in V(S)} L(v,u) \le \sum_{v\in V(G)} L(v,w)$.*

Alternatively, we could state the proposition above in the following way: For a cycle of length λ_α in a cactus G with the vertices on the cycle labeled x_1^α to x_λ^α as before, D_ℓ^α is minimized when X_ℓ^α is a star-shaped cactus. We skip the proof here, but one can easily see the idea from Fig. 3.6 and Fig. 3.7. Notice that either operation from G to H will reduce $\sum_{v\in V(G)} L(v,w)$ unless we have a star-shaped cactus with center $u=w$.

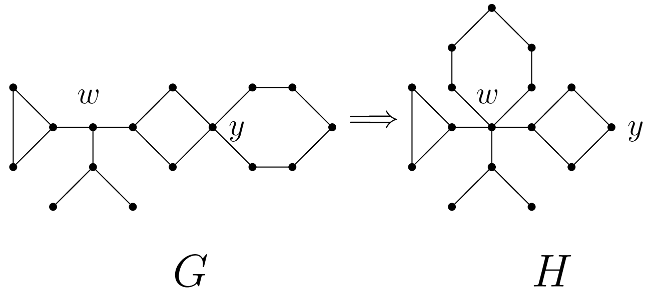


Figure 3.6: Moving X_y to w.

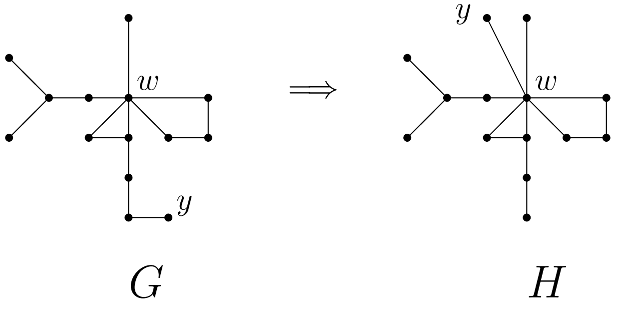


Figure 3.7: Appending y to w.

From the proposition, it follows that when G is a star-shaped cactus, $e(G)$ is minimized. The proof is omitted. Then for every cycle, $D_\ell^\alpha = 0$ for all but one x_ℓ. Consequently,

$$\begin{aligned}
e(G) &= 2\sum_{\alpha=1}^{s}\sum_{\ell=1}^{\lambda_\alpha} D_\ell^\alpha + n(n-r+2s-1) \\
&\geq 2\sum_{\alpha=1}^{s}\left(r + \left\lfloor \frac{\lambda_\alpha}{2} \right\rfloor \left\lfloor \frac{\lambda_\alpha+1}{2} \right\rfloor\right) + n(n-r+2s-1) \\
&= 2\sum_{\alpha=1}^{s}\left(\left\lfloor \frac{\lambda_\alpha}{2} \right\rfloor \left\lfloor \frac{\lambda_\alpha+1}{2} \right\rfloor\right) + 2rs + n(n-r+2s-1).
\end{aligned}$$

Remark: Of course, if one does not specify the cycle lengths, then certain bounds, like the unicyclic case, can be achieved. Also, it is interesting to see that $e(G)$ is minimized by a star-shaped cactus, which minimizes a number of graphical indices including $W(G)$ ([21]) among cacti with given parameters.

CHAPTER 4

ALGORITHMS ON SWITCHING COMPONENTS IN TREES

A *tree* $T = (V, E)$ is a connected, acyclic graph having vertex set $V(T)$ and edge set $E(T)$. Recall that for $u \in V(T)$, $d_T(u) = degree_T(u)$ denotes the degree of vertex u. For two vertices $u, v \in V(T)$, we denote the unique path which starts at u and ends at v as $P_T(u, v)$. The *distance*, $d_T(u, v)$, between u and v is then the number of edges in $P_T(u, v)$. If we let

$$g_T(u) = \sum_{v \in V(T)} d_T(u, v), \tag{4.1}$$

with $d_T(u, u) = 0$, then the *Wiener Index*, $\sigma(T)$, is given by

$$\sigma(T) = \frac{1}{2} \sum_{u \in V(T)} g_T(u). \tag{4.2}$$

We denote a tree T *rooted* at $r \in V(T)$ as (T, r). Let $h_T(u) = d_T(r, u)$ be the *height* of u in (T, r). If $h_T(u) < h_T(v)$ then we say u is an *ancestor* of v or v is a *descendant* of u. If $h_T(u) = h_T(v) - 1$ for two vertices u and v, then we say u is the *parent* of v and v is the *child* of u. When two vertices share the same parent we call them *siblings*.

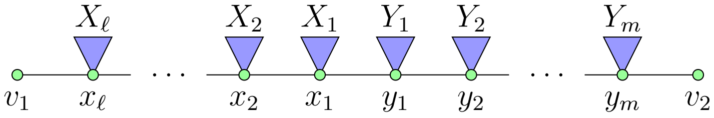


Figure 4.1: A path between two leaf vertices, v_1 and v_2.

We say a vertex, v, is a leaf vertex if $d(v) = 1$. Suppose that $v_1 x_1 x_2 x_3 \ldots x_n v_2$ is a path in T from a leaf vertex, v_1, to another leaf vertex, v_2. After the removal of all edges on the path there are still connected components left, each containing one of $x_1, x_2, \ldots, x_n$. Label these components as $X_1, X_2, \ldots, X_n$, respectively. Note that the component of a leaf vertex contains only the leaf vertex. Figure 4.1 shows the labelling of the vertices and components. Also, let $X_{\leq i}$ be the subtree consisting of all components X_j, all vertices x_j, and all edges $x_j x_{j-1}$, where $j \leq i$ (as depicted in Figure 4.2); and let $X_{\geq i}$ be the subtree consisting of all components X_j, all vertices x_j, and all edges $x_j x_{j+1}$, where $j \geq i$.

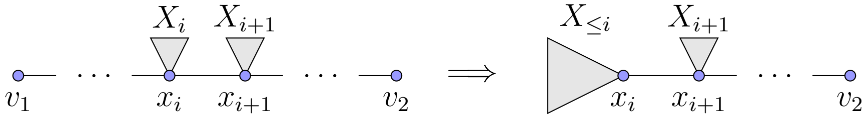


Figure 4.2: $X_{\leq i}$ is everything to the left of $x_i x_{i+1} \in E(G)$.

Suppose G is a tree. The number of subtrees of G is denoted by $f(G)$, and is equal to the number of possible connected structures that can be formed from the vertices and edges of G. Let $f^G_{x_i}(X_i)$ be the number of subtrees in the component X_i which contain the vertex x_i for $i = 1, 2, \ldots, n$. Similarly, $f^G_{x_{i-1}}(X_{\leq i-1})$ and $f^G_{x_{i+1}}(X_{\geq i+1})$ are the number of subtrees in $X_{\leq i-1}$ containing x_{i-1} and the number of subtrees in $X_{\geq i+1}$ containing x_{i+1}.

The degree sequence of a graph is the multiset containing the degrees of all non-leaf vertices in descending order. The following result was proven in [29]:

Theorem 4.0.2. *Given the degree sequence and the number of vertices, the greedy tree minimizes the Wiener Index.*

Details of a greedy tree can be found in Chapter 5. We use an approach similar to the one used in [29] to develop an algorithm to find the maximal tree wherein every step in the algorithm leads to an increase in the number of subtrees. In Chapter 5 we use this algorithm to find the maximal tree and second maximal tree.

Definition 4.0.3. *Suppose we have a path in a tree G between two leaf vertices, $v_1x_1x_2\ldots x_nv_2$, as in Figure 4.1. A 'component-switch', $S^G_{v_1,v_2}(X_i, X_j)$, of two components X_i and X_j is done by removing all edges with vertices in X_i adjacent to x_i and drawing an edge from each of those vertices to x_j, and removing all edges with vertices in X_j adjacent to x_j and drawing an edge from each of those vertices to x_i. Thus, X_j is now the component of x_i and X_i is the component of x_j.*

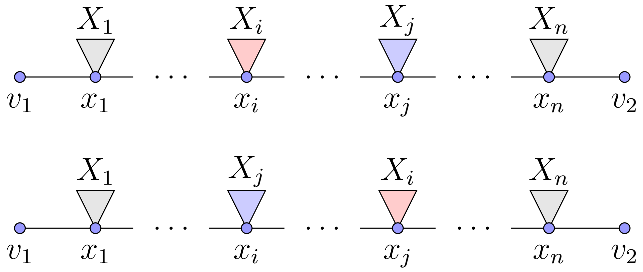


Figure 4.3: Example of a component switch.

Definition 4.0.4. *Suppose we have a path in a tree G between two leaf vertices, $v_1x_1x_2\ldots x_nv_2$. A 'tail-switch', $S^G_{v_1,v_2}(X_{\leq i}, X_{\geq j})$, of two tails $X_{\leq i}$ and $X_{\geq j}$ (where $i < j$) is done by removing all edges with vertices in $X_{\leq i}$ adjacent to x_i and drawing an edge from each of those vertices to x_j, and removing all edges with vertices in $X_{\geq j}$ adjacent to x_j and drawing an edge from each of those vertices to x_i.*

Definition 4.0.5. *Suppose that $p = d_G(x_k) < d_G(y_k) = q$ in Figure 4.5. Without loss of generality, let the $Y_{k,i}$ components for $i = 1$ to q be ordered from smallest to largest by the number of subtrees. In this situation we have $Y^G_k = Y'_k \bigcup Y''_k$. Then, letting H be the tree obtained by moving Y''_k to x_k, we will have $d_H(x_k) = q$ and $d_H(y_k) = p$. We will refer to the operation described above as a 'degree-switch', denoted by $H = S^G_{v_1,v_2}(\emptyset_{x_k}, Y''_k) := R^G_{v_1,v_2}(x_k, y_k)$.*

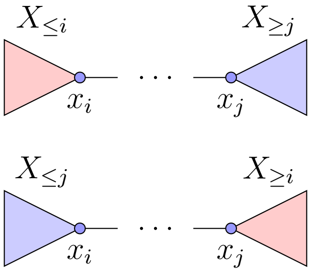


Figure 4.4: Example of a tail switch.

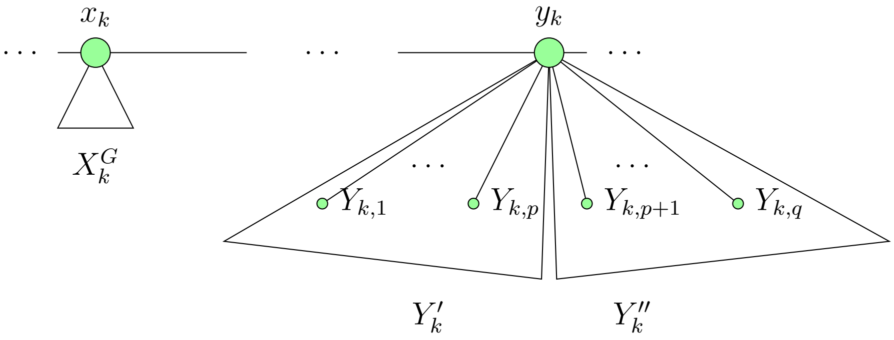


Figure 4.5: When $p = d(x_k) < d(y_k) = q$.

Consequently any subtree in X_i which previously contained x_i will now contain x_j, and any subtree in X_j which previously contained x_j will now contain x_i. Note that the positions of x_i and x_j on the path do not change.

Definition 4.0.6. *Let G be a tree. Let S_1 be a switching on a path in the tree G, and denote the resulting tree as G_1. Now perform a second switching, S_2, with respect to a path (possibly a different path than for S_1) in G_1, and call the resulting graph G_2. If we have m such switchings, then we say $S_1 S_2 \ldots S_m$ is a 'chain of switchings' of length m from G to G_m.*

Remark 4.0.7. *A chain of switchings may not be unique, that is there may exist more than one chain of switchings to get from a tree G to a tree G_m.*

4.1 Phase I of the switching algorithm

For convenience we introduce the following notations:

$$f^G_{x_k}(X_k) = C^G_k \qquad f^G_{x_k}(X_{\leq k}) = C^G_{\leq k} \qquad f^G_{x_k}(X_{\geq k}) = C^G_{\geq k}$$

Lemma 4.1.1. *Suppose we have $C^G_{\leq k-1} < C^G_{\geq k+2}$. If $C^G_k > C^G_{k+1}$ then switching X_k and X_{k+1} will result in an increase in the number of subtrees.*

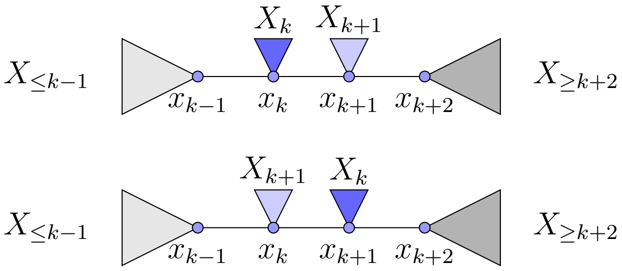


Figure 4.6: G, before switching, and H, after switching.

Proof. Let G denote the tree prior to switching and H the tree after switching, as in Figure 4.6. When switching two components, X_k and X_{k+1}, the only subtrees that will affect the change in the total number of subtrees are those which have at least one branch in X_k or X_{k+1} and at least one branch in $X_{\leq k-1}$ or $X_{\geq k+2}$ (we can form a bijection between all other subtrees in G with all other subtrees in H). All such trees contain either x_k or x_{k+1}. Then the change in the number of subtrees is the number of subtrees which contain x_k or x_{k+1} in H minus the number of subtrees which contain x_k or x_{k+1} in G.

In G, the number of subtrees containing vertex x_k can be counted as follows:

1. consider the number of subtrees in the tail $X_{\leq k}$ which contain x_k, given by $C_k^G(1 + C_{\leq k-1}^G)$ where 1 counts the edge $x_k x_{k-1}$ and $C_{\leq k-1}^G$ is the number of subtrees in $X_{\leq k-1}$ containing x_{k-1};

2. consider the number of subtrees containing the edge $x_k x_{k+1}$ and the number of subtrees in the tail $X_{\geq k+1}$ containing x_{k+1}, given by $1 + C_{\geq k+1} = 1 + C_{k+1}(1 + C_{\geq k+2})$;

3. the number of subtrees in G containing x_k is given by the product of the two terms,
$$f_{x_k}(G) = C_k^G(1 + C_{\leq k-1}^G)(1 + C_{k+1}(1 + C_{\geq k+2})).$$

The number of subtrees containing x_{k+1} is found similarly, and is given by

$$f_{x_{k+1}}(G) = C_{k+1}^G(1 + C_{\geq k+2}^G)(1 + C_k^G(1 + C_{\leq k-1}^G)).$$

In H, the number of subtrees containing x_k is counted identically, but reversing the roles of C_k and C_{k+1} since they have been switched in G. Hence,

$$f_{x_k}(H) = C_{k+1}^G(1 + C_{\leq k-1}^G)(1 + C_k(1 + C_{\geq k+2})),$$

and

$$f_{x_{k+1}}(H) = C_k^G(1 + C_{\geq k+2}^G)(1 + C_{k+1}^G(1 + C_{\leq k-1}^G)).$$

Consequently, we have

$$\begin{aligned}
&f(H) - f(G) \\
&= C^G_{k+1}(1 + C^G_{\leq k-1})(1 + C^G_k(1 + C^G_{\geq k+2})) \\
&\quad + C^G_k(1 + C^G_{\geq k+2})(1 + C^G_{k+1}(1 + C^G_{\leq k-1})) \\
&\quad - C^G_k(1 + C^G_{\leq k-1})(1 + C^G_{k+1}(1 + C^G_{\geq k+2})) \\
&\quad - C^G_{k+1}(1 + C^G_{\geq k+2})(1 + C^G_k(1 + C^G_{\leq k-1})) \\
&= (C^G_k - C^G_{k+1})((1 + C^G_{\geq k+2}) - (1 + C^G_{\leq k-1})) \\
&= (C^G_k - C^G_{k+1})(C^G_{\geq k+2} - C^G_{\leq k-1}) \\
&> 0
\end{aligned}$$

Notice that the number of subtrees containing both x_k and x_{k+1} does not change from G to H. In future proofs we ignore such subtrees, considering only the subtrees containing exactly 1 of the vertices.

□

The following algorithm is based on the bubblesort algorithm, which sorts an array of numbers from smallest to largest or vice versa. Here we apply it to the switching of components on a path between two leaf vertices in G. Let $C_{\leq 0} = C_{\geq n} = 1$ refer to the number of nonempty subtrees in the components of our two leaf vertices, v_1 and v_2, in the following algorithm.

Algorithm 4.1.2. ***(Phase I Switching Algorithm)***. *Suppose G is a tree. Let $v_1, v_2 \in V(G)$ be leaf vertices. Let* ***Label 1*** *be the labelling of the path from v_1 to v_2 denoted by $v_1x_1x_2\ldots x_nv_2$, and let* ***Label 2*** *be a re-labelling of the path from v_2 to v_1 given by $v_2x_1x_2\ldots x_nv_1$. Then x_1 is adjacent to v_1 and x_n is adjacent to v_2 in* ***Label***

***1** and x_1 is adjacent to v_2 and x_n is adjacent to v_1 in **Label 2**. $H = P_1^G(v_1, v_2)$ is given below.*

```
$H = (\emptyset, \emptyset)$ {Empty graph}
$m = 0$
$G_0 = G$
while $H \neq G_m$ do
  $k = 1$
  $H = G_m$ {Terminate program if no change in $G_m$}
  Label $x_i$'s according to Label 1.
  while $C^{G_m}_{\leq k-1} < C^{G_m}_{\geq k+2}$ do
    if $C^{G_m}_{k} > C^{G_m}_{k+1}$ {Conditions of Lemma 4.6} then
      $m = m + 1$ {Update only if an increase will occur}
      $G_m = S^{G_{m-1}}_{v_1,v_2}(X_k, X_{k+1})$ {Increases # of subtrees}
      $k = k + 1$
    else
      $k = k + 1$
    end if
  end while
  $k = 1$
  Relabel $x_i$'s according to Label 2.  {Repeat for Label 2}
  while $C^{G_m}_{\leq k-1} < C^{G_m}_{\geq k+2}$ do
    if $C^{G_m}_{k} > C^{G_m}_{k+1}$ then
      $m = m + 1$
      $G_m = S^{G_{m-1}}_{v_2,v_1}(X_k, X_{k+1})$
      $k = k + 1$
```

```
    else
      $k = k + 1$
    end if
  end while
end while
```

In the algorithm, G_m updates only if there is an increase in the number of subtrees from G_{m-1} to G_m. There are a finite number of possible arrangements of the components on the path, and if $f(G_m) = f(G_{m-k})$ for $k > 0$, then $f(G_{m-k}) < f(G_m)$ leads to a contradiction. Thus, the algorithm must terminate at some point.

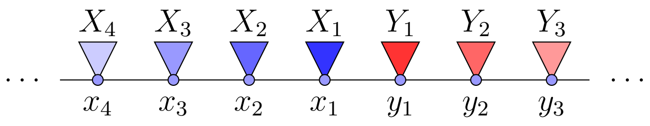


Figure 4.7: Re-labelling of path after Phase I Switching Algorithm.

In the graph G described by Figure 4.8, let

$$f^G_{x_k}(X_k) = C^G_k,$$

$$f^G_{y_k}(Y_k) = D^G_k,$$

$$f^G_{u_k}(U_k) = E_k,$$

$$f^G_{x_{k-1}}(X_{\leq k-1} \cup U_{\geq 1} \cup Y_{\geq 1}) = C^G_{\leq k-1}(1 + E^G_{\geq 1}(1 + D^G_{\geq 1})),$$

$$f^G_{y_{k-1}}(Y_{\leq k-1} \cup U_{\leq l} \cup X_{\geq 1}) = D^G_{\leq k-1}(1 + E^G_{\leq l}(1 + C^G_{\geq 1})),$$

$$f^G_{x_{k+1}}(X_{\geq k+1}) = C^G_{\geq k+1},$$

$$f^G_{y_{k+1}}(Y_{\geq k+1}) = D^G_{\geq k+1}.$$

Corollary 4.1.3. *Let G be a tree with a path from leaf vertex v_1 to leaf vertex v_2 that has been sorted by the Phase I Switching Algorithm. Let x_1 be the vertex on the path with the largest component, and label the rest of the vertices as in Figure 4.7, (without loss of generality, let y_1 and x_2 be labelled so that $D_1 \geq C_2$). Then*

$$C_1^G \geq C_2^G \geq \dots \qquad \text{and} \qquad D_1^G \geq D_2^G \geq \dots.$$

Proof. The proof is by contradiction. Suppose that the path cannot be labelled in such a way. We know that the Phase I Switching Algorithm partially sorts both ends of the path so that, after re-labelling according to Figure 4.8, $C_1^G \geq C_2^G \geq \dots$ and $D_1^G \geq D_2^G \geq \dots$, with $E_1^G \leq C_1^G$ and $E_\ell^G \leq D_1^G$ where $l \geq 1$ (otherwise, relabel so that U_1 is X_1 or U_ℓ is Y_1).

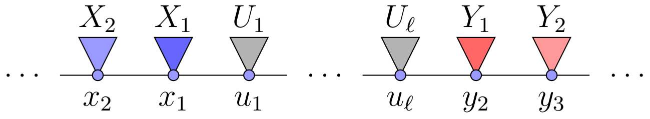


Figure 4.8: Tree leading to a contradiction.

Suppose $l \geq 2$. For the algorithm not to have switched X_1 and U_1, it must be true that $C_{\geq 2}^G \geq E_{\geq 2}^G(1 + D_{\geq 1})$. Similarly, for Y_1 and U_ℓ we have $D_{\geq 2}^G \geq E_{\leq \ell-1}^G(1 + C_{\geq 1})$. Note that $E_{\leq \ell-1}^G(1 + C_{\geq 1}) > C_{\geq 2}^G$ and $E_{\geq 2}^G(1 + D_{\geq 1}) > D_{\geq 2}^G$. Then we have

$$D_{\geq 2}^G \geq E_{\leq \ell-1}^G(1 + C_{\geq 1}) > C_{\geq 2}^G \text{ and}$$
$$C_{\geq 2}^G \geq E_{\geq 2}^G(1 + D_{\geq 1}) > D_{\geq 2}^G,$$

which is a contradiction. For the case when $\ell = 1$, replace $E_{\leq \ell-1}^G(1 + C_{\geq 1})$ with $C_{\geq 1}^G$ and $E_{\geq 2}^G(1 + D_{\geq 1})$ with $D_{\geq 1}^G$, and the argument still holds.

□

4.2 Phase II of the switching algorithm

Although it does not involve the switching of components, we introduce Lemma 4.2.1 here to simplify the proofs of the other lemmas in this section. For the lemmas that follow, suppose that our path is labelled as in Figure 4.7.

Lemma 4.2.1. *Suppose we have* $C_1^G \geq D_1^G \geq C_2^G \geq \ldots \geq C_{k-1}^G \geq D_{k-1}^G$. *Let*

$$\begin{aligned} C = \ & C_{k-1}^G + C_{k-1}^G C_{k-2}^G + \ldots + C_{k-1}^G \ldots C_1^G + C_{k-1}^G \ldots C_1^G D_1^G \\ & + C_{k-1}^G \ldots C_1^G D_1^G \ldots D_{k-1}^G, \end{aligned}$$

and

$$\begin{aligned} D = \ & D_{k-1}^G + D_{k-1}^G D_{k-2}^G + \ldots + D_{k-1}^G \ldots D_1^G + D_{k-1}^G \ldots D_1^G C_1^G \\ & + D_{k-1}^G \ldots D_1^G C_1^G \ldots C_{k-1}^G. \end{aligned}$$

Then $C \geq D$. *Similarly, suppose* $C_1^G \geq D_1^G \geq C_2^G \geq \ldots \geq C_{k-1}^G \geq D_{k-1}^G \geq C_k^G$. *Let*

$$\begin{aligned} C' = \ & C_k^G + C_k^G C_{k-1}^G + \ldots + C_k^G \ldots C_1^G + C_k^G \ldots C_1^G D_1^G \\ & + C_k^G \ldots C_1^G D_1^G \ldots D_{k-1}^G, \end{aligned}$$

and

$$\begin{aligned} D' = \ & D_{k-1}^G + D_{k-1}^G D_{k-2}^G + \ldots + D_{k-1}^G \ldots D_1^G + D_{k-1}^G \ldots D_1^G C_1^G \\ & + D_{k-1} \ldots D_1^G C_1^G \ldots C_k^G. \end{aligned}$$

Then $D' \geq C'$.

Proof. Suppose we have $C_1^G \geq D_1^G \geq C_2^G \geq \ldots \geq C_{k-1}^G \geq D_{k-1}^G$. For $j = 1, 2, \ldots k-1$, clearly we have $C_{k-1} \ldots C_{k-j} \geq D_{k-1} \ldots D_{k-j}$. Let $j \leq k-1$.

$$\begin{aligned} & C_{k-1}^G \ldots C_1^G D_1^G \ldots D_j^G - D_{k-1}^G \ldots D_1^G C_1^G \ldots C_j^G \\ = \ & C_1^G \ldots C_j^G D_1^G \ldots D_j^G (C_{j+1}^G \ldots C_{k-1}^G - D_{j+1}^G \ldots D_{k-1}^G) \\ \geq \ & 0. \end{aligned}$$

This implies $C \geq D$. The other case, $D' \geq C'$, can be shown similarly.

$\square$

Lemma 4.2.2. *Suppose we have* $C_1^G \geq D_1^G \geq C_2^G \geq \ldots \geq C_{k-1}^G \geq D_{k-1}^G$ *for some* k. *If* $D_{\geq k}^G > C_{\geq k}^G$ *then switching* $D_{\geq k}^G$ *and* $C_{\geq k}^G$ *will result in in an increase in the number of subtrees. Likewise, suppose we have* $C_1^G \geq D_1^G \geq C_2^G \geq \ldots \geq D_{k-1}^G \geq C_k^G$. *If* $C_{\geq k+1}^G > D_{\geq k}^G$, *then switching* $C_{\geq k+1}^G$ *and* $D_{\geq k}^G$ *will result in an increase in the number of subtrees.*

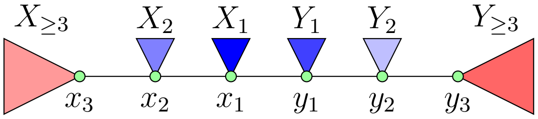


Figure 4.9: Example depicting the conditions of Lemma 4.2.2.

Proof. Suppose we have the first case, with G being the tree prior to switching and H the tree after switching. Let $G \sim X_{\geq k}$ be the subtree of G that excludes all subtrees with branches in $X_{\geq k}$. The number of subtrees in G that contain x_k and do not contain y_k is found as follows:

1. the number of subtrees in the tail $X_{\geq k}$ is by definition $C_{\geq k}$;

2. the number of subtrees in $\{x_k x_k - 1\} \bigcup X_{\leq k-1} \bigcup Y_{\geq 1} \sim Y_{\geq k}$ is given by

$$\begin{aligned} & f_{x_k}(\{x_k x_k - 1\} \bigcup X_{\leq k-1} \bigcup Y_{\geq 1} \sim Y_{\geq k}) \\ = \;& 1 + C_{\leq k-1}^G(1 + D_1(1 + D_2(\ldots(1 + D_{k-1})))) \\ = \;& 1 + C; \end{aligned}$$

3. finally, multiply the two terms together to obtain,

$$f_{x_k}(G \sim Y_{\geq k}) = C_{\geq k}(1 + C).$$

Similarly, one can find

$$f_{y_k}(G \sim X_{\geq k}) = D_{\geq k}(1 + D).$$

In H, simply reverse the roles of $C_{\geq k}$ and $D_{\geq k}$. Hence,

$$f_{x_k}(H \sim Y_{\geq k}) = D_{\geq k}(1 + C),$$

$$f_{y_k}(H \sim X_{\geq k}) = C_{\geq k}(1 + D).$$

We now compute $f(H) - f(G)$,

$$\begin{aligned} & f(H) - f(G) \\ = \;& D^G_{\geq k}(1 + C) + C^G_{\geq k}(1 + D) - C^G_{\geq k}(1 + C) - D^G_{\geq k}(1 + D) \\ = \;& (D^G_{\geq k} - C^G_{\geq k})(C - D) + 0 \\ \geq \;& 0. \end{aligned}$$

Notice $f(H) - f(G) > 0$ if $C_j > D_j$ for some $1 \leq j \leq k - 1$. The other case can be shown similarly.

□

Lemma 4.2.3. *Suppose we have* $C^G_1 \geq D^G_1 \geq C^G_2 \geq D^G_2 \geq \ldots \geq D^G_{k-1} \geq C^G_k$ *and* $C^G_{\geq k+1} \geq D^G_{\geq k+1}$. *If* $D^G_k > C^G_k$ *then switching* D^G_k *and* C^G_k *will result in an increase in the number of subtrees. Likewise, suppose we have* $C^G_1 \geq D^G_1 \geq C^G_2 \geq D^G_2 \geq \ldots \geq C^G_k \geq D^G_k$ *and* $D^G_{\geq k+1} \geq C^G_{\geq k+2}$ *for some* k. *If* $C^G_{k+1} > D^G_k$ *then switching* C^G_k *and* D^G_k *will result in an increase in the number of subtrees.*

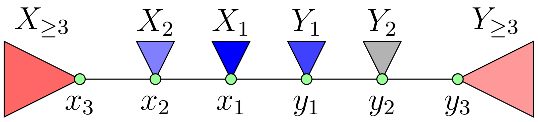


Figure 4.10: Example depicting conditions of Lemma 4.2.3.

Proof. Suppose we have the first case. Let G be the graph before switching X_k and Y_k and we let H be the tree after switching. We now consider the difference in the number of subtrees. The following are obtained as before (excluding the subtrees containing both x_k and y_k),

$$\begin{aligned}
f_{x_k}(G \sim Y_{\geq k}) &= C_k^G(1 + C_{\geq k+1}^G)(1 + C), \\
f_{y_k}(G \sim X_{\geq k}) &= D_k^G(1 + D_{\geq k+1}^G)(1 + D), \\
f_{x_k}(H \sim Y_{\geq k}) &= D_k^G(1 + C_{\geq k+1}^G)(1 + C), \\
f_{y_k}(H \sim X_{\geq k}) &= C_k^G(1 + D_{\geq k+1}^G)(1 + D).
\end{aligned}$$

Hence,

$$\begin{aligned}
&f(H) - f(G) \\
&= D_k^G(1 + C_{\geq k+1}^G)(1 + C) + C_k^G(1 + D_{\geq k+1}^G)(1 + D) \\
&\quad - C_k^G(1 + C_{\geq k+1}^G)(1 + C) - D_k^G(1 + D_{\geq k+1}^G)(1 + D) \\
&= (D_k^G - C_k^G)[(1 + C_{\geq k+1})(1 + C) - (1 + D_{\geq k+1}^G)(1 + D)] \\
&\geq 0.
\end{aligned}$$

Notice $f(H) - f(G) > 0$ if $C_j > D_j$ for some $1 \leq j \leq k - 1$. The other case can be shown similarly.

□

We now apply Lemmas 4.2.2 and 4.2.3 to formulate a new algorithm, combining component-switches and tail-switches to sort a path. The algorithm will terminate if the number of non-empty subtrees in both tails are equal to 1, indicating that the tails contain only the endpoints of the path.

Algorithm 4.2.4. ***(Phase II Switching Algorithm)***. *Suppose G is a tree. Let $v_1, v_2 \in V(G)$ be leaf vertices. Choose x_1 to be the vertex whose component has the largest number of subtrees. If the two vertices adjacent to x_1 have an equal number of subtrees in their components, without loss of generality let y_1 be the vertex closest to v_1 and x_2 be the other vertex, otherwise let y_1 be the vertex with the larger number of subtrees in its component. Label the other vertices according to Figure 4.7. Then $H = P_2^G(v_1, v_2)$ is given below.*

```
$H = (\emptyset, \emptyset)$
$m = 0$
$G_m = G$
$k = 1$
while $C^{G_m}_{\geq k+1} \neq 1$ OR $D^{G_m}_{\geq k+1} \neq 1$ $\{$Terminate program if leaf vertex$\}$ do
  if $C^{G_m}_{k} < D^{G_m}_{k}$ $\{$Conditions of Lemma 4.2.3$\}$ then
    if $C^{G_m}_{\geq k+1} < D^{G_m}_{\geq k+1}$ $\{$Conditions of Lemma 4.2.2$\}$ then
      $m = m + 1$
      $G_m = S^{G_{m-1}}_{v_1,v_2}(X_{\geq k}, Y_{\geq k})$
    else
      $m = m + 1$
      $G_m = S^{G_{m-1}}_{v_1,v_2}(X_k, Y_k)$
    end if
  end if
  if $D^{G_m}_{k} < C^{G_m}_{k+1}$ $\{$Repeat for $y_k$ and $x_{k+1}\}$ then
```

```
    if $D^{G_m}_{\geq k+1} < C^{G_m}_{\geq k+2}$ then
      $m = m + 1$
      $G_m = S^{G_{m-1}}_{v_1,v_2}(Y_{\geq k}, X_{\geq k+1})$
    else
      $m = m + 1$
      $G_m = S^{G_{m-1}}_{v_1,v_2}(Y_k, D_{k+1})$
    end if
  end if
  $k = k + 1$
end while
$H = G_m$
```

After the Phase II Switching Algorithm has finished, the path from v_1 to v_2 is labelled in such a way that

$$C^G_1 \geq D^G_1 \geq C^G_2 \geq \ldots \geq C^G_n \geq D^G_n$$

$$\text{and } C^G_{\geq 1} \geq D^G_{\geq 1} \geq C^G_{\geq 2} \geq \ldots \geq C^G_{\geq n} \geq D^G_{\geq n}$$

for a path of odd length, $2n - 1$, and

$$C^G_1 \geq D^G_1 \geq C^G_2 \geq \ldots \geq D^G_n \geq C^G_{n+1}$$

$$\text{and } C^G_{\geq 1} \geq D^G_{\geq 1} \geq C^G_{\geq 2} \geq \ldots \geq D^G_{\geq n} \geq C^G_{\geq n+1}$$

for a path of even length, $2n$.

4.3 Phase III of the switching algorithm

The final phase of the switching algorithm uses degree-switches to sort the degrees of the vertices on the path..

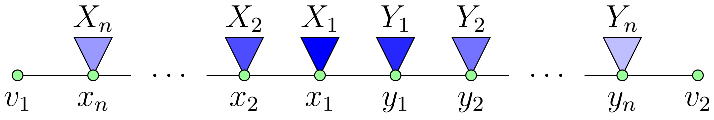


Figure 4.11: Path of length $2n-1$ after Phase II switching algorithm.

Lemma 4.3.1. *Let G be a tree, with a path from leaf vertex v_1 to leaf vertex v_2 that has been sorted by the Phase I and Phase II Switching Algorithms. For $k = 1, 2, \ldots, n$, if $d(x_k) < d(y_k)$ then performing an degree-switch, as described by Definition 4.0.5, will result in an increase in the number of subtrees. Similarly, if $d(y_k) < d(x_{k+1})$ then an degree-switch with X''_{k+1} will result in an increase in the number of subtrees, where X''_{k+1} is defined in the same way as Y''_k (See Figure 4.5).*

Proof. We show only the case where $d(x_k) < d(y_k)$, the other case follows similarly. Let H be the tree after moving Y''_k to x_k. Also, let $D'_k = f_{y_k}(Y'_k)$ and $D''_k = f_{y_k}(Y''_k)$.

Notice that $D^G_k = D'_k D''_k$. The following are found as before,

$$\begin{aligned}
f_{x_k}(G \sim Y_{\geq k}) &= C^G_k(1 + C^G_{\geq k+1})(1 + C),\\
f_{y_k}(G \sim X_{\geq k}) &= D'_k D''_k(1 + D^G_{\geq k+1})(1 + D),\\
f_{x_k}(H \sim Y_{\geq k}) &= C^G_k D''_k(1 + C^G_{\geq k+1})(1 + C),\\
f_{y_k}(H \sim X_{\geq k}) &= D'_k(1 + D^G_{\geq k+1})(1 + D).
\end{aligned}$$

Hence,

$$\begin{aligned}
&f(H) - f(G)\\
&= C^G_k D''_k(1 + C^G_{\geq k+1})(1 + C) + D'_k(1 + D^G_{\geq k+1})(1 + D)\\
&\quad - C^G_k(1 + C^G_{\geq k+1})(1 + C) - D'_k D''_k(1 + D^G_{\geq k+1})(1 + D)\\
&= (D''_k - 1)(C^G_k - D'_k)[(1 + C^G_{\geq k+1})(1 + C) - (1 + D^G_{\geq k+1})(1 + D)]\\
&\geq 0.
\end{aligned}$$

Notice $f(H) - f(G) > 0$ if $C_j > D_j$ for any $1 \leq j \leq n$ and if D''_k contains more than one vertex. □

The following algorithm applies Lemma 4.3.1. Notice that the algorithm terminates after any degree-switch. This is because the conditions for an increase in the number of subtrees as a result of a degree-switch requires the C_k^G and D_k^G components to be ordered according to Figure 4.11, which is no longer certain after D''_k has been moved to x_k. Then the path must be resorted by the Phase I and II algorithms after every degree switch.

Algorithm 4.3.2. ***(Phase III Switching Algorithm)*** *Let G be a tree, with a path from leaf vertex v_1 to leaf vertex v_2 that has been sorted by the Phase I and Phase II Switching Algorithms. Then $H = P_3^G(v_1, v_2)$ is given below.*

```
for $i = 1$ to $n$ do
  if $d(x_i) < d(y_i)$ then
    $H = R^G_{v_1,v_2}(x_i, y_i)$
    break {Terminate algorithm}
  end if
  if $d(y_i) < d(x_{i+1})$ then
    $H = R^G_{v_1,v_2}(y_i, x_{i+1})$
    break {Terminate algorithm}
  end if
end for
```

4.4 The complete switching algorithm

We now introduce our final algorithm which encompasses all three phases discussed above, wherein every step of the algorithm increases the number of subtrees.

Algorithm 4.4.1. ***(Switching Algorithm)*** *Let* G *be a tree with leaf vertices* $v_1, v_2, \ldots, v_l$. *Then* $H = SA(G)$ *is given below.*

```
$H = (\emptyset, \emptyset)$
$J = G$
$m = 0$
$G_m = 0$
while $H \neq J$ {Terminate when every path is sorted} do
  $H = J$
  $J = (\emptyset, \emptyset)$
  for $i = 1$ to $l$ do
    for $j = 1$ to $l$ do
      while $J \neq G_m$ {Terminate when path is sorted} do
        $J = G_m$
        $G_m = P_1^{G_m}(v_i, v_j)$ {Phase I}
        $G_m = P_2^{G_m}(v_i, v_j)$ {Phase II}
        $m = m + 1$
        $G_m = P_3^{G_{m-1}}(v_i, v_j)$ {Phase III}
      end while
    end for
  end for
end while
```

Remark 4.4.2. *Suppose $H = SA(G)$, where G is a tree with given degree sequence. Then for any path in H from one leaf vertex to another leaf vertex, we can label the vertices on the path according to Figure 4.11. On such paths we have,*

$$C_1^H \geq D_1^H \geq C_2^H \geq \ldots \geq C_n^H \geq D_n^H,$$
$$C_{\geq 1}^H \geq D_{\geq 1}^H \geq C_{\geq 2}^H \geq \ldots \geq C_{\geq n}^H \geq D_{\geq n}^H,$$
$$\text{and } d(x_1) \geq d(y_1) \geq d(x_2) \geq \ldots \geq d(x_n) \geq d(y_n).$$

for paths of odd length $2n-1$, and

$$C_1^H \geq D_1^H \geq C_2^H \geq \ldots \geq D_n^H \geq C_{n+1}^H,$$
$$C_{\geq 1}^H \geq D_{\geq 1}^H \geq C_{\geq 2}^H \geq \ldots \geq D_{\geq n}^H \geq C_{\geq n+1}^H,$$
$$\text{and } d(x_1) \geq d(y_1) \geq d(x_2) \geq \ldots \geq d(y_n) \geq d(x_{n+1}).$$

for paths of even length $2n$.

We introduce the following conjecture here:

Conjecture 4.4.3. *Suppose that H is a tree such that for any path between leaf vertices we have*

$$C_1^H \geq D_1^H \geq C_2^H \geq \ldots \geq C_n^H \geq D_n^H$$

for a path of odd length $2n-1$, and

$$C_1^H \geq D_1^H \geq C_2^H \geq \ldots \geq D_n^H \geq C_{n+1}^H$$

for a path of even length $2n$. Then we must have that

$$d(x_1) \geq d(y_1) \geq d(x_2) \geq \ldots \geq d(x_n) \geq d(y_n)$$

for the path of odd length, and

$$d(x_1) \geq d(y_1) \geq d(x_2) \geq \ldots \geq d(y_n) \geq d(x_{n+1})$$

.

The proof of this conjecture would make the Phase III algorithm unnecessary because every path between leaf vertices would already be sorted.

CHAPTER 5

MAXIMIZING THE NUMBER OF SUBTREES OF A TREE WITH GIVEN DEGREE SEQUENCE

5.1 Introduction

Suppose we have a tree with given degree sequence. A *greedy tree* can be constructed, as was listed in [29], in the following way:

Definition 5.1.1. *Suppose the degrees of the non-leaf vertices are given, the greedy tree is achieved by the following 'greedy algorithm':*

i) Label the vertex with the largest degree as v *(the root);*

ii)Label the children of v *as* $v_1, v_2, \ldots$, *assign the largest degrees available to them such that* $d(v_1) \geq d(v_2) \geq \ldots$ *;*

iii) Label the children of v_1 *as* $v_{11}, v_{12}, \ldots$ *such that* $d(v_{11}) \geq d(v_{12}) \geq \ldots$ *then do the same for* $v_2, v_3, \ldots$ *respectively;*

iv) Repeat (iii) for all the newly labelled vertices, always start with the children of the labelled vertex with largest degree whose neighbors are not labelled yet.

Fig. 5.1 shows a greedy tree with degree sequence $\{4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 2, 2\}$.

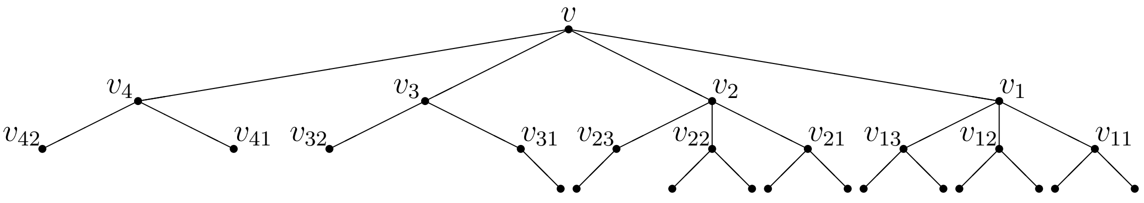


Figure 5.1: A greedy tree

The lemma below, found in [29], follows immediately from the definition of the greedy tree.

Lemma 5.1.2. *A rooted tree T with a given degree sequence is a greedy tree if:*

i) the root v has the largest degree;

ii) the heights of any two leaves differ by at most 1;

iii) for any two vertices u and w, if $h_T(w) < h_T(v)$, then $d(w) \leq d(u)$;

iv) for any two vertices u and w of the same height, $d(u) > d(w) \Rightarrow d(u') \geq d(w')$ for any successors u' of u and w' of w that are of the same height;

v) for any two vertices u and w of the same height, $d(u) > d(w) \Rightarrow d(u') \geq d(w')$ and $d(u'') \geq d(w'')$ for any siblings u' of u and w' of w or successors u'' of u' and w'' of w' of the same height.

From the Switching Algorithm developed in Chapter 4, Algorithm 4.4.1, we know that the maximal tree has the following three properties for any path from one leaf vertex to another:

$$\begin{aligned}
&C_1^H \geq D_1^H \geq C_2^H \geq \ldots \geq C_n^H \geq D_n^H,\\
&C_{\geq 1}^H \geq D_{\geq 1}^H \geq C_{\geq 2}^H \geq \ldots \geq C_{\geq n}^H \geq D_{\geq n}^H,\\
&\text{and } d(x_1) \geq d(y_1) \geq d(x_2) \geq \ldots \geq d(x_n) \geq d(y_n).
\end{aligned}$$

for paths of odd length $2n-1$, and

$$\begin{aligned}
&C_1^H \geq D_1^H \geq C_2^H \geq \ldots \geq D_n^H \geq C_{n+1}^H,\\
&C_{\geq 1}^H \geq D_{\geq 1}^H \geq C_{\geq 2}^H \geq \ldots \geq D_{\geq n}^H \geq C_{\geq n+1}^H,\\
&\text{and } d(x_1) \geq d(y_1) \geq d(x_2) \geq \ldots \geq d(y_n) \geq d(x_{n+1}).
\end{aligned}$$

for paths of even length $2n$. Using these properties we find the maximal and second maximal structures with respect to the number of subtrees.

5.2 Maximal tree

Theorem 5.2.1. *Given the degree sequence and the number of vertices, the greedy tree maximizes the number of subtrees for a tree with given degree sequence.*

Proof. The optimal tree must possess the properties given by the Switching Algorithm. We show that such a tree satisfies the properties of the greedy tree listed in Lemma 5.1.2. The diametral path of maximal tree G is the path of greatest length in G. If there are multiple paths having the greatest length, choose the path so that C_1 and $C_{\geq 1}$ is at least as big on that path as any other path.

PROPERTIES (i) and (ii). Consider the diametral path, from leaf vertex v_1 to v_2 in G and label the vertices such that $C_1 \geq D_1 \geq C_2 \geq \ldots \geq C_n \geq D_n$ for a path of odd length or $C_1 \geq D_1 \geq C_2 \geq \ldots \geq D_n \geq C_{n+1}$ for a path of even length. Suppose $d(x_1) = m$, and that we label the children of x_1 as $x_{1,1}, \ldots, x_{1,m}$, with v_1 in the tail of $x_{1,1} = x_2$ and v_2 in the tail of $x_{1,2} = y_1$. For the remainder of this case, suppose that the path is of odd length. The other case is similar.

Now consider a path from v_1 to another leaf vertex in the tail of $x_{1,2}$ (if there is not another leaf vertex in $x_{1,2}$ choose the tail of $x_{1,3}$, and so on). We know that the length of this new path must be less than or equal to the length of the diametral path. Recall that $d(x_1) \geq d(x_2) \geq \ldots \geq d(x_n)$. If we label the remaining vertices on the path $x_{1,2} = u_1, u_2, \ldots u_k$, we must have $n-1 \leq k \leq n$. Otherwise we would not have $d(x_1) \geq d(u_1) \geq d(x_2) \geq \ldots \geq d(u_{n-1}) \geq d(x_n)$ or $d(x_1) \geq d(u_1) \geq d(x_2) \geq$

$\ldots \geq d(x_n) \geq d(u_n)$, which must be true of every path between leaf vertices for the optimal tree.

Repeating this argument for paths between v_1 and any other leaf vertex not in the tail of $x_{1,1}$ gives us that for all leaf vertices, v_i, not in the tail of $x_{1,1}$, the length of the path from v_1 to v_i is equal to $2n-1$ or $2n-2$. By a similar argument one also gets that the path between v_i and all leaf vertices in the tail of $x_{1,1}$ must have length equal to $2n-1$ or $2n-2$. Thus, if we root G at x_1, we must have that $d(x_1)$ has the largest degree and that the heights of all leaf vertices differ by at most 1.

PROPERTIES (iii), (iv), and (v). Consider a path from any leaf vertex v_i to any other leaf vertex v_j. Suppose that $u, w \in V(G)$ lie on the path, and that $h(u) < h(w)$. Then if z is the closest common ancestor of u and w, we have that u is closer to z than w is to z. We can label the path such that $z = x_1$ and $d(x_1') \geq d(y_1)' \geq d(x_2') \ldots$, and thus $d(z,u) < d(z,w) \Longrightarrow d(u) > d(w)$.

For u and w the same height, with $d(u) > d(w)$, suppose z is the closest common ancestor. For any path between leaf vertices containing u and w, we can label the path such that $d(x_1') \geq d(y_1)' \geq d(x_2') \ldots$. Thus, we have $u = x_i'$, $w = y_i'$, and for any successor u' of u and successor w' of w having the same height, the $d(u') \geq d(w')$.

$\square$

5.3 Second maximal tree

We now know that the structure of the tree produced by the switching algorithm is a greedy tree. The greedy tree is the unique structure that maximizes the number of subtrees of a tree, thus we now know that the algorithm always converges to the maximal tree. The following corollary then follows immediately from Theorem 4.4.1.

Corollary 5.3.1. *Starting with the optimal tree, any tree with the same degree sequence can be achieved in a finite number of switchings wherein every switching leads to a decrease in the optimal tree.*

Proof. Let G be the optimal tree and H be the tree we wish to obtain. Then $G = SA(H)$ is a chain of switchings from H to G wherein every switch leads to an increase in the number of subtrees. By reversing the order of the chain of switchings we obtain a new chain of switchings from G to H wherein every switch leads to a decrease in the number of subtrees.

□

Corollary 5.3.2. *Let G be the optimal tree. The n-th maximal tree can be achieved with at most $n-1$ switches from the maximal tree.*

Proof. Let H be the n-th maximal tree. There exists a chain of switchings of length m to get from H to G wherein every switch leads to a tree with a greater number of subtrees than the previous tree. Then if $m > n-1$, there exists at least n trees with a greater number of of subtrees than H, which contradicts our assumption that H is the n-th optimal tree.

□

Hence, we now know that the second maximal tree is exactly one switch away from the maximal tree. The switch could be a component switch, tail switch, or an degree-switch.

Lemma 5.3.3. *Let G be the maximal tree. For any component switch or degree-switch outlined by Phase I, II, or III of the Switching Algorithm, there exists a tail switch which produces a decrease in the number of subtrees at least as small as that of the component switch or degree-switch, i.e. the tree produced by the tail switch has at least as many subtrees as either of the trees produced by a component switch or degree-switch.*

Proof. **CASE 1**. Suppose we have the situation outlined in Lemma 4.1.1. In the maximal tree we have $X_k < X_{k+1}$ and we wish to switch them to make a decrease in the number of subtrees. Notice that switching the components is the same as switching the tails in that scenario, as desired.

CASE 2. Suppose that we have the situation outlined in Lemma 4.2.3. In the maximal tree we have $X_k \geq Y_k$ and we wish to switch them to make a decrease in the number of subtrees. If $X_k = Y_k$, then switching them will result in no change, thus, assume $X_k > Y_k$. Notice that the branches that make up X_k and Y_k are actually tails of paths between different pairs of leaf vertices. Recall that $d(x_k) \geq d(y_k)$. Hence, if X_k has only one branch, this forces Y_k to have only one branch as well. Thus, switching X_k and Y_k is effectively a tail switch and we are done.

Assume now that X_k has more than one branch. Because G is a greedy tree, every branch in X_k has at least as many subtrees as the branch with the most subtrees in Y_k. Then switch the smallest branch in X_k with the largest branch in Y_k. One can easily find, using similar methods used in the proofs of Lemmas 4.1.1, 4.2.3, and 4.3.1, that such a tree has more subtrees than the tree produced by switching X_k and Y_k. The same can be shown for the switching of Y_k with X_{k+1}.

CASE 3. Suppose we have the situation outlined in Lemma 4.3.1, with $d(x_k) > d(y_k)$ $(d(y_k) > d(x_{k+1}))$. Similar to the proof for **CASE 2**, one can show that switching the smallest branch of X_k (Y_k) with the largest branch of Y_k (X_{k+1}) will result in a smaller decrease than swapping the degrees of X_k and Y_k (Y_k and X_{k+1}). □

As a consequence of Lemma 5.3.3, we have that the second maximal tree is the maximum of all trees that are one tail switch away from the optimal tree.

5.4 Conjectures and future work

The following conjectures would help to reduce the number of tail switches we must check in order to find the second maximal tree.

Conjecture 5.4.1. *Suppose* $k \geq 2$, *and that switching* $C_{\geq k}$ *and* $D_{\geq k}$ *decreases the number of subtrees. Then switching* $D_{\geq k}$ *and* $C_{\geq k+1}$ *or* $C_{\geq k+1}$ *and* $D_{\geq k+1}$ *will produce a smaller decrease.*

Conjecture 5.4.2. *The second optimal tree is obtained by a tail switch within the two levels of greatest height in the optimal tree.*

CHAPTER 6

CONCLUSIONS

6.1 On the Randić index and extremal cacti

We make a modest progress on generalizing the results on extremal cacti for the Randić index by showing that the extremal cacti have no cut edge unless it is a pendant edge. This result leads to the other case that is based on a plausible conjecture, and is proven for several specific α. The conjecture might be an interesting problem in the research of analytic number theory or differential equations. The approaches discussed can be easily employed to study the same problem for any specific value of α. In particular the case $\alpha = 1$ is presented and variations of this problem can be easily developed.

6.2 The degree distance v.s. the Wiener index

We considered the correlation between the degree distance and the Wiener index of several graphs. In particular, a nice formula with an error term is presented for cycles, unicyclic graphs, and cacti in general. We also analyzed this error term for unicyclic graphs and cacti, providing sharp bounds and the extremal cacti that achieve these bounds. Interestingly some of these extremal structures coincide with the ones that provide extremal values of some other graphical indices. A potential next step is to extend this result to hexagon systems and conformational changes of molecular graphs.

6.3 Algorithms on switching components in trees

A switching algorithm is developed, wherein every step in the algorithm is shown to increase the number of subtrees. The final result of this algorithm is a characterization of the maximal tree. In Chapter 5, the algorithm is used to find the unique tree that maximizes the number of subtrees among all trees with the given degree sequence. The algorithm itself is not very efficient. To potentially increase the efficiency of the algorithm, a conjecture is introduced which would make the third phase of the algorithm unnecessary.

6.4 Maximizing the number of subtrees

Using the algorithm developed in Chapter 4, we find that the tree resulting from the switching algorithm is the unique structure that maximizes the number of subtrees. Several lemmas are then presented and proven, and two conjectures are introduced, to characterize the second maximal tree.

REFERENCES

[1] Bollobás, B., Erdos, P., Graphs of extremal weights, *Ars Combin.* **50** (1998), 225–233.

[2] Dankelmann, P., Gutman, I., Mukwembi, S., Swart, H.C., On the degree distance of a graph, *Discrete Appl. Math.* **157** (2009), 2773–2777.

[3] Delorme, C., Favaron, O., Rautenbach, D., Closed formulas for the numbers of small independent sets and matchings and an extremal problem for trees, *Discrete Applied Math.* **130** (2003), 503–512.

[4] Delorme, C., Favaron, O., Rautenbach, D., On the Randić index, *Discrete Math.* **257** (2002), 29–38.

[5] Dobrynin, A.A., Entringer, R., Gutman, I., Wiener index of trees: theory and applications, *Acta Appl. Math.* **66** (2001), 211–249.

[6] Dobrynin, A.A., Kochetova, A.A., Degree distance of a graph: a degree analogue of the Wiener index, *J. Chem. Inf. Comput. Sci.* **34** (1994), 1082–1086.

[7] Du, Z., Zhou, B., Minimum Wiener Indices of Trees and Unicyclic Graphs of Given Matching Number, *MATCH.*, **63** (1) (2010), 101–112.

[8] Fischermann, M., Hoffmann, A., Rautenbach, D., Székely, L.A., Volkmann, L., Wiener index versus maximum degree in trees, *Discrete Appl. Math.* **122** (1–3) (2002), 127–137.

[9] Gray, D., Wang, H., On the Randić index and extremal cacti, *To appear in Congressus Numerantium.*

[10] Gray, D., Wang, H., The degree distance v.s. the Wiener index. *Submitted.*

[11] Gray, D., Wang, H., Maximum number of subtrees for a tree with given degree sequence, *In preparation.*

[12] Gutman, I., Selected properties of the Schultz molecular topological index, *J. Chem. Inf. Comput. Sci.*, **34** (1994), 1087–1089.

[13] Jelen, F., Triesch, E., Superdominance order and distance of trees with bounded maximum degree, *Discrete Appl. Math.* **125** (2-3), 225–233.

[14] Kirk, R., Wang, H., Largest number of subtrees of trees with given maximum degree, *SIAM Journal on Discrete Mathematics* **22** (3) (2008), 985–999.

[15] Klein, D.J., Mihalić, Z., Plavšić, D., Trinajstić, N., Molecular topological index: A relation with the Wiener index, *J. Chem. Inf. Comput. Sci.*, **32** (1992), 304–305.

[16] Li, X., Shi, Y., A Survey on the Randić index, *Match* **59** (2008), 127-156.

[17] Lin, A., Luo, R., Zha, X., A sharp lower bound of the Randić index of cacti with r pendants, *Discrete Applied Mathematics* **156** (2008), 1725–1735.

[18] Lu, M., Liu, H., Tian, F., The connectivity index, *MATCH Commun Math. Comput. Chem.* **51** (2004), 149–154.

[19] Lu, M., Zhang, L., Tian, F., On the Randić index of cacti, *MATCH Commun Math. Comput. Chem.* **56** (2006), 551–556.

[20] Liu, H., Lu, M., Tian, F., Trees of extremal connectivity index, *Discrete Applied Math.* **154** (2006), 106–119.

[21] Liu, H., Lu M., A Unified Approach to Cacti for Different Indices, *MATCH.*, **58** (2007), 193–204.

[22] Randić, M., On characterization of molecular branching, *J. Am. Chem. Soc.* **97** (1975) 6609–6615.

[23] Rautenbach, D., A note on trees of maximum weight and restricted degrees, *Discrete Math.* **271** (2003), 335–342.

[24] Szekély, L.A., Wang, H., On subtrees of trees, *Adv. Appl. Math.* **34** (2005), 138–155.

[25] Szekely, L.A., Wang, H., Binary trees with the largest number of subtrees with at least one leaf, *Congressus Numerantium* **177** (2005), 147–169.

[26] Tomescu, A.I., Properties of connected graphs having minimum degree distance, *Discrete Appl. Math.* **309** (2009), 2745–2748.

[27] Tomescu, A.I., Unicyclic and bicyclic graphs having minimum degree distance, *Discrete Appl. Math.* **156** (2008), 125–130.

[28] Wang, H., Extremal trees with given degree sequence for the Randić index, *Discrete Mathematics* **308** (2008), 3407–3411.

[29] Wang, H., The extremal values of the Wiener index of a tree with given degree sequence, *Discrete Applied Mathematics* **156** (2008), 2647-2654.

[30] Wiener, H., Structural determination of paraffin boiling points, *J. Amer. Chem. Soc.* **69** (1947), 17–20.