Clemson University
TigerPrints

All Theses | Theses

12-2023

Generalized Vulnerability Measures of Graphs

Julia VanLandingham
jvanlan@clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_theses

Part of the Discrete Mathematics and Combinatorics Commons

Recommended Citation

VanLandingham, Julia, "Generalized Vulnerability Measures of Graphs" (2023). *All Theses*. 4206.
https://tigerprints.clemson.edu/all_theses/4206

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.

Generalized Vulnerability Measures of Graphs

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science
Mathematical Sciences

by
Julia VanLandingham
December 2023

Accepted by:
Dr. Wayne Goddard, Committee Chair
Dr. Beth Novick
Dr. Matthew Macauley

Abstract

Several measures of vulnerability of a graph look at how easy it is to disrupt the network by removing/disabling vertices. As graph-theoretical parameters, they treat all vertices alike: each vertex is equally important. For example, the integrity parameter considers the number of vertices removed and the maximum number of vertices in a component that remains. We consider the generalization of these measures of vulnerability to weighted vertices in order to better model real-world applications. In particular, we investigate bounds on the weighted versions of connectivity and integrity, when polynomial algorithms for computation exist, and other characteristics of the generalized measures.

Acknowledgments

I would like to extend great thanks to my advisor Dr. Wayne Goddard for guiding me through this project and for being an overall fantastic advisor. Also to my other committee members, Dr. Beth Novick and Dr. Matthew Macauley, for their service and for taking the time to read through and provide very detailed edits to my paper.

Table of Contents

List of Figures

Chapter 1

Introduction

There are many ways one can quantify the "goodness" of a graph as a network. In this thesis we focus on the case of node failure and the worse case thereof. The associated parameters are sometimes called measures of vulnerability. That is, the vulnerability of a graph is a measurement of how the removal or disabling of a subset of vertices disrupts the graph. Multiple measures of vulnerability exist and are each applicable for different kinds of situations. Some of the most common measures of vulnerability are the connectivity, integrity, and toughness. In 1988, Goddard [7] investigated the integrity and toughness in general and how they relate to vulnerability.

The connectivity is a very natural measure of the vulnerability of a graph and lends itself nicely to a basic intuition of how resilient a graph is to disruption. Using this as a measure of vulnerability is also convenient, as there are many known results about connectivity. This includes helpful bounds such as the following.

Proposition 1.1. *The connectivity of a graph is at most the minimum degree of the graph.*

However, oftentimes the connectivity of a graph can be over-sensitive to local weaknesses while the rest of the graph may still be intact, though disconnected. For example, the star and the graph obtained by adding a leaf to the complete graph both have connectivity 1. However, they differ greatly in the damage done to the overall network by the removal of a cut-vertex [3]. This leads us to investigate other measures that may give a better grasp of the overall disruption that disconnection causes. In this thesis we focus on integrity and toughness.

The integrity of a graph can be thought of as how resistant a graph is to fragmentation, and attempts to provide a measure of how disconnected the graph becomes. Integrity was introduced by Barefoot, Entringer, and Swart in 1987 [3] specifically as a measure of vulnerability and then investigated in more depth in 1988 by Goddard [7].

Definition 1.1. For any graph G, the **integrity** of G, denoted $I(G)$, is defined as

$$I(G) := \min_{S \subset V(G)} \Big\{ |S| + m(G - S) \Big\},$$

where $m(G)$ is the maximum cardinality of a component of G.

Integrity has several very nice properties including that the removal of a single vertex can only lower it by at most 1 [7]. Also, it is at least the chromatic number of the graph. Goddard outlines more formulas for computation, bounds, and theorems regarding integrity in [7]. For a survey, see [1]. There are several drawbacks to integrity as well, a major one of which is that, while there are some formulas for specific classes of graphs such as paths, in general it is NP-complete to calculate [7].

The toughness of a graph was introduced in 1973 by Chvátal [5] as a graphical parameter related to work with Hamiltonicity rather than directly as a measure of vulnerability. Toughness also provides a measure of a graph's resilience to fracture, but takes into account the number of fragments left instead of the size of fragments as in integrity.

Definition 1.2. For any graph G, the **toughness** of G, denoted by $\tau(G)$, is defined as

$$\tau(G) := \min_{S \in J(G)} \left\{ \frac{|S|}{k(G-S)} \right\},$$

where $J(G) := \{S \subset V(G) : S \text{ is a cut set or } G-S \text{ is trivial}\}$ and $k(G)$ is the number of components of G.

Much of the work on toughness has been about thresholds for the existence of particular subgraphs, especially a Hamilton cycle. However, there has also been work on bounds and complexity. It too is NP-hard to calculate in general [7]. For a survey, see [4].

1.1 Our Contribution

In the previously mentioned measures of vulnerability, all vertices in a graph are considered equally important. In many real-world applications, this assumption is inaccurate as some vertices many be more important or harder to disable than others. To better model these situations, we generalize several of the vulnerability parameters to weighted graphs.

We first review some background definitions and theorems in Section 2. Then we discuss weighted integrity in Section 3 and fully weighted toughness in Section 4. Lastly we discuss other ideas and future work in Section 5.

Chapter 2

Graph Theory Background

2.1 Basic Definitions

We will use the following definitions and notation throughout this paper. Note that we only deal with undirected simple graphs throughout. We use the following notation for standard classes of graphs.

- K_n - the complete graph of order n
- $K_{r,s}$ - the complete bipartite graph with partite sets of order r and s
- P_n - the path on n vertices
- C_n - the cycle on n vertices

We also use the following definitions for standard graph parameters.

Definition 2.1. The **degree** of a vertex v is the number of its neighbors. The maximum vertex degree is denoted as Δ and the minimum vertex degree as δ.

Definition 2.2. A nonempty graph G is **connected** if every pair of vertices are connected via a path, otherwise it is **disconnected**.

Definition 2.3. A set of vertices S is a **cut-set** of the graph G if $G - S$ is disconnected.

Definition 2.4. The **connectivity** of a graph G, denoted $\kappa(G)$, is the minimum cardinality of S such that $G - S$ is disconnected or reduced to a single vertex.

Definition 2.5. A set of vertices S is **independent** if they are pairwise non-adjacent.

Definition 2.6. The **independence number** of a graph G, denoted $\alpha(G)$, is the maximum cardinality of an independent set.

Definition 2.7. A subset of vertices C is a **vertex cover** of G if every edge of G is incident to some vertex of C.

Definition 2.8. The **covering number** of G, denoted $\beta(G)$, is the minimum cardinality of a vertex cover of G.

Definition 2.9. A proper (vertex) k-coloring of a graph G is defined as a vertex coloring from a set of k colors such that no two adjacent vertices share a common color. That is, a proper k-coloring of G is a mapping $c : V \to \{1, 2, \ldots, k\}$ such that: $\forall e = \{u, v\} \in E : c(u) \neq c(v)$.

Definition 2.10. The **chromatic number** of G, denoted $\chi(G)$, is the minimum number of colors in a proper coloring of G.

2.2 Background on Integrity

Various results about ordinary integrity are known, including formulas for calculating the integrity of specific classes of graphs. In 1987, Barefoot, Entringer, and Swart [2, 3] proved the following about specific graphs.

Theorem 1. *The integrity of:*

1. *the complete graph* K_n *is* n

2. *the edgeless graph is 1*

3. *the star* $K_{1,n}$ *is* 2

4. *the path* P_n *is* $\lceil 2\sqrt{n+1} \rceil - 2$

5. *the cycle* C_n *is* $\lceil 2\sqrt{n} \rceil - 1$

6. *the complete bipartite graph* $K_{r,s}$ *is* $1 + \min\{r, s\}$

7. *any complete multipartite graph of order* n *and largest partite set of order* r *is* $n - r + 1$.

In 1990, Goddard and Swart [6] gave an alternative formulation of integrity along with a characterization of some vertices that achieve the minimum presented.

Theorem 2. *The integrity of a nontrivial graph* G *is,*

$$I(G) = \min_{v \in V} \{m(G), 1 + \min I(G - v)\}.$$

If G *is connected, then this can be restated as*

$$I(G) = 1 + \min_{v \in V} I(G - v).$$

Theorem 3. *If a graph* G*, has a vertex* v *for which* $deg(v) \geq I(G - v)$*, then* $I(G) = 1 + I(G - v)$.

We also have the following two theorems that relate the integrity of a graph to other interesting parameters. The first gives bounds on integrity that are sharp and the second shows cases when equality holds [1].

Theorem 4. *For any graph G of order n,*

1. $I(G) \leq \beta(G) + 1$

2. $I(G) \geq \delta(G) + 1$

3. $I(G) \geq \min_t \max\{d_t, t-1\}$ *where the degrees of G are $d_1 \geq d_2 \geq \cdots \geq d_n$*

4. $I(G) \geq \chi(G)$

5. $I(G) \geq (n - \kappa(G))/\alpha(G) + \kappa(G)$

6. $I(G) \geq 2\sqrt{\tau n} - \tau$*, if G is not complete, where τ is the toughness.*

Recall that $2K_2$ is the graph of two disjoint copies of K_2.

Theorem 5. *For any graph G, the integrity is*

1. $I(G) = \kappa(G) + 1$ *if and only if* $\kappa(G) = \beta(G)$

2. $I(G) = \beta(G) + 1$ *if and only if G does not contain $2K_2$ as an induced subgraph*

3. $I(G) = \delta(G) + 1$ *if and only if $G \cong rK_n$ or $G \cong rK_n + F$ for some graph F satisfying* $\delta(F) \geq |G| - (2r-1)n - 1$.

2.3 Background on Toughness

As mentioned before, toughness was originally defined to help determine a condition for when a graph is Hamiltonian. Thus, we have many theorems relating these two properties. We also have various theorems relating toughness to the existence of factors in a graph. More can be read about these relationships in [4].

As with integrity, determining the toughness of a general graph is NP-hard and even determining if a graph is 1-tough, meaning that the toughness is at least

1, is NP-hard. Recall that a cubic graph is one in which every vertex has degree 3. Determining the toughness of cbuic graphs is still NP-hard; however, we do have an upper bound on the toughness in terms of its independence number [4].

Theorem 6. *Let G be a noncomplete cubic graph on n vertices with independence number α. Then*

$$\tau(G) \leq \min\left\{\frac{2n-3\alpha}{n-\alpha}, \frac{2\alpha}{4\alpha-n}\right\}.$$

In 1997, Goddard [8] considered the toughness of a specific type of cubic graph. A *cycle permutation graph* is obtained from two disjoint cycles of the same length by adding a matching between the cycles. The following are bounds on the toughness of these graphs based on their order [8].

Theorem 7. *Let G be a cycle permutation graph on $2m$ vertices. Then*

$$\tau(G) \begin{cases} \leq 4/3 & m \equiv 0,1 \mod 4 \\ < 4/3 & m \equiv 2 \mod 4 \\ \leq 4/3 + 4/(9m-3) & m \equiv 3 \mod 4. \end{cases}$$

We also have lower bounds on the toughness of a general connected graph in terms of its connectivity and genus [4]. Recall that the genus of a graph is the minimum genus of an orientable surface on which the graph can be drawn without edges crossing.

Theorem 8. *Let G be a connected graph with connectivity κ and genus γ. Then*

1. $\tau > \dfrac{\kappa}{2} - 1$ *if* $\gamma = 0$
2. $\tau \geq \dfrac{\kappa(\kappa-2)}{2(\kappa-2+2\gamma)}$ *if* $\gamma \geq 1$.

The next theorem gives sufficient conditions to find a 1-tough spanning subgraph of a given graph [4].

Theorem 9. *Let G be a graph on $n \geq 4$ vertices with $\tau(G) > 1$. Then there exists a spanning subgraph H of G with $\tau(H) = 1$.*

Recently, Matthews and Sumner [10] showed that the toughness is exactly half the connectivity if the graph is claw-free.

Chapter 3

Weighted Integrity

The weighted integrity was introduced by Ray et al. in 2006 [13]. They defined the problem and showed results about its complexity, discussed later in Section 3.2. We take a different viewpoint, and instead, investigate bounds on the value of the weighted integrity as well as other properties. We consider graphs with weightings on the vertices where the weight of each vertex is nonnegative and the overall weight of the graph is positive.

Definition 3.1. Let G be a graph with weighting w. The **weighted integrity** of G, denoted $I_w(G)$, is

$$I_w(G) := \min_{S \subseteq V(G)} \Big\{ w(S) + m_w(G - S) \Big\},$$

where $m_w(X)$ denotes the weight of the component of X that has largest weight and $w(X)$ denotes the sum of the weights of vertices in X.

For example, C_4 with an associated weighting w and its weighted integrity is shown below in figure 3.1.

Figure 3.1: $I_w(G) = 3.5$

In this section, we discuss some exact formulas for computing the weighted integrity of special classes of graphs, some we discuss bounds for the weighted integrity of a general graph, and finally, we state some interesting properties of the weighted integrity.

3.1 Fundamental Bounds

In order to compare integrity and weighted integrity fairly, we define a **normalized weighting** as one where the total weight of the vertices sums to the order of the graph. Notice this also means that when all weights are 1, we recover the ordinary integrity of the graph exactly. Also note that we can normalize any given weighting of a graph.

We first investigate the weighted integrity of the two most extreme cases: when G is edgeless, and when G is complete.

Proposition 3.1. *For a normalized weighting w of a graph G, $I_w(G) \geq 1$. Further, $I_w(G) = 1$ if and only if G is edgeless and every vertex has weight 1.*

Proof. Suppose first that G is edgeless with every vertex having weight 1. Then clearly $I_w(G) = 1$, because we can remove no vertices and all components will have weight 1.

Now suppose that $I_w(G) = 1$. Notice if the weight of any vertex is greater than 1, then either S has weight more than 1, or there will be a component of $G - S$ of weight greater than 1, and thus $I_w(G) > 1$. So all vertices must have weight 1. But then if any two vertices are connected, either $w(S) \geq 1$ and $m_w(G - S) \geq 1$, or we have a component of weight greater than 1. So then G is edgeless. ■

Proposition 3.2. *For any normalized weighting w, $I_w(K_n) = n$.*

Proof. Notice first that for any subset S of vertices removed, the remaining graph will still be connected (or vertexless) and thus the weight of the remaining component of largest weight is exactly $n - w(S)$. So then $I_w(G) = w(S) + n - w(S) = n$. ■

If G is not complete, then we may assume S is a cut set. Because otherwise, $G - S$ is connected and thus $m_w(G - S) = w(G - S)$, so that the sum would be n and thus not minimized. We now consider the complete bipartite graph.

Proposition 3.3. *Let $K_{r,s}$ be the complete bipartite graph with weighting w and let $x_1, \ldots, x_r$ represent the weights of the vertices in the first partite set and $y_1, \ldots, y_s$ the weights of the vertices in the second partite set. Then*

$$I_w(K_{r,s}) = \min \left\{ \left(\sum_{i=1}^{r} x_i \right) + \max_{1 \leq j \leq s} \{y_j\}, \left(\sum_{j=1}^{s} y_j \right) + \max_{1 \leq i \leq r} \{x_i\} \right\}.$$

Proof. We know that the set removed to achieve the weighted integrity must be a cut set of the graph. So one must remove all vertices from one of the two partite sets. Upon removing all the vertices in one set, the remaining component of largest weight will just be the vertex of largest weight from the other partite set. So the weighted integrity is the minimum of the two values obtained from removing each set. That gives the desired result. ■

For the above graphs we have given formulas for their weighted integrity given a normalized weighting. However, in general this is not the case because the weighted integrity problem reduces to the integrity problem when all the weights are 1 and finding the integrity of a graph is an NP-complete problem [7]. So, we seek instead a bound on the maximum possible value the weighted integrity of a graph.

Proposition 3.4. *For all graphs G of order n with normalized weighting w, $I_w(G) \leq n$. Furthermore, there exists a normalized weighting w of G such that $I_w(G) = n$.*

Proof. Notice that trivially $I_w(G) \leq n$ since the sum of all the weights equals n.

To achieve this bound, simply give one vertex u weight n and all others weight 0. Either the vertex u contributes to $w(S)$ or to $m_w(G-S)$; and then $I_w(G) = n$. ■

In the previous result we established the maximum possible weighted integrity of a graph. We now determine the minimum weighted integrity. Before we can discuss what the minimum possible value is, we need a lemma.

Lemma 3.5. *For normalized weightings, the minimal possible weighted integrity of a noncomplete graph G occurs when the removed set S has weight 0 and the rest of the weight is distributed equally amongst the remaining components.*

Proof. Suppose $I_w(G)$ is minimized over all normalized weightings w of G with S the set of vertices so that $0 < w(S) = x$. Then $I_w(G) = w(S) + m_w(G-S)$. Notice that if we rearrange the weighting w such that $w(S) = 0$ while reallocating weights equally to each vertex in $V - S$, then the component of maximum weight of $G - S$ now has weight less than x more than in the original weighting. Thus, $I_{w'}(G) < I_w(G)$, a contradiction of the choice of w.

For any removed set S, the components that remain should have weight evenly spread across them; so the weighted integrity will be minimized when each component has the same weight of $n/k(G-S)$. ■

We now investigate the minimum possible weighted integrity a general graph may have.

Proposition 3.6. *For a graph G of order n, independence number α, and normalized weighting w, $n/\alpha \leq I_w(G)$. Furthermore, there exists a normalized weighting w such that $I_w(G) = n/\alpha$.*

Proof. By Lemma 3.5, we need only consider

$$I_w(G) = \min_{S \subseteq V(G)} \left\{ w(S) + \frac{n - w(S)}{k(G-S)} \right\} = \min_{S \subseteq V(G)} \left\{ \frac{n}{k(G-S)} \right\},$$

where $k(G - S)$ is the number of components remaining after removing S.

The quantity is clearly minimized when the number of components in $G-S$ is maximized. But we know that the maximum number of components that can occur when vertices are removed is equal to α. Thus $k(G-S) \leq \alpha$ and so $\dfrac{n}{\alpha} \leq I_w(G)$.

To show the tightness of this bound, let C be a minimum vertex cover of G and let every vertex of C have weight 0. Recall now that the size of a minimum vertex cover is equal to $n - \alpha$. So there are exactly $n - (n - \alpha) = \alpha$ isolated vertices left in $G - C$. If we weight each of the vertices in $G - C$ with weight n/α, then $I_w(G) = n/\alpha$. ∎

Proposition 3.6 gives the general lower bound for a graph G, from which we deduce the following corollaries.

Corollary 3.7. *For a normalized weighting w, the weighted integrity of a connected graph G of order n is at least $n/(n-1)$.*

Proof. Since a connected graph has independence number at most $n - 1$, the bound follows from Proposition 3.6. ∎

Corollary 3.8. *For a graph G of order n, $I_w(G) = n/(n-1)$ if and only if $G = K_{1,n-1}$, where the vertex of degree $n-1$ has weight 0 and all other vertices have weight $n/(n-1)$.*

Proof. Suppose first that $G = K_{1,n-1}$, where the vertex of degree $n-1$ has weight 0 and all other vertices have weight $n/(n-1)$. Then clearly $I_w(G) = n/(n-1)$.

Now suppose that $I_w(G) = n/(n-1)$. By Lemma 3.5, we know the removed set has weight 0 and the remaining components have weighting $n/(n-1)$. But that means that when we remove one vertex we have $n-1$ components left, which can only occur when $G = K_{1,n-1}$. ■

Corollary 3.9. *The weighted integrity of a k-connected graph of order n is at least $n/(n-k)$.*

Proof. Notice that the minimum size of a cut set of any k-connected graph is at least k by definition. So then it follows from Lemma 3.5 that the weighted integrity must be at least $n/(n-k)$. ■

Corollary 3.10. *The weighted integrity of the complete bipartite graph $K_{r,s}$, where $r < s$ and $r + s = n$, is at least*

$$\frac{n}{n-r} = \frac{n}{s}.$$

Proof. Notice that $\alpha = s$ and apply Proposition 3.6. ■

We now show that for a general graph G it is possible for $I_w(G)$ to take on all real values between the bounds proved in Propositions 3.6 and 3.4.

Proposition 3.11. *For any graph G, there exists a normalized weighting w_x of G such that $I_{w_x}(G) = x$, $\dfrac{n}{\alpha} \leq x \leq n$.*

Proof. Consider the weighting w_x of G where some minimum vertex cover C is given total weight 0, one of the vertices of $G - C$ is given weight x, and all remaining vertices of $G - C$ are given weight $(n - x)/(\alpha - 1)$. Note that the total weight is $x + (\alpha - 1)(n - x)/(\alpha - 1) = n$.

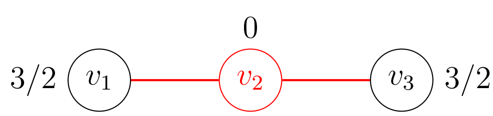
By removing C, we get

$$I_{w_x}(G) = \max\left\{x, \frac{n - x}{\alpha - 1}\right\}$$

Now by rearranging, we see that $x \geq (n - x)/(\alpha - 1)$ whenever $x \geq n/\alpha$ whence $I_{w_x}(G) = x$. ■

3.2 Further Results

In this section, we explore some other interesting properties of the weighted integrity. Recall that for ordinary integrity, the chromatic number of G is a lower bound for $I(G)$. However, a simple counterexample shows that this is not the case for weighted integrity. For example, the path graph P_3 has chromatic number of 2, but by giving the two endpoints weight $3/2$ and the middle vertex weight 0, the weighted integrity would be $I_w(G) = 3/2 < 2 = \chi(G)$.



The following proposition considers a graph constructed by taking each vertex of a graph G and duplicating it such that the new vertex has the same neighborhood as the original vertex, is connected to the original vertex, and has the same weight as the original vertex. We call this graph the **2-duplicative graph of** G and denote it as G^2. In general, we replace each vertex with a k-clique in the same manner and give every vertex the same weight as the original vertex. We call this the **k-duplicative graph of** G and denote it as G^k.

An example is shown below in Figure 3.2:

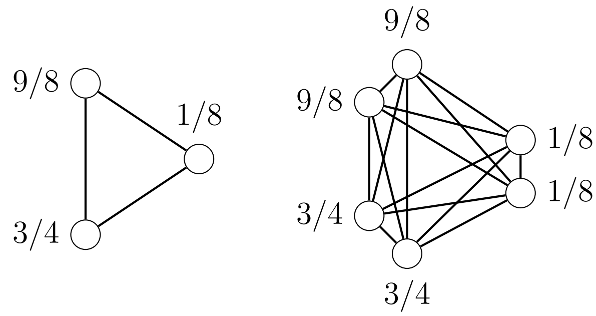


Figure 3.2: K_3 and K_3^2

Proposition 3.12. *If G is a graph of order n with normalized weighting w, then $I_w(G^2) = 2I_w(G)$.*

Proof. We first prove that if S is a set of vertices of G such that $I_w(G) = w(S) + m_w(G-S)$, then S^2 is a subset of vertices of G^2 such that $I_w(G^2) = w(S^2)+m_w(G^2 - S^2)$. Let S be such a subset and for ease of notation, say that the component of largest weight in $G - S$ is M. First notice that for any subset K of vertices of G, since the neighbor set of K^2 is the same as that of K and all weights are duplicated in the construction, we know that $w(K^2) = 2w(K)$. Since $w(M) \geq w(K)$ for all other subsets K of vertices of G, then $2w(M) \geq 2w(K)$. Thus M^2 is the component of largest weight in $G^2 - S^2$ and $w(M^2) = 2w(M)$.

Now notice that since $I_w(G) = w(S) + w(M)$ is minimum, then $2I_w(G) = 2w(S) + 2w(M) = w(S^2) + w(M^2)$ is also minimum and hence $I_w(G^2) = w(S^2) + w(M^2) = 2I_w(G)$. ■

This leads to an easy generalization.

Corollary 3.13. *If G is a graph of order n with normalized weighting w, then $I_w(G^k) = kI_w(G)$.*

Proof. Same as in the $k = 2$ case, but replace everything with k and it holds. ■

In Proposition 3.12, notice that the statement does not hold if the newly duplicated vertices are not connected, as it could be the case that a single vertex is the component of largest remaining weight and then in the duplicated version the component of largest remaining weight would still have the same weight instead of having double the weight of the original largest weighted remaining component. Thus in this case, if we denote the 2-duplicative graph without the new vertices connected as $G^{2'}$, we can only conclude that $I_w(G^{2'}) \leq 2I_w(G)$.

As mentioned before, finding the weighted integrity in general is NP-complete. However, in 2006, Ray et al. introduced a polynomial-time algorithm for interval graphs [13]. We investigate the specific case of paths here. Define the parameter $D_m^w(G)$ as the minimum weight that must be removed so that each component of G has weight at most m.

Then the weighted integrity equals

$$I_w(G) = \min_{m \in \mathbb{R}^+} \; m + D_m^w(G).$$

Lemma 3.14. *For a weighted path P there is a polynomial-time algorithm to compute $D_m(P)$ for any value of m.*

Proof. We use dynamic programming. Number the vertices of the path from 1 to n, and let w_i denote the weight of vertex i. Let $P[i, j]$ denote the subpath consisting of vertices i through j. For $1 \leq j \leq n$, let $D(m, j) := D_m(P[j, n])$. We need to find $D(m, 1)$. Then if $P[j, n]$ has total weight at most m, one has $D(m, j) = 0$. Otherwise one has the recurrence

$$D(m, j) = \min_{k} w_{k-1} + D(m, k)$$

where the minimum is over all $k > j$ such that $P[j, k-2]$ has weight at most m. ■

Next, we claim that there is a polynomial number of possibilities for the weight of a maximum component in the weighted path. Since there are at most $n(n+1)/2$ possible components, namely the $P[i, j]$, we can conclude the following.

Proposition 3.15. *For a weighted path, there is a polynomial-time algorithm to compute the weighted integrity.*

Also, for a cycle, one can remove each vertex in turn and find the weighted integrity of the resultant path using the above algorithm.

Proposition 3.16. *For the weighted cycle there is a polynomial-time algorithm to compute the weighted integrity.*

On the other hand, Ray and Deogun showed that finding the weighted integrity of a tree is NP-complete [12].

While our focus is very much on the worst-case adversarial assignments of weights, the question of what happens with random weights might be of interest. For example, we implemented the path algorithm and executed it on paths with a random normalized weighting. (That is, one originally obtained from a Dirichlet distribution and then normalized). It is unclear what the asymptotics are for $I_w(P_n)$ with random w; the data suggests it is $O(\sqrt{n})$. In contrast, the ordinary integrity grows like $2\sqrt{n}$. A plot of this is shown in Figure 3.3.

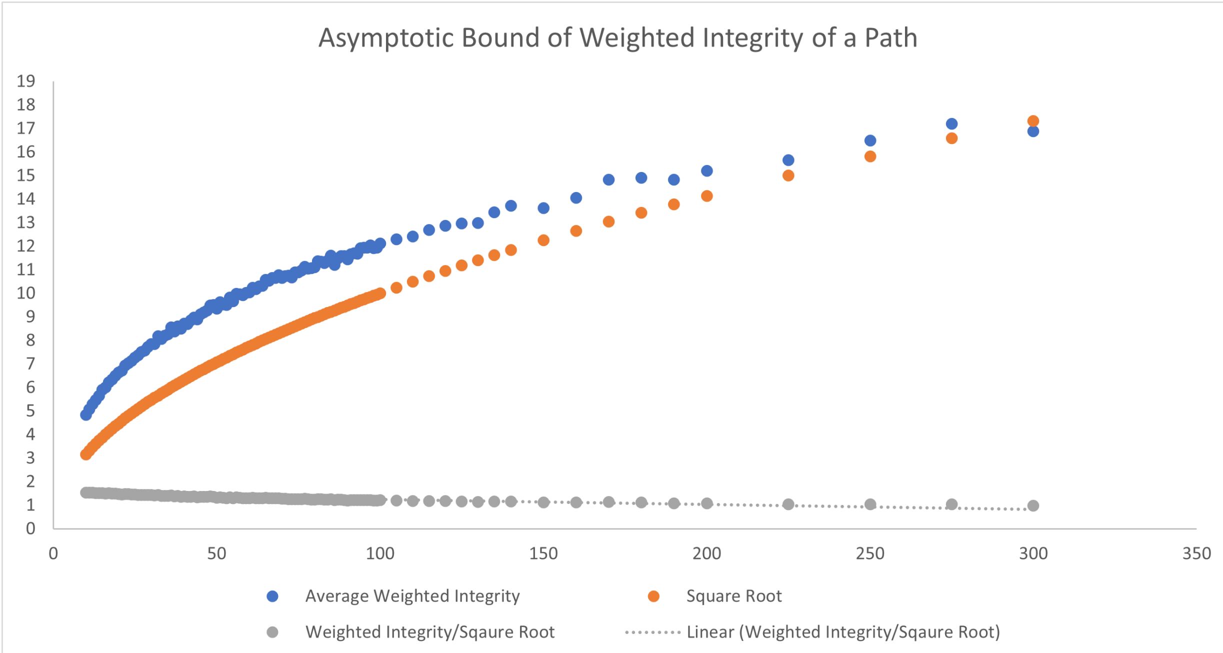


Figure 3.3: Asymptotic Bound of Weighted Integrity of a Path

Chapter 4

Fully Weighted Toughness

Recently, both Katona, Kovács, and Varga [9] and Shi and Wei [14] defined the **weighted toughness**. They introduced a nonnegative weighting w on the vertex set, and replaced $|S|$ by $w(S)$ which, as in Section 3, denotes the sum of the weights of the vertices in S. In this section, we propose an alternative way to generalize toughness to weighted graphs. In the spirit of measures of vulnerability, we consider not just the weights of the vertices removed but also the weights of those that remain. Our motivation is that, if one has for example the graph $2K_2$ where two of the vertices have weight 100 and two have weight 1, it seems reasonable to view the situation with the two vertices of weight 100 in separate components as more "disconnected" than the situation with the two vertices of weight 100 in the same component.

Therefore, it makse sense to replace the numerator of the original definition of toughness with $w(S)$ rather than $|S|$. Next, we turn our attention to the denominator.

For a graph H where the vertices have weighting w, define

> $k_w(H)$ *to be the sum, over the components of* H*, of the maximum weight of a vertex in each component.*

Note that if all weights are 1, this definition reverts to the original one of $k(H)$. This leads to the following definition.

Definition 4.1. The **fully weighted toughness** of a graph G with weighting w is defined to be

$$\tau_w(G) = \min \left\{ \frac{w(S)}{k_w(G-S)} \right\},$$

where the minimum is taken over all $S \subseteq V(G)$ such that S is a cut-set of G, or $G-S$ has one vertex. A set achieving the minimum is called a **tough set**.

For example, C_4 with an associated weighting w and its fully weighted toughness is shown in Figure 4.1.

Figure 4.1: The fully weighted toughness of C_4 and the weighting shown is $\tau_w(G) = \frac{3/2}{5/2} = \frac{3}{5}$

Note that due to the form of the tough ratio, one can scale all weights by the same positive factor and the fully weighted toughness does not change. Furthermore, one can restrict the sets S that need to be considered. As defined for example in [13], a cut-set is **strong** if $G-S'$ has fewer components than $G-S$ for all S' strictly contained in S. Also, a dominating vertex u is one such that every vertex of $V-u$ is adjacent to u.

Lemma 4.1. *In the definition of fully weighted toughness, for cut-sets S, one may restrict to those that are strong; and for S where $G - S$ is a single vertex v, one may restrict to v being a dominating vertex.*

Proof. If cut-set S has a subset S' where $G - S'$ has the same number of components as $G - S$, then $w(S') \leq w(S)$ and $k_w(G - S') \geq k_w(G - S)$. If $G - S$ is a single vertex v and there exists a vertex x not adjacent to v, then $S' = S - \{x\}$ is a cut-set with $w(S') \leq w(S)$ and $k_w(G - S') \geq k_w(G - S)$. In each case, the ratio for S' would be at most the ratio for S. ∎

It should be noted that allowing a set S whose removal leaves a single vertex affects the fully weighted toughness of more than just the complete graph. Consider for example the path P_3 on 3 vertices. If only cut-sets are allowed, one can place any positive weight on the middle vertex and zero on the end-vertices and get infinite fully weighted toughness. But if one allows the set S to be the two end-vertices, then it is straight-forward to show (see Proposition 4.6) that the fully weighted toughness of P_3 is at most 1, which coincides with the fact that the ordinary toughness of a bipartite graph is at most 1.

The fully weighted toughness problem is in general NP-hard, since having all weights 1 reduces to the ordinary toughness.

In the first section, we discuss first some formulas for calculating the fully weighted toughness of special classes of graphs and bounds on the fully weighted toughness in general. We then turn our focus to the maximization of the fully weighted toughness of graphs of specific classes of graphs and investigate when the maximum fully weighted toughness is equal to the ordinary toughness.

4.1 Some Bounds and Examples

We start with the calculation of the fully weighted toughness of complete multipartite graphs.

Lemma 4.2. *Let G be a complete r-partite graph with partite sets $X_1, \dots, X_r$. If w is a weighting with total weight W, then*

$$\tau_w(G) = \frac{W}{\max\limits_{1 \le i \le r} w(X_i)} - 1\,.$$

Proof. By Lemma 4.1, the only sets S that need to be considered are each the union of all but one partite set. If $S = V(G) - X_i$, then $w(S) = W - w(X_i)$ while $k_w(S) = w(X_i)$; thus $\tau_w(G) \le W/w(X_i) - 1$. This ratio is minimized when $w(X_i)$ is maximized. ■

Corollary 4.3. *For the complete graph K_n and weighting w, $\tau_w(K_n) = \dfrac{n - w_k}{w_k}$, where w_k is the maximum weight of any single vertex.*

Proof. Apply Lemma 4.2 and find a common denominator and we get the desired result. ■

Corollary 4.4. *For any graph G of order n with weighting w, $0 \le \tau_w(G) \le n - 1$.*

Proof. We know that for any graph G with n vertices, $\tau_w(G) \le t_w(K_n)$. Also, the maximum weighted vertex in any graph, $w_k \ge 1$. So then by Proposition 4.3 $\tau_w(G) \le \dfrac{n-1}{1} = n - 1$.

We also see that weighting any cut set with weights 0 leads to $t_w(G) = 0$. ■

We use averaging for several bounds. In particular, we need the elementary property of the mediant.

Lemma 4.5. *If a_i's are nonnegative reals and b_i positive reals, then the mediant $M = (\sum a_i)/(\sum b_i)$ of the fractions a_i/b_i is at least the minimum value of a_i/b_i and at most the maximum value of a_i/b_i. Furthermore, if the minimum and maximum fractions are not equal, then M lies strictly between the two.*

Our first bound involves the chromatic number.

Proposition 4.6. *If graph G has chromatic number r, then $\tau_w(G) \leq r - 1$ for any weighting w.*

Proof. Consider a weighting w of G with total weight W. Let $A_1, \ldots, A_r$ be a (proper) coloring of G. Consider removing the set $V(G) - A_i$ from the weighted G. The resultant graph has no edges, and hence $\tau_w(G) \leq (W - w(A_i))/w(A_i)$. Taking the mediant of these r ratios, we have

$$\tau_w(G) \leq \frac{\sum_{i=1}^{r} W - w(A_i)}{\sum_{i=1}^{r} w(A_i)} = \frac{rW - W}{W} = r - 1.$$

This gives the desired bound. ■

As noted in Lemma 4.2, the bound of Proposition 4.6 is achievable by any complete r-partite graph if one puts equal total weight on each partite set. In particular, this example shows that the fully weighted toughness of a graph is not bounded above by $n/\alpha - 1$ (where n denotes the order and α the independence number), as it is for ordinary toughness. For example, the complete graph minus an edge, $K_n - e$, can have fully weighted toughness $n - 2$.

For a tree T, Pippert [11] observed that the ordinary toughness is determined by the maximum degree, namely $\tau(T) = 1/\Delta(T)$. The same is not true for fully weighted toughness, but we observe next that one may at least assume that the set S is a singleton.

We will need the following concepts. Given a tree T and a set S of vertices, we define the S-**portion** of T to be the subgraph induced by the vertices of S and all vertices on paths between vertices of S. This is a subtree of T. Further, we define a vertex x of S to be S-**extremal**, or simply **extremal**, if it is an end-vertex in the S-portion; in other word, there exists a bridge that separates x from $S - \{x\}$.

Proposition 4.7. *The fully weighted toughness of a tree T that is not a star is achieved by removing a single vertex.*

Proof. Consider a tough set S of minimum cardinality. By Lemma 4.1, we may assume it is a strong cut-set; thus S does not contain any end-vertex. Suppose S is not a singleton.

Consider an S-extremal vertex x. Let A denote the set of vertices v of $V(T)-S$ such that in T every path from v to a vertex of $S-\{x\}$ goes through x. Let B denote the set of vertices v of $V(T) - S$ such that in T every path from v to x goes through at least one vertex of $S - \{x\}$. Let C be the remaining vertices in $V(T) - S$. Note that C might be empty, but if it is nonempty, then the subgraph induced by C is connected. Let $\alpha = m_w(A)$, $\beta = m_w(B)$, and $\gamma = m_w(C)$, where we set $\gamma = 0$ if C is empty. Figure 4.2 gives a depiction.

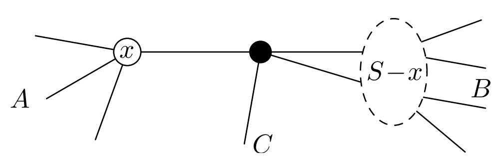


Figure 4.2: Splitting a tree

Now, if one removes vertex x from T, then the components of $T - x$ are those induced by A, and one component consisting of the vertices of B, C, and all of $S-\{x\}$. The maximum weight of a vertex in the latter component is at least γ. It follows that

$m_w(T-x) \geq \alpha+\gamma$. By a similar reasoning, it follows that $m_w(T-(S-\{x\})) \geq \beta+\gamma$. On the other hand, $m_w(T-S) = \alpha+\beta+\gamma$.

Thus, the fully weighted toughness of the tree T is bounded above by both

$$\tau_x \leq \frac{w(x)}{\alpha+\gamma} \quad \text{and} \quad \tau_{S-x} \leq \frac{w(S-\{x\})}{\beta+\gamma}.$$

By considering the mediant of τ_x and τ_{S-x}, it follows that the fully weighted toughness of the tree is at most

$$\tau_w(T) \leq \frac{w(x)+w(S-x)}{(\alpha+\gamma)+(\beta+\gamma)} = \frac{w(S)}{\alpha+\beta+2\gamma} \leq \frac{w(S)}{\alpha+\beta+\gamma} = \tau_w(T)\,.$$

By Lemma 4.5, both τ_x and τ_{S-x} equal $\tau_w(T)$, and hence both $\{x\}$ and $S-\{x\}$ contradict the choice of S. ■

Proposition 4.8. *The fully weighted toughness of a star is achieved by removing the center vertex or all the other vertices.*

Notice that propositions 4.7 and 4.8 give us the following directly:

Corollary 4.9. *There exists a polynomial time algorithm for finding the fully weighted toughness of a tree*

Given an unweighted graph, one can ask for the weights that minimize or maximize the fully weighted toughness. The former question is trivial. As mentioned in Corollary 4.4, one can simply put weight 0 on a cut-set, and then the fully weighted toughness is 0. The latter question is more interesting, and that is what we consider next.

4.2 Maximizing Fully Weighted Toughness

For a graph G, we define MFWT(G) as the maximum value of $\tau_w(G)$ over all weightings w of G. Since one can give every vertex weight 1, it follows that $\mathrm{MFWT}(G) \geq \tau(G)$. At the same time, if vertex v has maximum weight, one can choose $S = V(G) - \{v\}$, and so

$$\mathrm{MFWT}(G) \leq (n-1)w(v)/w(v) = n-1$$

, where $n = |V(G)|$.

It is immediate from Lemma 4.2 that if G is a complete r-partite graph, then $\mathrm{MFWT}(G) = r - 1$, attained by making the weight on each partite set equal. There are also graphs where the maximum fully weighted toughness is arbitrarily small. Consider the octopus graph O_s defined by starting with the star with s leaves and subdividing each edge exactly once.

Lemma 4.10. *For $s \geq 1$, $\mathrm{MFWT}(O_s) = 1/\sqrt{s}$.*

Proof. Since O_1 is a complete bipartite graph, by the above comment, $\mathrm{MFWT}(O_1) = 1$. So consider the case $s \geq 2$. Since O_s is not a star, it follows from Proposition 4.7 that the only S one need consider are the singletons $\{v\}$ where v is not an end-vertex. Thus the weights of the end-vertices appear only in the denominator of ratios, and since we are interested in the maximum value of the ratios, these weights may be assumed to be 0.

Let $w_1, \ldots, w_s$ denote the weights of the degree-2 vertices and y the weight of the center. Then we have $\tau_w(O_s) \leq w_i/y$ for each i (since the maximum component weight is at least y), and $\tau_w(O_s) \leq y/\sum w_i$. It can easily be checked that the biggest value of the minimum of these ratios is achieved when all the w_i are equal, and $y = \sqrt{s}w_i$. That is, $\mathrm{MFWT}(O_s) \leq 1/\sqrt{s}$. But the stated weights also show equality. ■

Figure 4.3 shows an optimal weighting of O_5.

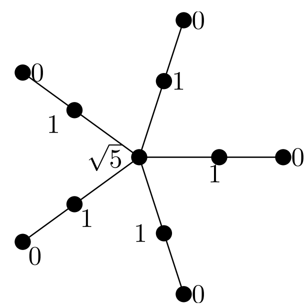


Figure 4.3: An optimal weighting of the octopus graph O_5 that maximizes $\tau_w(O_5)$.

We continue with an example that further demonstrates the contrast with ordinary toughness. We will need the following lemma.

Lemma 4.11. *If vertices u and v of a graph G are adjacent and have the same neighbors, then in computing* $\mathrm{MFWT}(G)$*, one may assume the two vertices get the same weight.*

Proof. Suppose w is a weighting with $w(u) > w(v)$. Let weighting w' be obtained from w by increasing the weight of v to equal that of u. We claim that $\tau_{w'}(G) \geq \tau_w(G)$. Consider first a strong cut-set S. Then either both u and v are in S or neither is in S. If neither is in S, then u and v are in the same component of $G - S$, and so $k_w(G - S) = k_{w'}(G - S)$ and $\tau_{w'}(G) = \tau_w(G)$. If both vertices are in S, then $w'(S) > w(S)$ and $\tau_{w'}(G) > \tau_w(G)$. Consider second a set where $G - S$ is a single vertex. The smallest ratio for such S is achieved where $G - S$ is a vertex of maximum weight, so the largest tough ratio for such a set is the same for w' as it is for w. This proves the claim. ■

Lemma 4.12. *Let G be the graph formed by the join of clique K_a with disjoint cliques $K_{b_1} \cup \ldots \cup K_{b_s}$ with $s \geq 2$ and $b = \max(b_1, \ldots, b_s)$. Then*

$$\mathrm{MFWT}(G) = \frac{2ab}{\sqrt{(a-1)^2 + 4ab} - (a-1)}.$$

Proof. Assume that $b_1 = b$. By the Lemma 4.12, we may assume that for each constituent clique, every vertex in that clique has the same weight. By scaling, we may assume that each vertex in K_a has weight 1, and each vertex in K_{b_i} has weight w_i.

The fully weighted toughness is achieved either by removing all of K_a, or by removing all vertices of G except one vertex of K_a. Thus, the fully weighted toughness is given by

$$\min \left\{ \frac{a}{w_1 + \cdots + w_s},\ a - 1 + \sum_{i=1}^{s} b_i w_i \right\}.$$

For a fixed value of $\sum w_i$, the second quantity is maximized at $w_2 = \cdots = w_s = 0$. (That is, we may put all the weight on the biggest clique.) Then the first quantity is decreasing in w_1 while the second is increasing in w_1. Hence the minimum is maximized where the two quantities are equal. So we have

$$a/w_1 = a - 1 + bw_1.$$

In this equation, we can solve for w_1, and deduce that the maximum ratio is the stated bound. ∎

For example, if $a = 1$, then the value in Lemma 4.12 simplifies to $\sqrt{b}$, while the ordinary toughness is $1/s$. Figure 4.4 shows an example weighting for the join of K_1 with cliques $K_2 \cup K_3$.

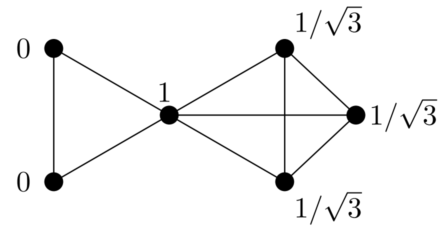


Figure 4.4: The join of K_1 with $K_2 \cup K_3$ and an optimal weighting.

Now, if we fix a, by calculus the expression in Lemma 4.12 is increasing in b. So for fixed a and the order of the graph, the bound is maximized by making b as large as possible; that is, by taking $s = 2$ and $b_2 = 1$ so that $b = n - a - 1$. So, not unexpectedly, the maximum fully weighted toughness of a graph with given connectivity a is achieved by the graph obtained by taking the clique K_{n-1} and adding one new vertex and joining it to a vertices of the clique. Thus we obtain the following.

Proposition 4.13. *If G is a graph of order n and connectivity a, then*

$$\tau_w(G) \leq \frac{2a(n-a-1)}{\sqrt{(a-1)^2 + 4a(n-a-1)} - (a-1)},$$

and this bound is sharp.

Since a graph with connectivity 1 is not Hamiltonian, it follows that no value of the fully weighted toughness is sufficient to guarantee Hamiltonicity. Even if one imposes a large value of connectivity, there is no threshold. For, graphs in the above family need not be 1-tough even for given connectivity, but can have arbitrarily large fully weighted toughness.

There are other simple cases where $\mathrm{MFWT}(G) > \tau(G)$.

Lemma 4.14. *If nontrivial graph G is connected and has a unique tough set for ordinary toughness, then* $\mathrm{MFWT}(G) > \tau(G)$.

Proof. Assume S is the ordinary tough set. Then for some $\varepsilon > 0$, define a weighting w by giving the vertices in S weight $1 + \varepsilon/|S|$ and the other vertices weight 1. Since the total increase in weight is ε, if X is any other candidate set, then $k_w(G - X) \leq k(G-X)+\varepsilon$. Note that if a, b, c, d are positive integers and $a/b < c/d$, then $(a+\varepsilon)/b < c/(d+\varepsilon)$ for all ε sufficiently small. Thus $(|S| + \varepsilon)/k(G - S) < |X|/(k(G - X) + \varepsilon)$, provided ε is small enough. That is, S remains the unique tough set, and $\tau_w(G) > \tau(G)$. ■

4.2.1 Graphs That Cannot be Toughened

We show next that for certain graphs their fully weighted toughness is at most their ordinary toughness. One simple example is the 4-cycle. Let $x_1, \ldots, x_4$ be the vertices with weights $w_1, \ldots, w_4$. Then $\{x_1, x_3\}$ is a cut-set where x_2 and x_4 are in different components, and vice versa. It follows that the fully weighted toughness is always at most the minimum of $(w_1 + w_3)/(w_2 + w_4)$ and $(w_2 + w_4)/(w_1 + w_3)$, which is at most 1, the ordinary toughness.

One can extend this idea to other symmetric graphs. Fix some cut-set S of the graph G. Say $G - S$ has r components with vertex sets $S^1, \ldots, S^r$ of orders $x_1, \ldots, x_r$. If Y is a cut-set such that $G - Y$ is isomorphic to $G - S$, then one can number the components $Y^1, \ldots, Y^r$ of $G - Y$ such that Y^j is isomorphic to S^j for $1 \leq j \leq r$. An S-family is a collection $Y_1, \ldots, Y_\ell$ of cut-sets such that $G - Y_i$ is isomorphic to $G - S$ for each i. We say that an S-family **rotates** if

(i) each vertex of G is in an equal number of the Y_i (necessarily $\ell|S|/n$ times)

(ii) for each j, the collection $Y_1^j, \ldots, Y_\ell^j$ contains every vertex of G the same number of times (necessarily $\ell x_j/n$ times).

We say that a graph is **rotatable** if every cut-set has a rotating family.

For example, it is easy to see that cycles are rotatable. As a specific instance, the 5-cycle has up to symmetry only one strong cut-set S, namely two nonadjacent vertices, and its removal yields two components. Say the vertices are a, b, c, d, e in order. Here is the S-family that rotates:

Y_i	Y_i^1	Y_i^2
a, c	b	d, e
b, d	c	e, a
c, e	d	a, b
d, a	e	b, c
e, b	a	c, d

Another rotatable graph is the balanced complete bipartite graph: if S is the cut-set consisting of one partite set, then a rotating S-family is given by it and the other partite set. Another example is the rooks graph, defined as the Cartesian product of two complete graphs.

Proposition 4.15. *If graph G is rotatable, then* $\mathrm{MFWT}(G) = \tau(G)$.

Proof. Fix a weighting w of G, and consider a tough set S for ordinary toughness. By assumption, there is a rotating S-family $Y_1, \ldots, Y_\ell$. For each Y_i, the fully weighted toughness of G is at most $w(Y_i)/k_w(G - Y_i)$. By replacing the maximum in each component by the average in each component, we have that

$$k_w(G - Y_i) \geq \sum_{j=1}^{r} \frac{w(Y_i^j)}{x_j}\,.$$

Let M be the mediant of the ratios $w(Y_i)/k_w(G - Y_i)$ for $1 \leq i \leq \ell$. As the fully weighted toughness of G is at most each of these ratios, it follows that

$$\tau_w(G) \leq M = \frac{\sum_{i=1}^{\ell} w(Y_i)}{\sum_{i=1}^{\ell} k_w(G - Y_i)} \leq \frac{\sum_{i=1}^{\ell} w(Y_i)}{\sum_{i=1}^{\ell} \sum_{j=1}^{r} w(Y_i^j)/x_j}.$$

Since each vertex appears in $\ell|S|/n$ of the Y_i, the numerator simplifies to $\ell|S|W/n$, where W is the total weight on the vertices of G. Similarly, the denominator simplifies as

$$\sum_{i=1}^{\ell} \sum_{j=1}^{r} \frac{w(Y_i^j)}{x_j} = \sum_{j=1}^{r} \sum_{i=1}^{\ell} \frac{w(Y_i^j)}{x_j} = \sum_{j=1}^{r} \frac{\ell x_j W/n}{x_j} = \frac{r\ell W}{n}.$$

Thus

$$\tau_w(G) \leq M \leq \frac{\ell|S|W/n}{r\ell W/n} = \frac{|S|}{r} = \frac{|S|}{k(G-S)} = \tau(G),$$

as required. ■

4.2.2 Paths

In this section we calculate the maximum fully weighted toughness of the path. The ordinary toughness of a path is $\frac{1}{2}$.

Proposition 4.16. *For $n \geq 4$, the value* $\mathrm{MFWT}(P_n)$ *is the (positive) root of the equation $\alpha^2 + \alpha^3 + \cdots + \alpha^{(n-1)/2} = \frac{1}{2}$ if n is odd, and of $\alpha + \alpha^2 + \cdots + \alpha^{(n-2)/2} = 1$ if n is even.*

In particular, the maximum fully weighted toughness of such a path lies strictly between $\frac{1}{2}$ and 1 For example, $\mathrm{MFWT}(P_5) = 1/\sqrt{2}$ and $\mathrm{MFWT}(P_6) = (\sqrt{5}-1)/2$. An optimal weighting of each is shown in Figure 4.5.

Proof. Let w be a weighting. Say the vertices of the path are $v_1, \ldots, v_n$ from left to right with weights $w_1, \ldots, w_n$. By Proposition 4.7, we know that the toughness ratio τ_w is achieved by removing exactly one of the non-end vertices. Let τ_i denote the

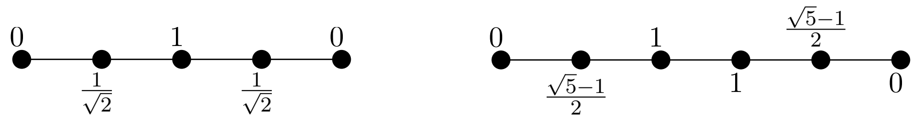


Figure 4.5: Optimal weightings of short paths

ratio achieved by the removal of vertex v_i for $2 \leq i \leq n-1$. Our goal is an upper bound for the minimum of these $n-2$ ratios.

Then $\tau_i = w_i/(x+y)$, where x is the maximum weight to the left of vertex v_i and y is the maximum weight to the right of vertex v_i. This quantity is at most $\frac{1}{2}$ if both x and y are at least w_i. It follows that the fully weighted toughness of a path is at most $\frac{1}{2}$, unless the weight distribution has no local minimum: that is, the weights are strictly increasing up to the maximum weight, which occurs either once or on two consecutive vertices, and then the weights are strictly decreasing from there on.

Since one can scale all weights and not change the fully weighted toughness, we may assume that the maximum weight is 1. Assume that w_ℓ is the leftmost occurrence of the maximum weight and w_r is the rightmost occurrence. As noted above, either $r = \ell$ or $r = \ell+1$. The weight w_1 only appears in the ratio τ_2 and in the denominator. Thus we may assume w_1 is as small as possible, namely 0. Similarly we may assume $w_n = 0$.

Claim 1. *We may assume* $\tau_2 = \tau_3 = \cdots = \tau_{\ell-1} = w_2$*, and so*

$$w_i = w_2^{i-1} + w_2^{i-2} + \cdots + w_2$$

for all $3 \leq i < \ell$*.*

Proof. Assume $\ell > 3$ and consider the weight w_2. It appears in only two of the ratios, namely $\tau_2 = w_2/1$ and $\tau_3 = w_3/(1+w_2)$. Then if $\tau_2 \neq \tau_3$, one can adjust

w_2 until the ratios are equal; note that this increases the smaller of the two ratios, and thus cannot decrease τ_w. That is, we may assume that $\tau_2 = \tau_3$, and thus that the equation $w_2(w_2+1) = w_3$ holds. Note that this adjustment preserves $w_2 < w_3$.

Similarly, assume $\ell > 4$ and consider the weight w_3. It appears only in the ratio τ_3 in the numerator, and in the ratio τ_4 in the denominator. If $\tau_3 \neq \tau_4$, we can adjust w_3 and w_2 simultaneously, maintaining the equality $w_2(w_2+1) = w_3$, until $\tau_3 = \tau_4$. Again, this increases the smaller of the ratios, and does not affect the monotonicity of the weights. Thus we may assume that $w_4/(w_3+1) = w_3/(w_2+1) = w_2$. In particular, the equation $w_4 = w_2^3 + w_2^2 + w_2$ holds. The remainder of the claim is proved similarly. $\triangleleft$

By a similar argument, we may assume $\tau_{r+1} = \tau_{r+2} = \ldots = \tau_{n-1} = w_{n-1}$, and that

$$w_{n+1-i} = w_{n-1}^{i-1} + w_{n-1}^{i-2} + \ldots + w_{n-1}$$

for all $3 \leq i < n - r$.

Claim 2. *We may assume that*

$$w_2 = w_{n-1}.$$

Proof. By the above discussion, $\tau_w = \min\{w_2, w_{n-1}, \tau_\ell, \tau_r\}$. Note that if $\ell = r$, then $\tau_\ell = 1/(w_{\ell-1} + w_{\ell+1})$; otherwise $\tau_\ell = 1/(w_{\ell-1}+1)$ and $\tau_r = 1/(1+w_{r+1})$. Suppose $w_2 \neq w_{n-1}$; say $w_2 > w_{n-1}$. Then one can adjust w_2 down, which decreases $w_{\ell-1}$, and this change can only increase τ_ℓ (while preserving τ_r if $r > \ell$) and so does not decrease τ_w. (And note that this cannot affect the monotonicity of the weights.) Thus the claim follows. $\triangleleft$

Let $\alpha = w_2 = w_{n-1}$. Let $A(k) = \sum_{j=1}^{k} \alpha^j$. Then $w_i = A(i-1)$ for $2 \leq i \leq \ell - 1$ and $w_i = A(n-i)$ for $r+1 \leq i \leq n-1$. Now, assume w is the optimal weighting;

that is, the one where $\tau_w(P_n)$ is maximized.

Assume first that this weighting has $r = \ell$. Out of all weightings achieving the optimum, choose one where ℓ is as close to the middle as possible. Suppose that $\ell \neq (n+1)/2$. Without loss of generality, we may assume $\ell > (n+1)/2$. Then by the above formulas, we have $w_{\ell-2} \geq A(n/2-2)$ while $w_{\ell+1} \leq A(n/2-2)$. Thus $w_{\ell-2} \geq w_{\ell+1}$.

We claim that one can swap the weights $w_{\ell-1}$ and 1 without decreasing τ_w. For, this swap affects only the two ratios $\tau_{\ell-1}$ and τ_ℓ, which change from $w_{\ell-1}/(1+w_{\ell-2})$ and $1/(w_{\ell-1}+w_{\ell+1})$ to $\tau'_{\ell-1} = 1/(w_{\ell-2}+w_{\ell-1})$ and $\tau'_\ell = w_{\ell-1}/(1+w_{\ell+1})$, respectfully. Then $\tau'_{\ell-1} > \tau_{\ell-1}$ since $w_{\ell-1} < 1$, while $\tau'_\ell \geq \tau_{\ell-1}$ since $w_{\ell-2} \geq w_{\ell+1}$. Thus the new τ'_w is at least the old τ_w, which contradicts the choice of w. That is, we may assume that $\ell = (n+1)/2$. In particular, it follows that n is odd, and that the fully weighted toughness of P_n in this case is the solution to

$$\alpha = \frac{1}{w_{\ell-1}+w_{\ell+1}} = \frac{1}{2A(\frac{n-3}{2})};$$

that is, the solution to $A(\frac{n-1}{2}) - \alpha = \frac{1}{2}$.

Assume second that the optimal weighting is only achieved when $r = \ell + 1$. Suppose that $\ell \neq n/2$. Without loss of generality we may assume $\ell > n/2$. By the above formulas, we have $w_{\ell-1} > w_{r+1}$.

We claim one can decrease the weight w_r from its 1 without decreasing τ_w. For, this change affects only the two ratios τ_ℓ and τ_r. The former will increase. For the latter, we note that for $\varepsilon > 0$ sufficiently small, $(1-\varepsilon)/(1+w_{r+1}) > 1/(1+w_{\ell-1})$ since $w_{\ell-1} > w_{r+1}$. That is, the new τ'_w is at least the old τ_w if ε is small enough. The new weighting has a unique maximum weight, which contradicts the choice of w. Thus we may assume that $\ell = n/2$. In particular, it follows that n is even, and that the fully weighted toughness of P_n in this case is the solution to

$$\alpha = \frac{1}{1 + w_{\ell-1}} = \frac{1}{1 + A(\frac{n-4}{2})};$$

that is, the solution to $A(\frac{n-2}{2}) = 1$.

The upper bound on $\mathrm{MFWT}(P_n)$ follows. It can be achieved by assigning weights according to the formulas. ∎

4.2.3 Trees

We show here that if a tree has at least three vertices, then there is a weighting such that the fully weighted toughness exceeds the ordinary toughness. Recall that the ordinary toughness of a tree is $1/\Delta$ where Δ is the maximum degree.

Proposition 4.17. *If T is a tree of order at least 3, then* $\mathrm{MFWT}(T) > \tau(T)$.

Proof. We already know this fact for the star. (By Lemma 4.2, the maximum is 1, but the ordinary toughness is $1/(n-1)$.) So assume the tree is not a star. By Proposition 4.7, we need only consider singleton cut-sets. We will construct a weighting w such that $w(v)/m_w(T - v) > 1/\Delta$ for each cut-vertex v.

We define a partition $E_0, E_1, E_2, \ldots$ of $V(T)$ as follows. Let M be the set of vertices of maximum degree. Define $E_0 = V(T) - M$. Define E_1 to be the extremal vertices of M. Define E_2 to be the extremal vertices of $M - E_1$, and so on. It follows that for $i \geq 1$, for each vertex $v \in E_i$ there is one component of $T - v$ that contains all of $(E_i - v) \cup E_{i+1} \cup E_{i+2} \cup \cdots$. Figure 4.6 gives an example, where the numbers inside each vertex give the index i of the E_i to which that vertex belongs.

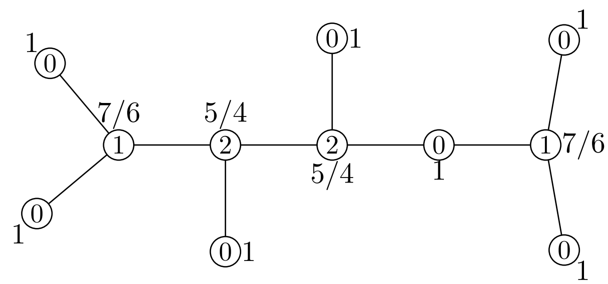


Figure 4.6: A weighted tree.

Now, let $\varepsilon = 1/\Delta$. For each $i \geq 0$, assign each vertex of E_i the weight

$$\mu_i = 1 + \varepsilon - \frac{\varepsilon}{2^i}\,.$$

Note that every vertex receives a weight that is at least 1 but less than $1+\varepsilon$. An example weighting is given above.

Let τ_i denote the minimum of the ratio $w(v_i)/k_w(T-v_i)$ over all $v_i \in E_i$. The numerator is always μ_i. For E_0, we have $k_w(T-v_0) < (\Delta-1)(1+\varepsilon)$, since v_0 has degree at most $\Delta - 1$. Thus

$$\tau_0 > \frac{1}{(\Delta-1)(1+\varepsilon)} = \frac{\Delta}{\Delta^2-1} > \frac{1}{\Delta}.$$

For $i \geq 1$, for E_i we have $k_w(T-v_i) < (\Delta-1)\mu_{i-1}+(1+\varepsilon)$, since v_i is extremal with respect to $M - E_{i-1}$, and so all components except one have maximum vertex weight at most μ_{i-1}. By the definition of the μ_i, it holds that $2\mu_i - \mu_{i-1} = 1+\varepsilon$. Thus

$$\tau_i > \frac{\mu_i}{(\Delta-1)\mu_{i-1}+(1+\varepsilon)} = \frac{\mu_i}{(\Delta-2)\mu_{i-1}+2\mu_i} \geq \frac{\mu_i}{\Delta\mu_i} = \frac{1}{\Delta}\,.$$

That is, we have shown that $w(v)/k_w(T-v) > 1/\Delta$ for all choices of cut-vertex v. ■

Chapter 5

Related Ideas and Conclusion

5.1 Weighted Connectivity

Another parameter that measures vulnerability is of course the connectivity. The connectivity of a graph gives a natural and overall insight about the vulnerability of a graph. One can ask similar questions as before.

Definition 5.1. Let G be a connected weighted graph with order n. The **weighted connectivity** of G, denoted $T_w(G)$, is the minimum weight required to remove to disconnect the graph or reduce it to a single vertex.

As with our discussion of weighted integrity, we consider normalized weightings. We find that calculating the weighted connectivity of a graph G may be quite difficult at times. However, given the ability to choose a weighting of vertices, the value can be always reduced to 0 as with the fully weighted toughness.

Proposition 5.1. *Let G be a connected weighted graph with order n. Then, $T_w(G) \geq 0$ and if $n > 1$ then there exists a normalized weighting of G such that $T_w(G) = 0$.*

Proof. Clearly $T_w(G) \geq 0$, since the weights of G are nonnegative. Also, given any graph G, we can find a minimal cut set of G and give each of those vertices weight 0 and assign weights to the remaining components in any way such that the total weight is n. The resulting weighted connectivity is $T_w(G) = 0$. ■

The more interesting question is when is the weighted connectivity maximized? Let $\mathscr{S}(G)$ denote the set of all minimal cut sets of G. The maximum weighted connectivity is the solution to the linear program

$$\begin{aligned} \max \quad & c \\ \text{s.t.} \quad & c \leq w(S) \text{ for all } S \in \mathscr{S}(G) \\ & w(V(G)) = n \\ & w_u \geq 0 \text{ for all } u \in V(G). \end{aligned}$$

Now we note that there exist graphs that have higher weighted connectivity than normal connectivity. We define MWC(G) of a graph G as the maximum weighted connectivity over all normalized weightings. An example is shown in Figure 5.1.

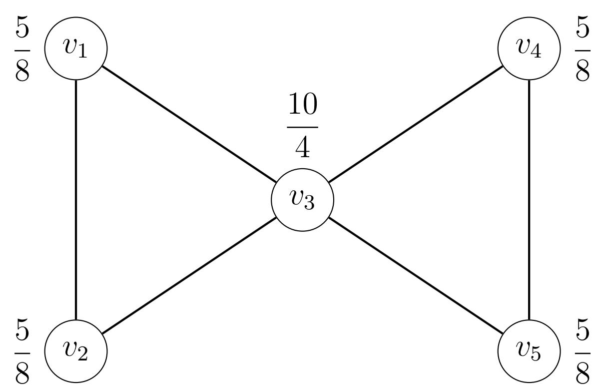


Figure 5.1: $MWC(G) = 2.5$, but G has connectivity 1.

Proposition 5.2. *For $n \geq 3$, $MWC(C_n) = 2$ and for all $n \neq 4$ the only weighting that achieves this maximum assigns all vertices with weight* 1.

Proof. Consider the cycle C_n. Every pair of nonadjacent vertices form a cut set. The weighted connectivity is thus at most the average weight of two nonadjacent vertices, which by symmetry, is at most 2.

To achieve equality, it is necessary that the average equals 2 and hence every minimal cut set has weight exactly 2. If $n \geq 7$, then for every pair of vertices u and v, there exists a vertex x such that $\{u, x\}$ and $\{v, x\}$ are cut sets; hence u and v must have the same weight. The argument for $n = 3, 5$ or 6 is similar. ∎

For $n = 4$, in the case of the cycle, the only cut sets are pairs of opposite vertices. It is sufficient that for every pair of opposite vertices, the weights sum to 2. Figure 5.2 illustrates two different normalized weightings w and w' of C_4 that both result in the weighted connectivity being 2.

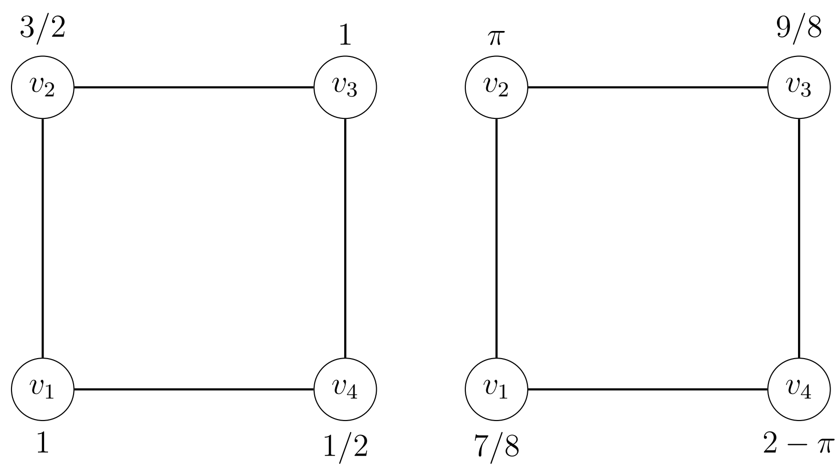


Figure 5.2: $T_w(C_4) = 2$ and $T_{w'}(C_4) = 2$.

5.2 Conclusion and Future Work

For further research, it would be interesting to determine what other classes of graphs have polynomial time algorithms for calculating the weighted integrity and fully weighted toughness, and which do not. Also, more calculations and bounds on both parameters for specific families of graphs would be helpful. We know little about when $\mathrm{MFWT}(G)$ is equal to the ordinary toughness and we know nothing about the complexity of calculating $\mathrm{MFWT}(G)$ at this time.

Bibliography

[1] K.S. Bagga, L.W. Beineke, Wayne Goddard, M.J. Lipman, and R.E. Pippert. A survey of integrity. *Discrete Applied Math.*, 37/38:13–28, 1990.

[2] C.A. Barefoot, R. Entringer, and H. Swart. Integrity of trees and powers of cycles. *Congr. Numer. 58*, pages 103–114, 1987.

[3] C.A. Barefoot, R. Entringer, and H.C. Swart. Vulnerability in graphs—a comparative survey. *J. Combin. Math. Combin. Computing.*, 1:13–22, 1987.

[4] Douglas Bauer, Haitze J. Broersma, and Edward Schmeichel. Toughness in graphs - a survey. *Graphs and combinatorics*, 22:1–35, April 2006.

[5] V. Chvátal. Tough graphs and hamilton circuits. *Discrete Math*, pages 215–228, 1973.

[6] W. Goddard and H.C. Swart. Integrity in graphs: Bounds and bascis. *J. Combin. Math. Combin. Computing. 7*, pages 139–151, 1990.

[7] Wayne Goddard. *On the Vulnerability of Graphs.* PhD dissertation, University of Natal, Durban, 1988.

[8] Wayne Goddard. The toughness of cubic graphs. *Graphs and Combin.*, 12:17–22, 1996.

[9] G.Y. Katona, I. Kovács, and K. Varga. The complexity of recognizing minimally tough graphs. *Discrete Appl. Math.*, 294:55–84, 2021.

[10] M.M. Matthews and D.P. Sumner. Hamiltonian results in $k_{1,3}$-free graphs. *J. Graph Theory*, 8:139–146, 1984.

[11] Raymond E. Pippert. On the toughness of a graph. In *Graph theory and applications (Proc. Conf., Western Michigan Univ., Kalamazoo, Mich., 1972)*, Lecture Notes in Math., Vol. 303, pages 225–233. Springer, Berlin, 1972.

[12] Sibabrata Ray and Jitender S. Deogun. Computational complexity of weighted integrity. *J. Combin. Math. Combin. Computing.*, 16:65–73, 1994.

[13] Sibabrata Ray, Rajgopal Kannan, Danyang Zhang, and Hong Jiang. The weighted integrity problem is polynomial for interval graphs. *Ars Combin.*, 79:77–95, 2006.

[14] M. Shi and Z. Wei. A polynomial algorithm for weighted toughness of interval graphs. *J. Math.*, 2021.