2011

Global Secure Sets Of Trees And Grid-like Graphs

Yiu Yu Ho
University of Central Florida

Global secure sets of trees and grid-like graphs

by

Yiu Yu Ho
B.S. University of Central Florida, 2006
M.S. University of Central Florida, 2010

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2011

Major Professor:
Ronald Dutton

Abstract

Let $G = (V, E)$ be a graph and let $S \subseteq V$ be a subset of vertices. The set S is a *defensive alliance* if for all $x \in S$, $|N[x] \cap S| \geq |N[x] - S|$. The concept of defensive alliances was introduced in [KHH04], primarily for the modeling of nations in times of war, where allied nations are in mutual agreement to join forces if any one of them is attacked. For a vertex x in a defensive alliance, the number of neighbors of x inside the alliance, plus the vertex x, is at least the number of neighbors of x outside the alliance. In a graph model, the vertices of a graph represent nations and the edges represent country boundaries. Thus, if the nation corresponding to a vertex x is attacked by its neighbors outside the alliance, the attack can be thwarted by x with the assistance of its neighbors in the alliance.

In a different subject matter, [FLG00] applies graph theory to model the world wide web, where vertices represent websites and edges represent links between websites. A *web community* is a subset of vertices of the web graph, such that every vertex in the community has at least as many neighbors in the set as it has outside. So, a web community C satisfies $\forall x \in C, |N[x] \cap C| > |N[x] - C|$. These sets are very similar to defensive alliances. They are known as *strong* defensive alliances in the literature of alliances in graphs. Other areas of application for alliances and related topics include classification, data clustering, ecology, business and social networks.

Consider the application of modeling nations in times of war introduced in the first paragraph. In a defensive alliance, any attack on a single member of the alliance can be successfully defended. However, as will be demonstrated in Chapter 1, a defensive alliance may not be able to properly defend itself when multiple members are under attack at the same time. The concept of *secure sets* is introduced in [BDH07] for exactly this purpose. The non-empty set S is a secure set if every subset $X \subseteq S$, with the assistance of vertices in S, can successfully defend against *simultaneous attacks* coming from vertices outside of S. The exact definition of simultaneous attacks and how such attacks may be defended will be provided in Chapter 1.

In [BDH07], the authors presented an interesting characterization for secure sets which resembles the definition of defensive alliances. A non-empty set S is a secure set if and only if $\forall X \subseteq S, |N[X] \cap S| \geq |N[X] - S|$ ([BDH07], Theorem 11). The cardinality of a minimum secure set is the security number of G, denoted $s(G)$. A secure set S is a *global secure set* if it further satisfies $N[S] = V$. The cardinality of a minimum global secure set of G is the global security number of G, denoted $\gamma_s(G)$.

In this work, we present results on secure sets and global secure sets. In particular, we treat the computational complexity of finding the security number of a graph, present algorithms and bounds for the global security numbers of trees, and present the exact values of the global security numbers of paths, cycles and their Cartesian products.

To my wife Yuan Li

Acknowledgments

I would like to thank Dr. Ronald Dutton and Dr. Robert Brigham for their time and wisdom over years of supervision. My adviser Dr. Ronald Dutton has helped me greatly during my Ph.D. study. He is a careful reader and a helpful editor. He showed me the way of doing basic research by leading as an example. Through his guidance and rigorous standards, I have grown into a stronger problem solver, a better writer and gained the ability to do basic research. Thank you Dr. Dutton!

I would like to thank Dr. Robert Brigham for his weekly meetings with us and for reviewing my writings. Dr. Brigham provided many helpful comments which improved this dissertation and my writing skill. I would like to thank Dr. Pawel Wocjan for teaching to me the discipline of Randomized and Approximation Algorithms. The techniques I learned in his classes helped my research directly. I would like to thank Dr. Sheau-Dong Lang for his friendship and encouragements during my stay at University of Central Florida.

Thank you for being in my committee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

$N_G(x)$	The open neighborhood of x in graph G: $\{y : xy \in E(G)\}$, page 1
$N_G[x]$	The closed neighborhood of x in graph G: $N_G(x) \cup \{x\}$, page 1
$\deg(x)$	The degree of vertex x in a graph: $\lvert N(x)\rvert$, page 1
$\delta(G)$	The minimum degree of graph G: $\min\limits_{v \in V(G)} \{\deg(v)\}$, page 1
$\Delta(G)$	The maximum degree of graph G: $\max\limits_{v \in V(G)} \{\deg(v)\}$, page 1
$N[S]$	The closed neighborhood of $S \subseteq V(G)$ in graph G: $\bigcup_{x \in S} N[x]$, page 1
$G[S]$	The subgraph of G induced by S: a graph $(S, E(G) \cap (S \times S))$, page 1
$a(G)$	The cardinality of a minimum defensive alliance of G, page 4
$s(G)$	The cardinality of a minimum secure set of G, page 6
$\gamma(G)$	The cardinality of a minimum dominating set of G, page 8
$\gamma_s(G)$	The cardinality of a minimum global secure set of G, page 8
$G \times H$	The Cartesian product of two graphs G and H, page 11
T_v	The subtree of T rooted at vertex v, page 51

p_v	The parent of vertex v in a tree, when v is not the root, page 51
c_v	The number of children of vertex v in a tree, page 51
S_v	The set $S \cap V(T_v)$, the vertices in S within subtree T_v., page 54
$G_1 \cup G_2$	The graph obtained by taking the disjoint union of graphs G_1 and G_2, page 122
$G + uv$	The graph obtained by adding the edge uv to graph G, for $u, v \in V(G)$, page 122
$G - v$	The graph obtained by removing vertex v from graph G, for $v \in V(G)$, page 122

CHAPTER 1

INTRODUCTION

Let $G = (V, E)$ be a graph. Unless otherwise stated, a graph is connected, undirected with no self loops or multiple edges.

For $x \in V, N_G(x) = \{y : xy \in E\}$ is the *open neighborhood* of x and $N_G[x] = N_G(x) \cup \{x\}$ is the *closed neighborhood* of x. The *degree* of x is $\deg_G(x) = |N_G(x)|$, the number of neighbors of x in G. The *minimum degree* of G is $\delta(G) = \min\limits_{v \in V}\{\deg_G(v)\}$ and the *maximum degree* of G is $\Delta(G) = \max\limits_{v \in V}\{\deg_G(v)\}$.

For $S \subseteq V$, $N_G(S) = \bigcup_{x \in S} N_G(x)$ and $N_G[S] = N_G(S) \cup S$ are the *open* and *closed neighborhoods* of S in G, respectively. We omit the subscript G when the graph under consideration is clear. The set $N[S] - S$ is the *boundary* of S.

The vertex x is *isolated* in G if $N_G(x) = \emptyset$. The subgraph of G *induced* by S is the graph denoted $G[S]$, where $G[S] = (S, E \cap (S \times S))$. So, $G[S]$ contains every vertex in S, and all edges between vertices in S whenever they exist in G. Notation that is not introduced explicitly follows [GY03] or [Wes04].

Notice $N(x)$ does not contain vertex x, but $N(S)$ may contain vertices in S. More specifically, $N(S)$ contains all vertices in S that are not isolated in $G[S]$.

A vertex subset of G is a global secure set if it is a secure set ([BDH07]) and a dominating set ([HHS98a]). This work studies the problem of finding minimum global secure sets of trees and grid-like graphs (see Definition 1.4.4).

The study of *secure sets* arises from the study of *defensive alliances*. Section 1.1 gives a short introduction on defensive alliances, then Section 1.2 discusses secure sets as a generalization of defensive alliances. Section 1.3 introduces the notion of global secure sets, and Section 1.4 discusses the global secure sets problem when restricted to trees and grid-like graphs.

1.1 Defensive alliances

Definition 1.1.1. A non-empty set $S \subseteq V$ is a *defensive alliance* if for all $x \in S$, $|N[x] \cap S| \geq |N[x] - S|$.

The concept of defensive alliances was introduced in [KHH04]. The primary motivation was for the modeling of nations in times of war, where allied nations are in mutual agreement to join forces if one of them is attacked. For a vertex x in a defensive alliance, the number of neighbors of x inside the alliance, plus the vertex x, is at least the number of neighbors of x outside the alliance. In a graph model, the vertices of a graph represent nations and the edges represent country boundaries. Every country in question has a corresponding vertex in the graph. For any two countries sharing boundaries, their corresponding vertices are adjacent in the graph. Thus, if the nation corresponding to a vertex x is attacked by its neighbors outside the defensive alliance, the attack can be thwarted by x with the assistance of its neighbors in the alliance.

The above model assumes that each country has equal military strength. While this model can be applied to some novel strategic games, it does not usually apply in the real world, since countries have different military power. The situation can be rectified by splitting each country into territories with similar military strength. Thus, a vertex in the graph represents one territory on the map, and two vertices are adjacent in the graph when the territories are close enough where one may send forces to another for reinforcements (in case of allies) or a direct assault (in case of opposing forces). Then, each country corresponds to the subgraph containing that country's territories. The countries in a mutual agreement may form an alliance, in which case the territories belong to those countries form a defensive alliance in the graph.

For a more formal graph theoretical setting, let $G = (V, E)$ be a graph and $S \subseteq V$ a defensive alliance of G, according to Definition 1.1.1. For vertex $x \in S$, one can think of the vertices in $N[x]-S$ as *attackers* of x and those in $N[x]\cap S$ as *defenders* of x. Then, any vertex $x \in S$ has as least as many defenders as attackers. When vertices in a defensive alliance have mutually agreed to help each other, any attack on a single vertex of the alliance can be defended. Figures 1.1 and 1.2 illustrate examples of defensive alliances. The cardinality of a smallest defensive alliance of G is the *defensive alliance number* of G, denoted $a(G)$. The study of defensive alliances have so far revolved around the study of minimum defensive alliances and their cardinality. A literature review on defensive alliances will be presented in Chapter 2.

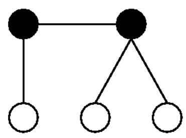

Figure 1.1: A defensive alliance is marked in black.

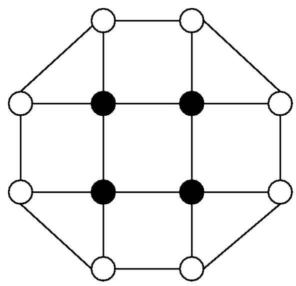

Figure 1.2: A defensive alliance is marked in black.

1.2 Secure sets

Consider the application of modeling nations (or military territories) in times of war discussed in the last section. In a defensive alliance, any attack on a single member of the alliance can be successfully defended. However, if multiple members are attacked by vertices (or forces) outside the alliance *simultaneously*, the alliance can be defeated. For example, consider Figure 1.2. If all eight of the vertices outside the alliance decide to attack the four vertices inside the alliance, this attack cannot be defended since the vertices inside the alliance are outnumbered. Thus, a defensive alliance may not be able to properly defend itself when multiple members are under attack at the same time. The concept of *secure sets* is introduced in [BDH07] for exactly this purpose. The set S is a secure set if every subset $X \subseteq S$, with the assistance of vertices in S, can defend against *simultaneous attacks* coming from vertices outside of S.

For $S \subseteq V$, the vertices in S are *defenders* and the vertices in $N[S] - S$ are *attackers*. Consider a situation where every attacker $y \in N[S] - S$ chooses one of its neighbor in $N(y) \cap S$ to *attack*. Given such a choice for all attackers, every defender $x \in S$ may choose a vertex in $N[x] \cap S$ to *defend*. An attacker can attack at most one vertex and a defender can defend at most one vertex. The attack is defended if, for every vertex $v \in S$, the number of vertices defending v is as many as the number of vertices attacking v. The set S is a *secure set* if every attack can be defended. The formal definition (based on [BDH07]) follows.

Definition 1.2.1.

1. Let $G = (V, E)$ be a graph. For any $S = \{s_1, s_2, \dots, s_k\} \subseteq V$, an *attack* on S is any k mutually disjoint sets $\mathscr{A} = \{A_1, A_2, \dots, A_k\}$ for which $A_i \subseteq N(s_i) - S$, $1 \leq i \leq k$. If $y \in A_i$, then y *attacks* s_i.

2. A *defense* of S is any k mutually disjoint sets $\mathscr{D} = \{D_1, D_2, \dots, D_k\}$ for which $D_i \subseteq N[s_i] \cap S$, $1 \leq i \leq k$. If $x \in D_i$, then x *defends* s_i. Note that this allows $s_i \in D_i$, in which case s_i defends itself.

3. An attack $\mathscr{A}$ is *defendable* if there exists a defense $\mathscr{D}$ such that $|D_i| \geq |A_i|$ for $1 \leq i \leq k$. Such a defense $\mathscr{D}$ is a *feasible* defense for the given attack $\mathscr{A}$.

4. A non-empty set S is a *secure set* if every attack on S is defendable.

Remark 1.2.2. Note that we may require $\mathscr{A}$ to be a partition of $N[S]-S$ and $\mathscr{D}$ a partition of S, and obtain an equivalent definition of secure sets.

Since the set V is a secure set, there is a smallest secure set of G. The cardinality of a minimum secure set of G is the *security number* of G, denoted $s(G)$. Figure 1.3 shows an example of a minimum secure set.

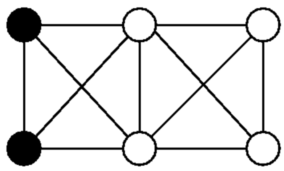

Figure 1.3: A minimum secure set is marked in black.

In [BDH07], the authors presented an interesting characterization for secure sets.

Theorem 1.2.3. ([BDH07], Theorem 11) A non-empty set $S \subseteq V$ is a secure set if and only if $\forall X \subseteq S, |N[X] \cap S| \geq |N[X] - S|$.

Theorem 1.2.3 is interesting and fundamental for two reasons.

1. The characterization simplifies the verification process of a secure set. Instead of finding feasible defenses for each possible attack (partition of $N[S] - S$), one can only consider subsets $X \subseteq S$ and examine its neighborhood. Although a naive verification using Theorem 1.2.3 still requires $\Omega(2^{|S|})$ operations, its complexity (in terms of both implementation and execution time) is nonetheless reduced.

2. The characterization gives a parallel to the definition of defensive alliances. Recall from Definition 1.1.1 that S is a defensive alliance if $\forall x \in S, |N[x] \cap S| \geq |N[x] - S|$. A secure set is a generalized form of defensive alliance in the sense that we require all subsets $X \subseteq S$ to satisfy $|N[X] \cap S| \geq |N[X] - S|$, in addition to single vertices.

The secure set characterization resembles a well known theorem about matching in bipartite graphs.

Theorem 1.2.4. (Hall's Matching Theorem) ([Hal35] or [Wes04] Theorem 3.1.11) An (A, B)-bipartite graph has a matching that saturates A if and only if $\forall X \subseteq A, |N(X)| \geq |X|$.

In Chapter 4, we will investigate the relationship between the secure set characterization and Hall's Matching Theorem. In particular, the chapter presents an alternative proof of Theorem 1.2.3 as an application of Hall's Theorem.

Notice every secure set is a defensive alliance, but the converse is not true. For example, the set in Figure 1.3 is a defensive alliance, but the sets in Figure 1.1 and 1.2 are not secure sets.

Observation 1.2.5. Every secure set is a defensive alliance. Consequently, $s(G) \geq a(G)$.

1.3 Global secure sets

Secure sets in graphs is a local property in the sense that the security of set S does not necessarily depend on the structure of the entire graph, but only the structure of the subgraph

$G[N[S]]$. In other words, one may verify the security of a set S without examining the entire graph the set belongs to. As a result, information about secure sets in a graph does not necessarily reflect properties of the graph, but rather properties of subregions inside the graph (certain subgraphs). To extend the security property to involve the entire graph, we may require additionally that $N[S] = V$. That is, requiring the set S to be a *dominating set*.

Definition 1.3.1. Let $G = (V, E)$ be a graph. A *dominating set* of G is a vertex subset $S \subseteq V$ such that for all $v \in V$, either $v \in S$ or there exists $u \in S$ with $uv \in E$.

Equivalently, S is a dominating set whenever $N[S] = V$. The cardinality of a minimum dominating set of G is the *domination number* of G, and is denoted by $\gamma(G)$. The theory of dominating sets has been studied extensively in literature. The reader may refer to [HHS98a] and [HHS98b] for a review on this subject. More recent developments on the subject of domination include [CJH04, Gra06, BGH08, FGK08, FGP08, RB08, DDH09, FGK09].

The set S is a *global secure set* if S is a dominating set and a secure set. Global secure sets have the security property, and their existence also involves the entire graph. Thus, global secure sets can be seen as a global property of the graph in question. The cardinality of a minimum global secure set of G is the *global security number* of G, denoted $\gamma_s(G)$. Notice $\gamma_s(G) \geq \max\{\gamma(G), s(G)\}$. Figure 1.4 shows an example of a minimum global secure set.

Similarly, the set S is a *global defensive alliance* if it is a dominating set and a defensive alliance. The cardinality of a minimum global defensive alliance of G is the *global defensive*

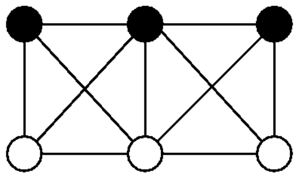

Figure 1.4: A minimum global secure set is marked in black.

alliance number of G, denoted $\gamma_a(G)$. Since global secure sets are also global defensive alliances, $\gamma_s(G) \geq \gamma_a(G) \geq \gamma(G)$.

Global secure sets studied in this work are different from *secure dominating sets* studied in [CFM03], [KM08] or [GM09]. In [CFM03], a secure dominating set refers to a dominating set $X \subseteq V$ such that $\forall y \notin X, \exists x \in (N(y) \cap X) : X - \{x\} \cup \{y\}$ is a dominating set. The cardinality of a minimum secure dominating set is sometimes also denoted by $\gamma_s(G)$ in the literature.

1.4 Trees and grid-like graphs

The study of secure sets revolves around the study of minimum secure sets and their cardinalities. Consider the following problems.

Problem 1.4.1. `Secure Set`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a secure set of cardinality k or less?

Problem 1.4.2. `Global Secure Set`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a global secure set of cardinality k or less?

There is no known polynomial algorithm for solving the above problems. In fact, given a vertex subset S in a graph, there is no known polynomial algorithm which can determine whether S is a secure set. This implies `Secure Set` and `Global Secure Set` may not be in the class NP, if P $\neq$ NP. Chapter 3 gives a throughout treatment on the complexity of finding minimum secure sets and minimum global secure sets.

Many problems in graph theory have no known polynomial solution for general graphs, but for which polynomial solutions exist when the graph under consideration is restricted to certain special classes. This is the case for (global) secure sets. `Secure Set` and `Global Secure Set` can be solved in polynomial time when the graph under consideration is restricted to trees and grid-like graphs. A central part of this work focuses on finding global security numbers of trees and grid-like graphs.

A *tree* is a connected acyclic graph. A tree with n vertices has exactly $n - 1$ edges and there exists exactly one path between any pair of distinct vertices. Chapters 5 and 6 investigate the global security numbers of trees and related problems.

Definition 1.4.3. The *Cartesian product* of two graphs G and H is a graph denoted $G \times H$, where $V(G \times H) = V(G) \times V(H)$ and $E(G \times H) = \{(v_i, u_i)(v_j, u_j) : (v_i = v_j \text{ and } u_iu_j \in E(H)) \text{ or } (v_iv_j \in E(G) \text{ and } u_i = u_j)\}$.

A *path* on $n \geq 2$ vertices, denoted P_n, is a graph with $V(P_n) = \{v_1, \ldots, v_n\}$ and $E(P_n) = \{v_i v_{i+1} : 1 \leq i \leq n-1\}$. A *cycle* on $n \geq 3$ vertices, denoted C_n, is a graph with $V(C_n) = \{v_1, \ldots, v_n\}$ and $E(C_n) = \{v_i v_{i+1} : 1 \leq i \leq n-1\} \cup \{v_1 v_n\}$. A *two-dimensional grid* $P_n \times P_m$ is the Cartesian product of two paths P_n and P_m. A *two-dimensional cylinder* $P_n \times C_m$ is the Cartesian product of a path P_n and a cycle C_m. A *two-dimensional torus* $C_n \times C_m$ is the Cartesian product of two cycles C_n and C_m.

Definition 1.4.4. The class of graphs which contains exactly P_n, C_n, $P_n \times P_m$, $P_n \times C_m$ and $C_n \times C_m$ is the class of *grid-like* graphs.

Note that the order of each path is at least two and the order of each cycle is at least three. For example, the smallest two-dimensional torus is $C_3 \times C_3$, which has 9 vertices. Chapters 7 and 8 investigate the global security numbers of grid-like graphs.

CHAPTER 2

LITERATURE REVIEW

Alliances appear in the real world when people, businesses, nations, etc. decide to unite for mutual support over a common interest. A defensive alliance can be used to model such a group. In [FLG00], a notion similar to that of a defensive alliance is used for modeling the world wide web. In this model, vertices of the graph represent websites and edges represent links between websites. A *web community* is a subset of vertices of the web graph, such that every vertex in the community has at least as many neighbors in the community as it has outside. So, a web community C satisfies $\forall x \in C, |N[x] \cap C| > |N[x] - C|$. These sets are very similar to defensive alliances. They are known as *strong* defensive alliances in the literature of alliances in graphs. In a strong defensive alliance, every vertex has at least as many neighbors inside the alliance as it has outside the alliance. Thus, if the edges of a graph represent similarity or closeness relationships between vertices (e.g., web graph, social network, etc.), a vertex inside a strong defensive alliance is considered to be in a close relationship with other vertices inside the strong alliance. This property can be used for different areas of classification, where groups of closely related entries form alliances.

In [Sha04], the above concept is applied to data clustering. The vertices of a graph are partitioned into alliances, where each alliance is treated as one cluster. [Sha04] presents algorithms for partitioning a graph into alliances and provides experimental results on its application to data clustering problems. Secure sets, being a more generalized version of alliances, can be similarly applied to the areas of collaborative agreement, classification and data clustering.

The rest of this chapter surveys existing literature for graph theoretical results related to defensive alliances, global defensive alliances and secure sets. Section 2.1 presents results on upper and lower bounds for each of these properties. Section 2.2 presents existing results on the computational complexity of these problems. Section 2.3 presents results on the exact values of the security numbers of some special classes of graphs.

2.1 Upper and lower bounds

Let K_n denote the complete graph on n vertices. Then, $s(K_n) = a(K_n) = \lceil n/2 \rceil$. So, the security number and the alliance number of a graph may be as large as $\lceil n/2 \rceil$. This observation may lead one to conjecture $\lceil n/2 \rceil$ as an upper bound for the security and alliance numbers of an arbitrary graph of order n. This conjecture is proven true for alliance numbers in [FLH03], and shown to be false for security numbers in [DLB08].

Theorem 2.1.1. ([FLH03], Theorem 2)

If G is a connected graph of order n, then $a(G) \leq \lceil n/2 \rceil$, and the bound is sharp.

Using the class of *Kneser graphs* as a counter-example, it was shown in [DLB08] that $\lceil n/2 \rceil$ is not a general upper bound for $s(G)$. Let m and k be positive integers such that $m \geq k$. The *Kneser graph* $K(m,k)$ is a graph whose vertex set contains $n = \binom{m}{k}$ vertices, each represents one of the k-element subsets of a m-element set. Two vertices in $K(m,k)$ are adjacent if and only if their corresponding subsets are disjoint. The security numbers of $K(m,2)$ are found to be as follows.

Theorem 2.1.2. ([DLB08], Theorem 25)

$$s(K(m,2)) = \begin{cases} 1 & \text{if } m \leq 4 \\ 5 & \text{if } m = 5 \\ \lceil (n+1)/2 \rceil & \text{if } m \geq 6 \end{cases}$$

Theorem 2.1.2 shows that if $m \geq 6$ and $n = \binom{m}{2}$ is even, then $s(K(m,2)) > \lceil n/2 \rceil$.

There is no known sharp upper bound for the security number of a graph in terms of only its order (Open Problem 9.2.1). In [DLB08] and [Dut09], bounds on $s(G)$ were given in terms of other invariants of G such as the minimum degree, degree sequence, girth, connectivity and domination number. We cite selected results below.

Theorem 2.1.3. ([DLB08], Theorem 3 and Theorem 12)

Let G be a graph with degree sequence $d_1 \leq d_2 \leq \cdots \leq d_n$. Then $\left\lceil (d_{s(G)}+1)/2 \right\rceil \leq s(G) \leq n - \lceil d_{k+1}/2 \rceil$, where $k = \max\{i : i \leq \lceil d_{i+1}/2 \rceil\}$.

Proposition 2.1.4. ([DLB08], Corollary 5 and Proposition 6)

Let G be a graph with minimum degree δ and let $\ell = \lceil (\delta + 1)/2 \rceil$. Then, $s(G) \geq \ell$, and equality holds if and only if there is a subset $S \subseteq V$ such that S induces a K_ℓ and $|N[S] - S| \leq \ell$.

Theorem 2.1.5. ([DLB08], Theorem 9)

Let G be a graph of order $n \geq 2$. If G has a secure set which is also a minimal dominating set, then $s(G) \leq \lfloor n/2 \rfloor$.

Theorem 2.1.6. ([DLB08], Theorem 10)

Let G be a graph of order n, girth $g \geq 5$ and minimum degree $\delta \geq 3$. Then $s(G) \leq n - 1 - \Delta \frac{(\delta-1)\lfloor (g-3)/2 \rfloor - 1}{\delta - 2}$.

Theorem 2.1.7. ([DLB08], Theorem 15)

Let G be a bipartite graph of order n and minimum degree δ. Then $s(G) \leq n - \delta$.

The *vertex connectivity* number of graph G, denoted $\kappa(G)$, is the cardinality of a smallest set of vertices whose removal disconnects G or reduces it to a single vertex. The *edge connectivity* number of graph G, denoted $\kappa_1(G)$, is the cardinality of a smallest set of edges whose removal disconnects G.

Theorem 2.1.8. ([DLB08], Theorem 16 and Theorem 17)

Let G be a graph of order n and minimum degree δ. If $\kappa(G) \leq \lceil \delta/2 \rceil$, then $s(G) \leq \lfloor (n - 2\lceil \delta/2 \rceil + \kappa(G))/2 \rfloor$. If $\kappa_1(G) < \delta$, then $s(G) \leq \lfloor n/2 \rfloor$.

Theorem 2.1.9. ([Dut09], Theorem 19)

Let $L(G)$ denote the line graph of G. For any positive integer m, $s(L(K_m)) \leq k(k-1)/2$, where $k = \lceil (2m+1)/3 \rceil$.

Proposition 2.1.10. ([Dut09], Corollary 15)

If G is a triangle-free graph, then $s(G) \geq 2\lceil (\delta(G)-1)/2 \rceil$.

Definition 2.1.11. A set $X \subseteq V$ is a *total dominating set* or an *open dominating set* if $N(X) = V$. The cardinality of a minimum total dominating set of G is the *total domination number* of G, denoted $\gamma_t(G)$.

In [HHH03], bounds on global defensive alliance numbers of various graphs are given. Selected results are cited below.

Lemma 2.1.12. ([HHH03], Lemma 4)

For any graph G with $\delta(G) \geq 2$, $\gamma_t(G) \leq \gamma_a(G)$. Furthermore, if $\Delta(G) \leq 3$, then $\gamma_t(G) = \gamma_a(G)$.

Theorem 2.1.13. ([HHH03], Theorem 11)

Let G be a graph of order n. Then, $\gamma_a(G) \geq (\sqrt{4n+1}-1)/2$, and this bound is sharp.

Proposition 2.1.14. ([HHH03], Corollary 13)

If G is a cubic graph or a 4-regular graph of order n, then $\gamma_a(G) \geq n/3$.

Theorem 2.1.15. ([HHH03], Theorem 15)

If G is a bipartite graph of order n and maximum degree Δ, then $\gamma_a(G) \geq (2n)/(\Delta+3)$, and this bound is sharp.

Theorem 2.1.16. ([HHH03], Theorem 17 and Theorem 21)

If T is a tree of order $n \geq 4$, then $(3n)/5 \geq \gamma_a(T) \geq (n+2)/4$, and both bounds are sharp.

In [BCH10], a more specialized lower bound than the one in Theorem 2.1.16 is given for the global defensive alliance numbers of trees.

Definition 2.1.17. A *leaf* of a tree is a vertex with degree one. A *support vertex* of a tree is a vertex adjacent to a leaf.

Theorem 2.1.18. ([BCH10], Theorem 7)

Let T be a tree of order $n \geq 2$ with ℓ leaves and s support vertices. Then, $\gamma_a(T) \geq (3n - \ell - s + 4)/8$, and the bound is sharp.

2.2 Complexity results

This section discusses results on the complexity of finding minimum (global) defensive alliances. The complexity of finding minimum (global) secure sets will be discussed in Chapter 3.

Consider the defensive alliance and global defensive alliance problems stated below.

Problem 2.2.1. `Defensive Alliance`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a defensive alliance of cardinality at most k?

Problem 2.2.2. `Global Defensive Alliance`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a global defensive alliance of cardinality k or less?

`Defensive Alliance` was shown to be NP-Complete in [JHM09], even when the input is restricted to split and chordal graphs. `Global Defensive Alliance` was shown to be NP-Complete in [CBD06], and was shown to remain NP-Complete when the input is restricted to planar graphs in [Enc09]. This gives rise to the hypothesis that finding minimum secure sets and minimum global secure sets (Problems 1.4.1 and 1.4.2) may be NP-Hard.

In related attempts, [FR07] and [Enc09] study the parameterized complexity of finding minimum defensive alliances and present parameterized algorithms for minimum defensive alliances and related problems. [ED08] presents a parameterized algorithm for finding minimum secure sets of graphs. [Dut06] presents exhaustive search algorithms and heuristics for finding $s(G)$. There is no known polynomial algorithm for Problem 1.4.1 or 1.4.2. Chapter 3 will give more detailed information regarding the complexity of secure sets.

2.3 Security number of certain classes of graphs

Many NP-Complete problems are known to have polynomial time solutions when the input graph is restricted to certain classes. This is the case for alliances and secure sets. Polynomial algorithms for finding the global defensive alliance number of a tree are presented in [Jam07] and [Enc09]. In addition, [Jam07] presents a polynomial algorithm for finding the defensive alliance number of a series parallel graph, and [Enc09] presents a polynomial algorithm for finding the global defensive alliance number of a graph with constant domino treewidth.

Chapter 6 will present a polynomial algorithm for finding the global security number of a tree. Chapters 7 and 8 will present the exact values of the global security numbers of grid-like graphs (Definition 1.4.4). There is no known polynomial algorithm for finding the security number or global security number of a series parallel graph (Open Problems 9.3.1 and 9.3.2).

The exact values of security numbers of some classes of graphs are known. Theorems 2.3.1, 2.3.2, 2.3.3 and 2.3.4 present the exact values of security numbers of maximum degree three graphs, outerplanar graphs, complete k-partite graphs and grid-like graphs, respectively.

Theorem 2.3.1. ([BDH07], Proposition 4, Part 5)

Let G be a graph with maximum degree three. If G is not a forest, let g be its girth. Define k to be g if G has at most one vertex of degree 2, and to be the number of vertices in a shortest path between degree 2 vertices otherwise. Then,

$$s(G) = \begin{cases} 1 & \text{if } \delta(G) \leq 1, \\ 2 & \text{if } k = 2 \text{ or } G \text{ contains either a } K_4 - e \\ & \text{or a } K_3 \text{ with a degree 2 vertex,} \\ \max\{3, \min\{k, g-1\}\} & \text{if } G \text{ contains an induced } K_{2,3} \\ & \text{or a } C_g \text{ with a degree 2 vertex,} \\ \min\{k, g\} & \text{otherwise.} \end{cases}$$

Theorem 2.3.2. ([Dut09], Theorem 21)

Let G be a connected outerplanar graph. Then,

$$s(G) = \begin{cases} 1 & \text{if } \delta(G) \leq 1 \\ 2 & \text{if } \delta(G) > 1 \text{ and there exists } xy \in E(G) \text{ such that } |N[x] \cup N[y]| \leq 4 \\ 3 & \text{otherwise} \end{cases}$$

Theorem 2.3.3. ([DLB08], Theorem 19)

Let $G = K_{n_1,n_2,\ldots,n_k}$ be a complete k-partite graph of order $n = \sum_{i=1}^{k} n_i$, where the i-th partite set contains exactly n_i vertices. If G is not a star or $\overline{K_n}$, then $s(G) = \lceil n/2 \rceil$, otherwise $s(G) = 1$.

Theorem 2.3.4. ([BDH07, KOY09])

1. $s(P_n) = 1$. ([BDH07], Proposition 4, Part 1(a))

2. $s(C_n) = 2$. ([BDH07], Proposition 4, Part 3)

3. $s(P_n \times P_m) = \min\{n, m, 3\}$. ([BDH07], Proposition 4, Part 4(a))

4. $s(P_n \times C_m) = \min\{2n, m, 6\}$. ([KOY09], Theorem 4.2)

5. $s(C_n \times C_m) = \left\{ \begin{array}{ll} 4 & \text{if } n = m = 3 \\ \min\{2n, 2m, 12\} & \text{if } \max\{n, m\} \geq 4 \end{array} \right\}$ ([KOY09], Theorem 3.9)

The rest of this section will cite results on finding minimum defensive alliances of grid-like graphs and partitioning a graph into (global) defensive alliances. The alliance numbers of grid-like graphs are found in [KHH04]. The problem of partitioning vertices of a grid graph into defensive alliances is studied in [HL07]. The problem of partitioning vertices of a tree into global defensive alliances is studied in [EG08]. Selected results are cited below.

Theorem 2.3.5. ([KHH04])

1. $a(P_n) = 1$. ([KHH04], Corollary 1)

2. $a(C_n) = 2$. ([KHH04], Corollary 2, (i))

3. $a(P_n \times P_m) = 2$. ([KHH04], Theorem 1, (ii))

4. $a(P_n \times C_m) = 2$. ([KHH04], Proposition 3, (i))

5. $a(C_n \times C_m) = girth(C_n \times C_m)$. ([KHH04], Theorem 2, (iv))

Let $G = (V, E)$ be a graph. The *alliance partition number* of G, denoted $\psi_a(G)$, is the maximum number of sets in a partition of V such that every set in the partition is a defensive alliance.

Theorem 2.3.6. ([HL07])

1. $\psi_a(P_n) = \lceil (n+1)/2 \rceil$. ([HL07], Theorem 2.1)

2. $\psi_a(P_2 \times P_m) = m$. ([HL07], Theorem 2.2)

3. $\psi_a(P_3 \times P_m) = \left\{ \begin{array}{ll} m & \text{if } m \text{ is odd} \\ & \\ m+1 & \text{if } m \text{ is even} \end{array} \right\}$ ([HL07], Theorem 2.3)

4. $\psi_a(P_n \times P_m) = \left\lfloor \dfrac{n-2}{2} \right\rfloor \left\lfloor \dfrac{m-2}{2} \right\rfloor + n + m - 2$ for $4 \leq n \leq m$. ([HL07], Theorem 2.5)

Let $G = (V, E)$ be a graph. The *global alliance partition number* of G, denoted $\psi_g(G)$, is the maximum number of sets in a partition of V such that every set in the partition is a global defensive alliance.

Theorem 2.3.7. ([EG08], Theorem 2.2)

Let G be a connected graph with minimum degree δ. Then, $1 \leq \psi_g(G) \leq 1 + \lceil \delta/2 \rceil$, and the bound is sharp.

Corollary 2.3.8. ([EG08], Corollary 3.1)

For any tree graph T, $1 \leq \psi_g(T) \leq 2$.

Theorem 2.3.9. ([EG08], Theorem 3.7)

Let T be a tree of order $n \geq 3$ with $\Delta(T) \leq 3$. Then, $\psi_g(T) = 2$ if and only if there exists a pair of leaves in T that are an odd distance from one another.

Other work on alliances can be found in [FFG04, RS06, CMR07, JD07, RA07, BDH09, FRS09, HWW09, RYS09, RS09, SR09].

CHAPTER 3

COMPLEXITY OF SECURE SETS

3.1 Introduction

This chapter discusses the complexity of finding minimum (global) secure sets of a graph and related problems. Recall from Definition 1.2.1 that for a graph $G = (V, E)$ and a vertex subset $S = \{s_1, s_2, \ldots, s_k\}$, an attack on S is a set of mutually disjoint subsets of $N[S] - S$, $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$ such that $A_i \subseteq N[s_i] - S$. An attack $\mathscr{A}$ is defendable if there exists a set of mutually disjoint subsets of S, $\mathscr{D} = \{D_1, D_2, \ldots, D_k\}$ such that $D_i \subseteq N[s_i] \cap S$ and $|D_i| \geq |A_i|$. The set $\mathscr{D}$ is a feasible defense for $\mathscr{A}$. A set S is a secure set if every attack on S is defendable. Set S is a global secure set if it is a secure set and a dominating set (Definition 1.3.1) of G. The security number of G is the cardinality of a minimum secure set in G, denoted $s(G)$. The global security number of G is the cardinality of a minimum global secure set of G, denoted $\gamma_s(G)$.

There are several problems of interest with regard to the properties of secure sets.

Problem 3.1.1. `Feasible Defense`

Given: A graph $G = (V, E)$, set $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$ and an attack $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$, such that $A_i \subseteq (N[s_i] - S)$ and $(A_i \cap A_j = \emptyset$ if $i \neq j)$.

Question: Is there a feasible defense for $\mathscr{A}$? That is, a set $\mathscr{D} = \{D_1, D_2, \ldots, D_k\}$ such that $D_i \subseteq (N[s_i] \cap S)$, $|D_i| \geq |A_i|$, and $(D_i \cap D_j = \emptyset$ if $i \neq j)$.

Section 3.2 presents a polynomial algorithm for finding a feasible defense (or determine there is none) for a given attack on a subset of vertices of a graph. This means `Feasible Defense` is in P.

Problem 3.1.2. `Is Secure`

Given: A graph $G = (V, E)$ and a subset $S \subseteq V$.

Question: Is S a secure set of G?

Section 3.3 discusses the complexity of `Is Secure`. In particular, we will show that `Is Secure` is in P if and only if P = NP (Corollary 3.3.8).

Finally, recall the problems `Secure Set` (Problem 1.4.1) and `Global Secure Set` (Problem 1.4.2) introduced in Chapter 1. They are stated again below for convenience.

Problem 3.1.3. `Secure Set`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a secure set of cardinality k or less?

Problem 3.1.4. `Global Secure Set`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a global secure set of cardinality k or less?

The problems `Secure Set` and `Global Secure Set` are central to the study of secure sets and global secure sets. Their complexity will be discussed in Section 3.4. In particular, we will show that

1. `Secure Set` is in P if and only if P = NP (Theorem 3.4.4).

2. If P = NP, then `Global Secure Set` is in P (Corollary 3.4.7).

3. If P $\neq$ NP, both problems may not be in NP.

To date, there is no known polynomial algorithm for solving the problems `Is Secure`, `Secure Set` or `Global Secure Set`.

3.2 Computing a feasible defense using network flow

This section presents a polynomial algorithm for `Feasible Defense` (Problem 3.1.1). Given a graph $G = (V, E)$, a set $S \subseteq V$ and an attack $\mathscr{A}$ on S, the algorithm will compute a feasible defense for $\mathscr{A}$ or determine if none exists. In [BDH07], a linear programming formulation was provided for `Feasible Defense` and claimed to always produce integral solutions. The algorithm presented here uses network flow techniques ([Rav93, KT05]) and will always terminate in polynomial time.

`Feasible Defense` can be formulated as a transportation problem and solved by single source single destination maximum network flow. Consider an input instance of `Feasible Defense`: a graph $G = (V, E)$, a set $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$ and an attack $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$ with $A_i \subseteq (N[s_i] - S)$ and ($A_i \cap A_j = \emptyset$ if $i \neq j$). Note that each vertex $s_i \in S$ is attacked by exactly $|A_i|$ attackers, and must be assigned at least $|A_i|$ defenders if a feasible defense is to exist. Construct a network $N = (V', A')$ with capacity function $c : A' \to \mathbb{Z}$ as follows.

1. $V' = \{s', t'\} \cup X \cup Y$, $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$. Note $|V'| = 2|S| + 2$.
2. $A' = \{(s', x_i) : 1 \leq i \leq k\} \cup \{(x_i, y_j) : s_i \in N[s_j] \text{ and } 1 \leq i, j \leq k\} \cup \{(y_j, t') : 1 \leq j \leq k\}$.
3. $$\begin{cases} c(s', x_i) = 1 \text{ for } 1 \leq i \leq k. \\ c(x_i, y_j) = 1 \text{ for } (x_i, y_j) \in A'. \\ c(y_j, t') = |A_j| \text{ for } 1 \leq j \leq k. \end{cases}$$

The network N contains exactly $2|S| + 2$ vertices. One may consider the vertices in X as supplies of defenders in S, with x_i modeling the supply of defender s_i, and vertices in Y as demands (for defenders) of those vertices in S being attacked. The constraint $c(s', x_i) = 1$ ensures each defender s_i may choose to defend at most one vertex, and $c(y_j, t') = |A_j|$ indicates each vertex s_j demands $|A_j|$ defenders. A vertex s_i may defend s_j only if $s_i \in N[s_j]$, in which case $(x_i, y_j) \in A'$ with capacity $c(x_i, y_j) = 1$. A feasible defense exists for the given attack if and only if the maximum $s' - t'$ flow of network N equals $\sum_{j=1}^{k} |A_j|$. Using the Ford-Fulkerson algorithm (see [FF56] or [CLR01] Chapter 26), the maximum flow of N may be computed in $O(f \times (|V'| + |A'|))$ time, where f is the maximum $s' - t'$ flow of network N and $f \leq \sum_{i=1}^{k} c(s', x_i) = |S|$. Since $|V'| = (2|S| + 2) \in O(|S|)$ and $|A'| \leq (2|S| + |S|^2) \in O(|S|^2)$, the algorithm will always terminate in $O(|S|^3)$ time.

If a feasible defense exists, then it can be extracted from the maximum flow solution of network N as follows. For each $(x_i, y_j) \in A'$ where the capacity is consumed, let s_i defend s_j (i.e., let $s_i \in D_j$). Since the maximum flow is $\sum_{j=1}^{k} |A_j| = \sum_{j=1}^{k} c(y_j, t')$, for each y_j in the network, the capacity of (y_j, t') must be fully consumed. This implies $|D_j| = |A_j|$, since each y_j must have its in-flow equal to its out-flow, where the out-flow is $|A_j|$.

3.3 Verifying the security of a set

This section discusses the problem `Is Secure` (Problem 3.1.2). Given a graph $G = (V, E)$ and a set $S \subseteq V$, is S a secure set of G? Can S defend against all possible attacks coming from $N[S] - S$, as specified in Definition 1.2.1? In this section, we will show that `Is Secure` is in the set Co-NP-Complete (Theorem 3.3.7), and a polynomial algorithm for `Is Secure` exists if and only if P = NP (Corollary 3.3.8). The section concludes with a remark on two naive methods for verifying the security of small sets. These methods are used to ensure the security of sets presented in Chapter 7.

To date, there is no known polynomial algorithm for solving `Is Secure`. Recall from Theorem 1.2.3 that a set $S \subseteq V$ is secure if and only if $\forall X \subseteq S, |N[X] \cap S| \geq |N[X] - S|$.

Corollary 3.3.1. If S is not a secure set, then there exists a *witness* set $W \subseteq S$ such that $|N[W] \cap S| < |N[W] - S|$. The absence of any witnesses implies a set is secure.

Then, the complement of `Is Secure` may be stated as follows.

Problem 3.3.2. `Witness`

Given: A graph $G = (V, E)$ and a set $S \subseteq V$.

Question: Does there exist $W \subseteq S$ such that $|N[W] \cap S| < |N[W] - S|$? In other words, is S not a secure set of G?

It is clear that `Witness` is in the class NP. An oracle may provide the desired set W, and a verifier can compute $|N[W] \cap S|$ and $|N[W] - S|$ in polynomial time. We show next that `Witness` is NP-Complete by providing a polynomial transformation from the known NP-Complete problem `Dominating Set` (see Definition 1.3.1 and [GJ79] P.190 [GT2]).

Problem 3.3.3. `Dominating Set`

Given: A graph $G = (V, E)$ and a positive integer $k < |V|$.

Question: Does G have a dominating set of cardinality k or less? That is, a set $S \subseteq V$ such that $|S| \leq k$ and $N[S] = V$.

Transformation 3.3.4. Let $G = (V, E)$ be a graph of order n where $V = \{v_1, v_2, \ldots, v_n\}$ and let $k < n$ be a positive integer. The graph G and the integer k specifies an instance of `Dominating Set`. Construct a graph $H = (V', E')$ as follows.

$V' = V \cup X \cup Y$, where V, X and Y are mutually disjoint, and

1. V is the vertex set of G.

2. $X = A \cup B \cup C$, where A, B and C are mutually disjoint, and $|A| = n^2 - n - k - 1$, $B = \{b_1, b_2, \ldots, b_n\}$ and $|C| = k + 1$. The vertices of A and C are not labeled. Note that $|X| = n^2$.

3. $Y = Y_1 \cup Y_2 \cup \cdots \cup Y_n$, where $Y_1, Y_2, \ldots, Y_n$ are mutually disjoint and $Y_t = \{y_{(t,j)} : 1 \leq j \leq n\}$ for $1 \leq t \leq n$. Note that $|Y_t| = n$ and $|Y| = n^2$.

$$
\begin{aligned}
E' = \quad & E \cup \{xx' : x, x' \in X \text{ and } x \neq x'\} \cup \{v_i a : a \in A \text{ and } 1 \leq i \leq n\} \\
& \cup \{v_i b_i : 1 \leq i \leq n\} \cup \{v_i y_{(t,j)} : v_j \in N_G[v_i] \text{ and } 1 \leq i, j, t \leq n\}
\end{aligned}
$$

Thus, H contains exactly $|V| + |X| + |Y| = 2n^2 + n$ vertices. The graph G is an induced subgraph of H. In graph H, X induces a complete subgraph (clique) and Y is an independent set. In addition, for each $v_i \in V$, v_i is adjacent to all vertices of A, the vertex b_i and vertex $y_{(t,j)}$ whenever v_j is in the closed neighborhood of v_i, for $1 \leq t \leq n$.

Given an instance of `Dominating Set` with $G \leftarrow G$ and $k \leftarrow k$, Transformation 3.3.4 provides an instance of `Witness` with $G \leftarrow H$ and $S \leftarrow (V \cup X)$.

Lemma 3.3.5. With reference to Transformation 3.3.4, G contains a dominating set of cardinality at most k if and only if $(V \cup X)$ contains a witness in H.

Proof. Consider an arbitrary non-empty subset $T \subseteq V$ in H.

$$\begin{aligned} |N_H[T] \cap (V \cup X)| &= |N_H[T] \cap V| + |N_H[T] \cap A| + |N_H[T] \cap B| + |N_H[T] \cap C| \\ &= |N_H[T] \cap V| + (n^2 - n - k - 1) + |T| \end{aligned} \tag{3.1}$$

Note that $|N_H[T] \cap V| = |N_H[T] \cap Y_t|$ for $1 \leq t \leq n$, so

$$\begin{aligned} |N_H[T] - (V \cup X)| &= |N_H[T] \cap Y| \\ &= \sum_{t=1}^{n} |N_H[T] \cap Y_t| \\ &= \sum_{t=1}^{n} |N_H[T] \cap V| \\ &= n \cdot |N_H[T] \cap V| \end{aligned} \tag{3.2}$$

Let S be a dominating set of G of cardinality k or less. Then, $S \subseteq V$ and S is also a subset of the vertices of H. By $N_G[S] = V$, $|N_H[S] \cap V| = n$. Then,

$$\begin{aligned} &|N_H[S] \cap (V \cup X)| \\ &= |N_H[S] \cap V| + (n^2 - n - k - 1) + |S| \quad \text{(By } S \subseteq V \text{ and Equation (3.1))} \\ &\leq n + (n^2 - n - k - 1) + k \quad \text{(By } |N_H[S] \cap V| = n \text{ and } |S| \leq k\text{)} \\ &= n^2 - 1 \\ &< n^2 \end{aligned}$$

$$= n \cdot |N_H[S] \cap V| \qquad \text{(By } |N_H[S] \cap V| = n\text{)}$$

$$= |N_H[S] - (V \cup X)| \quad \text{(By } S \subseteq V \text{ and Equation (3.2))}$$

In other words, $|N_H[S] \cap (V \cup X)| < |N_H[S] - (V \cup X)|$ and S is a witness of $(V \cup X)$ in H. Thus, if G contains a dominating set of cardinality k or less, then $(V \cup X)$ contains a witness.

Conversely, let W be a witness of $(V \cup X)$ in H. We will show that W is a valid dominating set of G of cardinality k or less by showing that $W \subseteq V$, $|W| \leq k$ and $|N_H[W] \cap V| = n$, in that order.

If $(W \cap X) \neq \emptyset$, then

$$|N_H[W] \cap (V \cup X)|$$

$$= |N_H[W] \cap V| + |N_H[W] \cap X|$$

$$= |N_H[W] \cap V| + |X| \qquad \text{(By } (W \cap X) \neq \emptyset \text{ and } X \text{ is clique)}$$

$$\geq |X|$$

$$= n^2$$

$$= |Y|$$

$$\geq |N_H[W] \cap Y|$$

$$= |N_H[W] - (V \cup X)|$$

This is a contradiction to W being a witness. Thus, $(W \cap X) = \emptyset$, and $W \subseteq V$. Next,

$$\begin{aligned}
|N_H[W] \cap (V \cup X)| &< |N_H[W] - (V \cup X)| && (W \text{ is a witness}) \\
|N_H[W] \cap V| + (n^2 - n - k - 1) + |W| &< n \cdot |N_H[W] \cap V| && (\text{By Eq. (3.1) and (3.2)}) \\
(n^2 - n - k - 1) + |W| &< (n-1) \cdot |N_H[W] \cap V|
\end{aligned} \tag{3.3}$$

If $|W| \geq k + 1$, then

$$\begin{aligned}
&(n^2 - n - k - 1) + |W| \\
&\geq (n^2 - n - k - 1) + k + 1 \\
&= n^2 - n \\
&= (n-1) \cdot n \\
&\geq (n-1) \cdot |N_H[W] \cap V| \qquad (\text{By } n = |V| \geq |N_H[W] \cap V|\ )
\end{aligned}$$

This is a contradiction to Equation (3.3). Thus, $|W| \leq k$.

If $|N_H[W] \cap V| < n$, then

$$\begin{aligned}
(n-1) \cdot |N_H[W] \cap V| &> (n^2 - n - k - 1) + |W| && (\text{By Equation (3.3)}) \\
(n-1) \cdot |N_H[W] \cap V| &> n \cdot (n-1) + |W| - k - 1
\end{aligned}$$

So,

$$\begin{aligned}
k &> n \cdot (n-1) + |W| - 1 - (n-1) \cdot |N_H[W] \cap V| \\
&= (n-1)(n - |N_H[W] \cap V|) + |W| - 1 \\
&\geq (n-1) + |W| - 1 && \text{(By } (n - |N_H[W] \cap V|) \geq 1\text{)} \\
&\geq (n-1) + 1 - 1 && \text{(By } |W| \geq 1\text{)} \\
&= (n-1)
\end{aligned}$$

Thus, $k > (n-1)$, or $k \geq n$. This is a contradiction since $k < |V| = n$ by the specification of `Dominating Set` (Problem 3.3.3). Thus, $|N_H[W] \cap V| = n$.

Since $W \subseteq V$ and $|N_H[W] \cap V| = n$, W is a dominating set of G with cardinality $|W| \leq k$. □

By Transformation 3.3.4 and Lemma 3.3.5,

Theorem 3.3.6. `Witness` (Problem 3.3.2) is NP-Complete.

Since `Witness` is the complement of `Is Secure`,

Theorem 3.3.7. `Is Secure` (Problem 3.1.2) is Co-NP-Complete.

Theorem 3.3.7, along with the next corollary, concludes the complexity of `Is Secure`.

Corollary 3.3.8. `Is Secure` (Problem 3.1.2) is in P if and only if P = NP.

Proof. If P = NP, then `Witness` (Problem 3.3.2), a problem in NP, will be in P. So, its complement `Is Secure` will also be in P.

Conversely, if `Is Secure` is in P, then its complement `Witness` is also in P. Since `Witness` is NP-Complete, the result is P = NP. □

The rest of this section presents two naive methods for solving the problem `Is Secure` in the lack of any polynomial solutions. Although polynomial algorithms are not likely to exist for `Is Secure`, brute force algorithms can still work sufficiently fast when the set to be verified is small.

To check if a set S is a secure set of a graph G, one may examine all possible attacks (according to Definition 1.2.1) and determine whether or not a feasible defense exists for each attack using the algorithm given in Section 3.2. Since every attacker in $N[S]-S$ may have several choices on which vertex of S to attack, there are an exponential number of attacks to consider. The complexity of this approach depends on the number of neighbors each attacker has in S. More specifically, the number of attack configurations equals $\prod_{y\in(N[S]-S)} |N[y]\cap S|$. A feasible defense for each attack can be found (or determined that none exists) in $O(|S|^3)$ time as seen in Section 3.2. Note that when verifying a secure set, we only need to consider cases where every attacker $y \in (N[S]-S)$ decides to attack a vertex in $(N[y]\cap S)$.

Another approach is to apply Theorem 1.2.3. Examine every subset $X \subseteq S$ and verify that $|N[X]\cap S| \geq |N[X]-S|$. There are $2^{|S|}$ subsets and it is not clear whether a procedure exists which does not examine at least $\Omega(2^{|S|})$ subsets of S.

Both approaches above require in the worst case exponential time to execute. In [Dut06], several other exhaustive search methods are given, but none has been shown to be effective against large problem instances.

3.4 Computing the security number of a graph

This section discusses the problem `Secure Set` (Problem 3.1.3). Given a graph $G = (V, E)$ and a positive integer $k < |V|$, does G have a secure set of cardinality k or less? We will show that `Secure Set` is in P if and only if P = NP (Theorem 3.4.4), and when P $\neq$ NP, `Secure Set` is probably not in NP. The section will conclude with some remarks on the problem `Global Secure Set` (Problem 3.1.4).

Note that in order for `Secure Set` to be in NP, there must be a polynomial verifier which can verify the existence of a secure set of cardinality k or less in a graph G, when *evidence* of such existence is presented. An obvious evidence from an oracle will be a subset of vertices of cardinality k or less. A polynomial algorithm must then verify that the set presented is in fact a secure set. But, by Corollary 3.3.8, such an algorithm does not exist unless P = NP. So, if P $\neq$ NP, it is unlikely for `Secure Set` to be in NP. For an in depth discussion of the class NP, see [GJ79] or [Gol10].

Next, Transformation 3.4.1 and Lemmas 3.4.2 and 3.4.3 will be used to establish Theorem 3.4.4, which states that `Secure Set` is in P if and only if P = NP.

Transformation 3.4.1. Let $G = (V, E)$ be a graph of order n and let $S \subseteq V$. The graph G and set S specifies an instance of `Is Secure` (Problem 3.1.2). Let $m = |S|$ and label the vertices of G with $S = \{v_1, v_2, \ldots, v_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$. Construct a graph $H = (V', E')$ as follows.

$V' = V \cup A \cup B \cup C$, where V, A, B and C are mutually disjoint, and

1. V is the vertex set of G.

2. $|A| = 3m + 1$. The vertices of A are not labeled.

3. $B = \{b_1, b_2, \ldots, b_m\}$.

4. $C = \{c_1, c_2, \ldots, c_m\}$.

$$
\begin{aligned}
E' = \quad & E \cup \{xx' : x, x' \in (A \cup B) \text{ and } x \neq x'\} \\
& \cup \{xx' : x, x' \in (B \cup C) \text{ and } x \neq x'\} \\
& \cup \{v_i b_i : 1 \leq i \leq m\} \cup \{v_i c_i : 1 \leq i \leq m\} \\
& \cup \{v_i x : (m+1) \leq i \leq n \text{ and } x \in (A \cup B)\}
\end{aligned}
$$

So, H contains exactly $n + 5m + 1$ vertices. The graph G is an induced subgraph of H. In graph H, the sets $(A \cup B)$ and $(B \cup C)$ induce complete subgraphs (cliques), and in addition, for each $v_i \in V$, if $i \leq m$, then $v_i \in S$ and v_i is adjacent to b_i and c_i, and if $i > m$, then $v_i \in (V - S)$ and v_i is adjacent to all vertices of $(A \cup B)$.

Given an instance of `Is Secure` with $G \leftarrow G$ and $S \leftarrow S$, Transformation 3.4.1 provides an instance of `Secure Set` with $G \leftarrow H$ and $k \leftarrow 2|S|$. Next, Lemma 3.4.2 will be used in the proof of Lemma 3.4.3. Lemma 3.4.3 shows that S is a secure set of G if and only if H contains a secure set of cardinality $2|S|$ or less.

Lemma 3.4.2. With reference to Transformation 3.4.1, S is a secure set of G if and only if $(S \cup C)$ is a secure set of H.

Proof. Let S be a secure set of G. Then, by Theorem 1.2.3, $\forall X \subseteq S, |N_G[X] \cap S| \geq |N_G[X] - S|$. We want to show that $(S \cup C)$ is a secure set of H. Let $X' \subseteq (S \cup C)$ be arbitrary and consider two cases.

1. $(X' \cap C) \neq \emptyset$. Let $X = (X' \cap S)$ and,

$$
\begin{aligned}
& |N_H[X'] \cap (S \cup C)| & \\
& = |N_H[X'] \cap C| + |N_H[X'] \cap S| & \\
& = |C| + |N_H[X'] \cap S| & \text{(By } (X' \cap C) \neq \emptyset \text{ and } C \text{ is a clique)} \\
& \geq |C| + |N_H[X] \cap S| & \text{(By } X \subseteq X'\text{)} \\
& \geq |C| + |N_H[X] \cap (V - S)| & (X \subseteq S \text{ and } S \text{ is a secure of } G) \\
& = |B| + |N_H[X] \cap (V - S)| & \text{(By } |C| = m = |B|\text{)} \\
& = |B| + |N_H[X'] \cap (V - S)| & \text{(No edges between } C \text{ and } (V - S)\text{)} \\
& = |N_H[X'] \cap B| + |N_H[X'] \cap (V - S)| & \text{(By } (X' \cap C) \neq \emptyset \text{ and } (B \cup C) \text{ is a clique)} \\
& = |N_H[X'] - (S \cup C)| & \text{(No edges between } (S \cup C) \text{ and } A\text{)}
\end{aligned}
$$

So, if $(X' \cap C) \neq \emptyset$, then $|N_H[X'] \cap (S \cup C)| \geq |N_H[X'] - (S \cup C)|$.

2. $(X' \cap C) = \emptyset$. Then $X' \subseteq S$, and

$$
\begin{aligned}
&|N_H[X'] \cap (S \cup C)| \\
&= |N_H[X'] \cap C| + |N_H[X'] \cap S| \\
&= |X'| + |N_H[X'] \cap S| && \text{(By } X' \subseteq S\text{)} \\
&\geq |X'| + |N_H[X'] \cap (V - S)| && (X' \subseteq S \text{ and } S \text{ is a secure set of } G) \\
&= |N_H[X'] \cap B| + |N_H[X'] \cap (V - S)| && \text{(By } X' \subseteq S\text{)} \\
&= |N_H[X'] - (S \cup C)| && (X' \subseteq S \text{ and no edges between } S \text{ and } A)
\end{aligned}
\tag{3.4}
$$

So, if $(X' \cap C) = \emptyset$, then $|N_H[X'] \cap (S \cup C)| \geq |N_H[X'] - (S \cup C)|$.

In both cases $|N_H[X'] \cap (S \cup C)| \geq |N_H[X'] - (S \cup C)|$, so $(S \cup C)$ is a secure set of H. Conversely, suppose S is not a secure set of G. Let $X \subseteq S$ be a witness of S in G. Then,

$$
\begin{aligned}
&|N_H[X] \cap (S \cup C)| \\
&= |N_H[X] \cap C| + |N_H[X] \cap S| \\
&= |X| + |N_H[X] \cap S| && \text{(By } X \subseteq S\text{)} \\
&< |X| + |N_H[X] \cap (V - S)| && (X \text{ is a witness of } S \text{ in } G) \\
&= |N_H[X] \cap B| + |N_H[X] \cap (V - S)| && \text{(By } X \subseteq S\text{)} \\
&= |N_H[X] - (S \cup C)| && (X \subseteq S \text{ and no edges between } S \text{ and } A)
\end{aligned}
$$

Thus, $|N_H[X] \cap (S \cup C)| < |N_H[X] - (S \cup C)|$, and X is a witness of $(S \cup C)$ in H. Therefore, $(S \cup C)$ is not a secure set of H. $\square$

Lemma 3.4.3. With reference to Transformation 3.4.1, S is a secure set of G if and only if H contains a secure set of cardinality $2|S|$ or less.

Proof. Let S be a secure set of G. By Lemma 3.4.2, $(S \cup C)$ is a secure set of H. Since $|S \cup C| = 2|S|$, H contains a secure set of cardinality $2|S|$ or less.

Conversely, let S' be a secure set of H with $|S'| \leq 2|S|$. We want to show that S is a secure set of G. We first show that $S' = (S \cup C)$.

Suppose $S' - (S \cup C) \neq \emptyset$ and let $x \in S' - (S \cup C)$. Then, $x \in (A \cup B \cup (V - S))$ and $(A \cup B) \subseteq N_H[x]$.

$$\begin{aligned}
4m + 1 &= |A \cup B| \\
&\leq |N_H[x]| && \text{(By } (A \cup B) \subseteq N_H[x]\text{)} \\
&= |N_H[x] \cap S'| + |N_H[x] - S'| \\
&\leq 2 \cdot |N_H[x] \cap S'| && (x \in S' \text{ and } S' \text{ is a secure set)} \\
&\leq 2 \cdot |S'| \\
&\leq 4m && \text{(By } |S'| \leq 2|S| = 2m\text{)}
\end{aligned}$$

This is a contradiction. Thus, $S' - (S \cup C) = \emptyset$, and $S' \subseteq (S \cup C)$.

Next, assume that $(S \cup C) - S' \neq \emptyset$ and consider the following cases.

1. $(S' \cap C) = \emptyset$. Since $S' \subseteq (S \cup C)$, so $S' \subseteq S$ and,

$$
\begin{aligned}
&|N_H[S'] - S'| \\
&\geq |(N_H[S'] - S') \cap B| + |(N_H[S'] - S') \cap C| \\
&= 2|S'| \qquad \text{(By } S' \subseteq S\text{)} \\
&> |S'| \\
&= |N_H[S'] \cap S'|
\end{aligned}
$$

This is a contradiction since S' is a secure set in H.

2. $(S' \cap C) \neq \emptyset$. Consider two sub-cases.

2.1. $(C - S') = \emptyset$. Then, $C \subseteq S'$ and $(S - S') = (S \cup C) - S' \neq \emptyset$. Let $v_i \in (S - S')$ and consider vertex $c_i \in C$. Note that $c_i \in S'$, and

$$
\begin{aligned}
&|N_H[c_i] \cap S'| \\
&= |(N_H[c_i] \cap S') \cap C| + |(N_H[c_i] \cap S') \cap S| && \text{(By } S' \subseteq (S \cup C)\text{)} \\
&= |C| + |N_H[c_i] \cap S \cap S'| && \text{(By } C \subseteq S' \text{ and } C \text{ is a clique)} \\
&= |C| && \text{(By } v_i \notin S'\text{)} \\
&< |B| + 1 && \text{(By } |C| = |B|\text{)} \\
&= |B| + |\{v_i\}| \\
&= |N_H[c_i] - S'| && \text{(By } (B \cup C) \text{ is a clique and } v_i \notin S'\text{)}
\end{aligned}
$$

This is a contradiction since S' is a secure set.

2.2. $(C - S') \neq \emptyset$. Let $c_i \in (C - S')$ and since $(S' \cap C) \neq \emptyset$, let $c_j \in (S' \cap C)$. Then,

$$
\begin{aligned}
&|N_H[c_j] \cap S'| \\
&= |(N_H[c_j] \cap S') \cap C| + |(N_H[c_j] \cap S') \cap S| && \text{(By } S' \subseteq (S \cup C) \text{)} \\
&\leq (|C| - 1) + 1 && \text{(By } c_i \notin S' \text{)} \\
&= |C| \\
&< |B| + 1 \\
&= |N_H[c_j] \cap B| + |\{c_i\}| && \text{(By } (B \cup C) \text{ is clique)} \\
&\leq |N_H[c_j] - S'| && \text{(By } c_i \notin S' \text{)}
\end{aligned}
$$

This is a contradiction since S' is a secure set.

In all cases a contradiction results. Therefore, $(S \cup C) - S' = \emptyset$, and $(S \cup C) \subseteq S'$. So, $S' = (S \cup C)$ and $(S \cup C)$ is a secure set of H. By Lemma 3.4.2, S is a secure set of G. □

Theorem 3.4.4. `Secure Set` (Problem 3.1.3) is in P if and only if P = NP.

Proof. If P = NP, then by Corollary 3.3.8, `Is Secure` (Problem 3.1.2) is in P. Let graph $G = (V, E)$ and positive integer $k < |V|$ be an instance of `Secure Set`. An oracle may provide a secure set of cardinality k or less, and since `Is Secure` is in P, this set can be verified in polynomial time. This puts `Secure Set` in NP. Since P = NP, `Secure Set` is in P.

Conversely, if `Secure Set` is in P, then `Is Secure` may be solved in polynomial time by applying Transformation 3.4.1 and solving the obtained `Secure Set` instance in polynomial time. The answer is correct due to Lemma 3.4.3. This puts `Is Secure` in P, and by Corollary 3.3.8, P = NP. $\square$

The rest of this section discusses the complexity of finding minimum global secure sets and related problems. In particular, we show that a global secure set can be verified in polynomial time if and only if P = NP (Lemma 3.4.6). In addition, if P = NP, then a minimum global secure set can be found in polynomial time (Corollary 3.4.7), but it is not known whether the converse is true.

The problem of verifying a global secure set can be stated as follows.

Problem 3.4.5. `Is Global Secure`

Given: A graph $G = (V, E)$ and a subset $S \subseteq V$.

Question: Is S a global secure set of G?

A set S is a global secure set if it is a secure set and a dominating set. A dominating set can be verified in polynomial time, and a secure set can be verified in polynomial time if and only if P = NP.

Lemma 3.4.6. `Is Global Secure` (Problem 3.4.5) is in P if and only if P = NP.

Proof. If P = NP, then by Corollary 3.3.8, `Is Secure` is in P. Let $G = (V, E)$ and $S \subseteq V$ be an instance of `Is Global Secure`. It can be verified that $N[S] = V$ and S is a secure set, both in polynomial time. Thus, `Is Global Secure` is in P.

Conversely, suppose `Is Global Secure` is in P. Let $G = (V, E)$ and $S \subseteq V$ be an instance of `Is Secure`. Then, let graph $G[N[S]]$ and the set S be an instance of `Is Global Secure`. Since S is a dominating set of $G[N[S]]$, S is a global secure set of $G[N[S]]$ if and only if S is a secure set of G. Since `Is Global Secure` is in P, the above transformation provides a polynomial algorithm for `Is Secure`. So, `Is Secure` is in P, and by Corollary 3.3.8, P = NP. $\square$

Corollary 3.4.7. If P = NP, then `Global Secure Set` (Problem 3.1.4) is in P.

Proof. If P = NP, then by Lemma 3.4.6, `Is Global Secure` is in P, and one may verify a global secure set in polynomial time. Then, `Global Secure Set` is in NP. Since P = NP, `Global Secure Set` is in P. $\square$

We have not shown that the converse of Corollary 3.4.7 is true. This is posted as Open Problem 9.3.4. If P $\neq$ NP, then by Lemma 3.4.6, `Is Global Secure` is not in P, and `Global Secure Set` may not be in NP.

CHAPTER 4

CHARACTERIZATION OF SECURE SETS

4.1 Introduction

Consider the secure set characterization theorem (Theorem 1.2.3), stated again as Theorem 4.1.1.

Theorem 4.1.1. ([BDH07], Theorem 11) Let $G = (V, E)$ be a graph. A non-empty set $S \subseteq V$ is a secure set if and only if $\forall X \subseteq S, |N[X] \cap S| \geq |N[X] - S|$.

As mentioned in Section 1.2, Theorem 4.1.1 is fundamental to the study of secure sets. It states that the obvious necessary condition ($\forall X \subseteq S, |N[X] \cap S| \geq |N[X] - S|$) for a set to be secure is also sufficient. This theorem provides an equivalent definition of secure sets which is similar to the definition of defensive alliances (Definition 1.1.1). In addition, the applications of Theorem 4.1.1 are extensive, as demonstrated in a number of publications ([Dut06, DLB08, ED08, Dut09, KOY09, HD10, Jes10, DH10, HD11]), as well as in Chapter 3, and it will also be applied in later chapters of this work.

Theorem 4.1.1 resembles the well known Hall's Matching Theorem in graph theory.

Theorem 4.1.2. (Hall's Matching Theorem) ([Hal35] or [Wes04] Theorem 3.1.11) An (A, B)-bipartite graph has a matching that saturates A if and only if $\forall X \subseteq A, |N(X)| \geq |X|$.

A matching saturates set A if every vertex of A is incident to an edge of the matching. Notice the similarity between Theorem 4.1.1 and 4.1.2. Both conditions specify inequalities involving all subsets of the set in question, comparing the cardinalities of a subset and its neighborhoods. On the other hand, the computational complexity of these two conditions are quite distinct. The condition ($\forall X \subseteq A, |N(X)| \geq |X|$) of Hall's Theorem can be evaluated in polynomial time, by computing a maximum matching of the bipartite graph in question. But, as seen in Section 3.3, one cannot verify a secure set in polynomial time unless P = NP, so the condition of Theorem 4.1.1 cannot be evaluated in polynomial time unless P = NP.

Section 4.2 will present a solution to the problem `Feasible Defense` (Problem 3.1.1) using maximum bipartite matching. The model used in Section 4.2 is similar to the one given in Section 3.2, and it will establish some of the notation used in Section 4.3. Then, Section 4.3 presents a proof of Theorem 4.1.1 as an application of Hall's Theorem. This proof is an alternative to the original one given in [BDH07].

4.2 Computing a feasible defense using maximum matching

Consider an input instance of `Feasible Defense`: A graph $G = (V, E)$, a subset $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$ and an attack $\mathscr{A}$ on S, where $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$, $A_i \subseteq (N[s_i] - S)$

and ($A_i \cap A_j = \emptyset$ if $i \neq j$). Let $(N[S]-S) = \{a_1, a_2, \ldots, a_m\}$ and construct a bipartite graph $H = (V', E')$ as follows.

$V' = A \cup B$, where A and B are disjoint and each is an independent set, and

$$A = \{(a_i, s_j) : a_i \in A_j \text{ and } 1 \leq j \leq k\}$$

$$B = \{b_1, b_2, \ldots, b_k\}$$

$E' = \{((a_i, s_j), b_t) : s_t \in N[s_j]\}$.

The graph H has at most $m+k$ vertices. The vertices in A model demands for defenders and the vertices in B model supplies of defenders. Each vertex in A is labeled by a pair of the form (a, s) where $a \in (N[S]-S)$ and $s \in S$. For each attacker $a_i \in (N[S]-S)$, if $a_i \in A_j$, then a_i attacks s_j, in which case the pair (a_i, s_j) is in A. Since an attacker can attack at most one vertex, for $(a, s), (a', s') \in A$, if $(a, s) \neq (a', s')$, then $a \neq a'$, but s may be equal to s'. For $(a_i, s_j) \in A$, a_i attacks s_j in G, and this attack must be repelled by a vertex in $(N[s_j] \cap S)$, so for each $s_t \in N[s_j]$, (a_i, s_j) is adjacent to b_t, where b_t corresponds to the supply of defender $s_t \in S$, and s_t may be used to defend s_j. A feasible defense exists in G for the given attack $\mathscr{A}$ if and only if there exists a matching of H that saturates A (or equivalently, the cardinality of a maximum matching of H equals $|A|$).

If there exists a matching M that saturates A in graph H, then a feasible defense for attack $\mathscr{A}$ in graph G can be extracted as follows. For each $((a_i, s_j), b_t) \in M$, let s_t defend s_j (i.e., let $s_t \in D_j$). If a_i attacks s_j, then $(a_i, s_j) \in A$, and since M saturates A, there exists $((a_i, s_j), b_t) \in M$, and s_t defends s_j from the attacker a_i. In addition, since M is a matching, different attackers are defended by different defenders.

4.3 Secure sets characterization via Hall's Matching Theorem

Theorem Let $G = (V, E)$ be a graph. A non-empty set $S \subseteq V$ is a secure set if and only if $\forall X \subseteq S, |N_G[X] \cap S| \geq |N_G[X] - S|$.

Proof. Let $G = (V, E)$ be a graph and let $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$ be a subset of vertices.

Suppose S is a secure set of G and let X be an arbitrary subset of S. Consider an attack $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$ with $\bigcup_{s_j \in X} A_j = (N_G[X] - S)$. That is, every attacker adjacent to X attacks a vertex of X. Since S is a secure set, there exists a feasible defense $\mathscr{D} = \{D_1, D_2, \ldots, D_k\}$ with $|D_j| \geq |A_j|$ for $1 \leq j \leq k$. Then, $|N_G[X] \cap S| \geq \sum_{s_j \in X} |D_j| \geq \sum_{s_j \in X} |A_j| = |N_G[X] - S|$. So, $|N_G[X] \cap S| \geq |N_G[X] - S|$.

Conversely, suppose S is not a secure set. We want to show $(\exists X \subseteq S : |N_G[X] \cap S| < |N_G[X] - S|)$. Since S is not secure, let $\mathscr{A} = \{A_1, A_2, \ldots, A_k\}$ be an attack which is not defendable (i.e., no feasible defense exists for $\mathscr{A}$). Apply the transformation presented in Section 4.2 and construct a bipartite graph $H = (V', E')$. Since $\mathscr{A}$ is not defendable, no matching of H saturates A. By Hall's Theorem, there exists $T \subseteq A$ such that $|T| > |N_H(T)|$. Let $W_a = \{a_i : (a_i, s_j) \in T\}$ be the attackers in $(N_G[S] - S)$ corresponding to vertices of T and let $W = \{s_j : (a_i, s_j) \in T\}$ be the vertices attacked by those in W_a. Notice $W \subseteq S$ and $W_a \subseteq (N_G[W] - S)$. Then,

$|N_G[W] - S|$

$\geq |W_a|$ (By $W_a \subseteq (N_G[W] - S)$)

$= |T|$ $((a, s) \neq (a', s') \to a \neq a')$

$> |N_H(T)|$

$= |\{b_t : (a_i, s_j) \in T \text{ and } s_t \in N_G[s_j]\}|$

$= |\{s_t : (a_i, s_j) \in T \text{ and } s_t \in N_G[s_j]\}|$ ($b_t \in B$ corresponds to $s_t \in S$)

$= |\{s_t : s_j \in W \text{ and } s_t \in N_G[s_j]\}|$

$= |N_G[W] \cap S|$

Thus, there exists $W \subseteq S$ such that $|N_G[W] \cap S| < |N_G[W] - S|$. $\square$

CHAPTER 5

ROOTED SECURE SETS OF TREES

The last two chapters discuss the complexity and characterization of secure sets. Those chapters focus on the properties of secure sets in general graphs. As seen in Chapter 3, finding a minimum secure set or even verifying the validity of a secure set can be a difficult problem. But, these problems are not as intractable when the graphs under consideration are restricted to certain special classes. In the remaining chapters, we discuss problems related to secure sets on trees and grid-like graphs (Definition 1.4.4). This chapter and the next chapter discuss two problems related to secure sets on trees, `Rooted Secure Set` (Problem 5.1.1) on trees and `Global Secure Set` (Problem 1.4.2) on trees. Finally, Chapters 7 and 8 discuss the global security number of grid-like graphs.

5.1 Introduction

When the vertices of a graph represent a set of entities, such as countries, territories or people, it is interesting to ask for a smallest secure set that contains a given country, territory or person. Consider an example in national security, a minister of country C would like to establish an alliance with neighboring countries such that the alliance forms a secure set, in order to ensure protection for country C. Given the graph which models this situation (as described in Chapter 1), which countries should the minister include in the alliance? Problem 5.1.1 presents the situation in graph theoretical terms.

Problem 5.1.1. `Rooted Secure Set`

Given: A graph $G = (V, E)$, a vertex $r \in V$ and a positive integer $k < |V|$.

Question: Does there exist a secure set $S \subseteq V$, such that $|S| \leq k$ and $r \in S$?

`Rooted Secure Set` asks for a small secure set S that contains a specified vertex r in a graph G. The vertex r is called the *root* of S. A minimum secure set of G will not be of use for this problem unless the set contains r. In the context of existence of a polynomial algorithm, `Rooted Secure Set` is at least as hard as `Secure Set` (Problem 1.4.1).

Observation 5.1.2. If a polynomial algorithm exists for `Rooted Secure Set`, then one exists for `Secure Set`. One such algorithm takes the disjunction of the answers of $|V|$ `Rooted Secure Set` instances, where each instance specifies a different vertex in V as the root.

Note that a degree one vertex always forms a secure set. So in any tree T, $s(T) = 1$, as every tree of order $n \geq 2$ contains at least two degree one vertices. On the other hand, for a rooted secure set S containing vertex r, the cardinality of S is one only if $\deg(r) = 1$.

This chapter provides polynomial algorithms for finding the cardinality of a minimum rooted secure set of a tree. First, we examine the properties of such a set. Let r be a vertex of a tree T that must be contained in a secure set. Consider T as a rooted tree with root r. For vertex $v \in V(T)$, let T_v denote the subtree of T rooted at v with respect to r, and let p_v denote the parent of v in T_r when $v \neq r$. Let c_v denote the number of children of v in T_v. Notice a minimum rooted secure set of T containing r must be connected, for otherwise the maximal connected subset that contains r forms a smaller rooted secure set.

Lemma 5.1.3. Let T be a tree and $S \subseteq V(T)$ be a connected subset of vertices (i.e., $T[S]$ is a connected graph). Then, $|N(y) \cap S| = 1$ for all $y \in N[S] - S$.

Proof. Let $y \in (N[S] - S)$ and assume that $|N(y) \cap S| > 1$. Let u and v be distinct vertices in $N(y) \cap S$. Since $u, v \in S$ and $T[S]$ is a connected graph, there exists a path from u to v in $T[S]$ and a second path u, y, v outside $T[S]$. So, T contains a cycle, contradicting the assumption that T is a tree. □

Lemma 5.1.3 states that every vertex in $(N[S] - S)$ has exactly one neighbor in S. Thus, every vertex $y \in (N[S] - S)$ can only attack the one neighbor it has in S. This means there is an *unique* attack on S, and S is a secure set if and only if this unique attack is defendable. As seen in Section 3.2, the problem of deciding whether a given attack is defendable (i.e., `Feasible Defense`, Problem 3.1.1) can be solved in polynomial time. Therefore, when restricted to trees, `Is Secure` (Problem 3.1.2) can be solved in polynomial time, and `Rooted Secure Set` is in the class NP.

Definition 5.1.4. Let S be a connected subset of vertices of a tree T and let $\mathscr{A}$ be the unique attack on S. Then, for $x \in S$, let A_x denote the vertices in $(N[S] - S)$ attacking x.

So, $A_x \in \mathscr{A}$ and $A_x = N[x] - S$. If $\mathscr{D}$ is a defense of S (not necessarily a feasible defense for $\mathscr{A}$), then D_x denotes the vertices in S defending x. So, $D_x \in \mathscr{D}$ and $D_x \subseteq (N[x] \cap S)$.

Lemma 5.1.5. Let T be a tree with root r. Let S be a secure set of T containing r, such that $T[S]$ is a connected graph. Let $\mathscr{A}$ be the unique attack on S. If there exists a feasible defense $\mathscr{D}$ for $\mathscr{A}$ such that p_v defends v for some $v \in S - \{r\}$, then S is not a minimum rooted secure set.

Proof. Let $\mathscr{D}$ be a feasible defense of $\mathscr{A}$ for which p_v defends v. We show that S is not a minimum rooted secure set by showing that $S' = S - V(T_v)$ is a secure set containing r. Let $\mathscr{A}'$ be the attack on S'. Note that $A'_x = A_x$ for $x \in S' - \{p_v\}$ and $A'_{p_v} = A_{p_v} \cup \{v\}$. Consider a defense $\mathscr{D}'$ where $D'_x = D_x$ for $x \in S' - \{p_v\}$ and $D'_{p_v} = D_{p_v} \cup \{p_v\}$ ($p_v \in D_v$, but v no longer requires defending since $v \notin S'$). $\mathscr{D}'$ is a feasible defense for $\mathscr{A}'$, so S' is a rooted secure set and S is not minimum. $\square$

By Lemma 5.1.5, if S is a minimum secure set containing the root r, then in every feasible defense $\mathscr{D}$ of the unique attack $\mathscr{A}$, p_v does not defend v for all $v \in (S - \{r\})$. In other words, *a vertex in a minimum rooted secure set either defends itself or its parent, but never any of its children.*

So far, we established that a minimum rooted secure set is connected, and it must have a unique attack (Lemma 5.1.3), and any feasible defense for the attack will never assign a vertex to defend any of its children (Lemma 5.1.5).

Next, Section 5.2 presents an $O(n\Delta)$ algorithm for computing the cardinality of a minimum rooted secure set of a tree, where Δ is the maximum degree of the tree. A more specialized analysis will follow in Sections 5.3 and 5.4, which results in an $O(n \lg(\Delta))$ algorithm.

5.2 An $O(n\Delta)$ algorithm

This section presents an $O(n\Delta)$ algorithm for computing the cardinality of a minimum rooted secure set of a tree. We employ Wimer's method ([WHL85, Wim87]). The following is a brief overview of rooted tree compositions used in Wimer's method.

Definition 5.2.1. Rooted Trees ([WHL85])

1. The triple $(\{x\}, \emptyset, x)$ is a rooted tree with root x.

2. If $T_1 = (V_1, E_1, r_1)$ and $T_2 = (V_2, E_2, r_2)$ are rooted trees with roots r_1 and r_2 respectively, then $T_1 \circ T_2 = (V_1 \cup V_2, E_1 \cup E_2 \cup \{r_1 r_2\}, r_1)$ is a rooted tree with root r_1.

3. Nothing is a rooted tree unless it can be obtained by a finite number of applications of rules 1 and 2.

In Definition 5.2.1, Rule 1 states that a single vertex is a rooted tree. This is the smallest rooted tree. Rule 2 describes a *tree composition*, whereby a new rooted tree is constructed from two smaller rooted trees $T_1 = (V_1, E_1, r_1)$ and $T_2 = (V_2, E_2, r_2)$, by adding an edge between r_1 and r_2, and selecting r_1 as the root of the new tree. This operation is a binary operation and is denoted by the $\circ$ operator. Note that $\circ$ is not commutative.

In the algorithm that follows, let S be a minimum rooted secure set containing the root r of tree T. Let $v \in V(T)$ be arbitrary, let $S_v = S \cap V(T_v)$ be the vertices in S within the subtree T_v, and let $U = \{u_1, u_2, \ldots, u_k\}$ be the children of v. Recall that c_v denotes the number of children of v (i.e., $c_v = k$, we use them interchangeably for convenience). Consider the subtree T_v, and associate with it an array of integer values $\alpha(T_v) = \left\{\alpha_i(T_v) : -\left\lfloor \frac{c_v}{2} \right\rfloor - 1 \leq i \leq \left\lceil \frac{c_v}{2} \right\rceil + 1\right\}$.

Recall from Definition 5.1.4 that for $x \in S$, A_x denotes the set of attackers of x and D_x denotes the set of defenders of x, when an attack or defense is given.

Definition 5.2.2. The entry $\alpha_i(T_v)$ is an integer representing the cardinality of a minimum set S_v such that

(i) $v \in S_v$, and

(ii) for $x \in S_v - \{v\}$, $|D_x| \geq |A_x|$, and

(iii) $|D_v \cap U| + i \geq |A_v \cap U|$. That is, among the children of v, the difference between the number of attackers of v and the number of defenders of v is at most i, where i ranges from $\left(-\left\lfloor \frac{c_v}{2} \right\rfloor - 1\right)$ to $\left(\left\lceil \frac{c_v}{2} \right\rceil + 1\right)$, and

(iv) $\mathscr{A}$ is the unique attack on S and $\mathscr{D}$ is a defense for $\mathscr{A}$.

Note that i may be negative. The value of $\alpha_i(T_v)$ is the cardinality of a minimum set S_v, where if $\mathscr{A}$ is the unique attack on S and $\mathscr{D}$ is a defense (not necessarily feasible) for S, then every vertex in $S_v - \{v\}$ is protected. Furthermore, if $i \geq 0$, then v will be protected if it receives i more defenders in addition to its defenders among its children, and if $i < 0$, v is protected by its children and will remain protected if $|i|$ more attackers attack v.

To compute $\alpha(T_v)$, initialize $T_1 = (\{v\}, \emptyset, v)$, with associated values $\alpha(T_1)$. Then, let T_2 be $T_{u_1}, T_{u_2}, \ldots, T_{u_k}$ in sequence and compute $\alpha(T_2)$ (or $\alpha(T_{u_j}), 1 \leq j \leq k$) recursively. For each $T_2 = T_{u_j}$, apply tree composition on T_1 and T_2, and let this result be the new T_1 ($T_1 \leftarrow T_1 \circ T_2$), at the same time compute new values for the new $\alpha(T_1)$. One may consider each tree composition as attaching a child subtree T_{u_j} of v onto T_1, which is rooted at v. When every child subtree of v is attached, T_1 is transformed into T_v, and $\alpha(T_1)$ also contains the desired values of $\alpha(T_v)$.

We now discuss initial (base case) values for $\alpha(T_1)$, as well as how to compute the new values of $\alpha(T_1)$, given its old values and $\alpha(T_2)$. Initially, T_1 is a single vertex $\{v\}$. Since v has no children, the set $S_v = \{v\}$ is a valid configuration only if $i \geq 0$, in which case $|S_v| = 1$. The initial values of $\alpha_i(T_1)$ are then

$$\alpha_i(T_1) = \begin{cases} 1 & \text{if } 0 \leq i \leq \left(\left\lceil \frac{c_v}{2} \right\rceil + 1\right) \\ \infty & \text{if } \left(-\left\lfloor \frac{c_v}{2} \right\rfloor - 1\right) \leq i \leq -1 \end{cases} \tag{5.1}$$

Let $T_2 = T_{u_j}$ be a child subtree of v and let $r_2 = u_j$ be the root of T_2. Consider $T_{12} = T_1 \circ T_2$. When combining the trees T_1 and T_2, we must decide on the *role* of r_2 in this new tree T_{12}. There are three cases to be considered.

1. $r_2 \in S_v$ and r_2 defends v. In this case, $S_{r_2} \subset S_v$ and $|S_{r_2}| = \alpha_0(T_2)$. In other words, among the children of r_2, the number of attackers of r_2 is at most the number of defenders. Since r_2 defends v and not itself, the defenders of r_2 must be among the children of r_2.

2. $r_2 \in S_v$ and r_2 defends itself. In this case r_2 is included in S_v so that it does not attack v. That is, r_2 is a *neutral* vertex with respect to v. Then, $S_{r_2} \subset S_v$ and $|S_{r_2}| = \alpha_1(T_2)$. In other words, among the children of r_2, there may be one more attackers than defenders, accounting for the fact that r_2 is an additional defender of itself.

3. $r_2 \notin S_v$. Here, r_2 is an attacker of v, and $S \cap V(T_2) = \emptyset$.

As mentioned in Lemma 5.1.5 and the remarks that followed it, r_2 will never defend any of its children, otherwise, S is not a minimum rooted secure set. The associated values $\alpha_i(T_{12})$ can be computed following the 3 cases above.

$$\alpha_i(T_{12}) = \min \begin{cases} \alpha_{i+1}(T_1) + \alpha_0(T_2) & \text{(Type 1)} \\ \alpha_i(T_1) + \alpha_1(T_2) & \text{(Type 2)} \\ \alpha_{i-1}(T_1) & \text{(Type 3)} \end{cases} \tag{5.2}$$

Equation (5.1) describes the base case and equation (5.2) describes the recursive formulation for computing $\alpha(T_v)$. When computing $\alpha(T_v)$, we consider possible candidates for the set S_v, and compute partial feasible defenses where every vertex in S_v is protected, with the possible exception of v. When $v \neq r$, $\alpha(T_v)$ is then used for constructing $\alpha(T_{p_v})$, and at that point v will be protected.

The cardinality of a minimum secure set that contains r is $\alpha_1(T_r)$, since r has no parent and can defend itself. Note that the construction of $\alpha(T_v)$ only utilizes $\alpha_0(T_{u_j})$ and $\alpha_1(T_{u_j})$ for each $u_j \in U$. The final solution is $\alpha_1(T_r)$. Thus, an algorithm does not need to compute all entries of $\alpha(T_v)$, but only $\{\alpha_0(T_v), \alpha_1(T_v)\}$. Next, we present the pseudo-code of the algorithm, followed by a justification of the range $\left(-\left\lfloor \frac{c_v}{2} \right\rfloor - 1\right)$ to $\left(\left\lceil \frac{c_v}{2} \right\rceil + 1\right)$ of $\alpha_i(T_v)$ used in the algorithm.

Algorithm 5.2.3.

Input: A rooted tree T_v.

Output: $\alpha_0(T_v)$ and $\alpha_1(T_v)$.

$\texttt{RootedSecure}(T_v)$

1. Initialize $\alpha_i(T_1) = \begin{cases} 1 & \text{if } 0 \leq i \leq \left(\left\lceil \frac{c_v}{2} \right\rceil + 1\right) \\ \infty & \text{if } \left(-\left\lfloor \frac{c_v}{2} \right\rfloor - 1\right) \leq i \leq -1 \end{cases}$

2. For $j = 1$ to c_v

    2.1. $\{\alpha_0(T_2), \alpha_1(T_2)\} \leftarrow \texttt{RootedSecure}(T_{u_j})$

2.2. For $i = \max\{j - c_v, -j\}$ to $\min\{c_v + 1 - j, j\}$

$$\alpha_i(T_{12}) \leftarrow \min\{\alpha_{i+1}(T_1) + \alpha_0(T_2), \alpha_i(T_1) + \alpha_1(T_2), \alpha_{i-1}(T_1)\}$$

2.3. For $i = \max\{j - c_v, -j\}$ to $\min\{c_v + 1 - j, j\}$

$$\alpha_i(T_1) \leftarrow \alpha_i(T_{12})$$

3. Return $\{\alpha_0(T_1), \alpha_1(T_1)\}$

As aforementioned, the algorithm only needs to compute the values of $\alpha_0(T_v)$ and $\alpha_1(T_v)$. However, the entry of $\alpha_i(T_1)$ needs to be computed for other values of i as an intermediate step of the algorithm, as each child of v is attached. For example, in Figure 5.1 the only possible set that can realize $\alpha_0(T_v) = 3$ is $S_v = \{v, u_3, u_4\}$. As u_1 and u_2 are attached, both subtrees belong to Type 3 of the recurrence in equation (5.2), and the instance of T_1 after T_{u_1} and T_{u_2} are attached to v looks like that of Figure 5.2. The set marked in Figure 5.2 is not a valid α_0 or α_1 configuration, but it must be computed since it will lead to a valid α_0 or α_1 configuration of T_v. In this case, we store the partial tree in Figure 5.2 as $\alpha_2(T_1)$ and it will eventually lead to a solution of $\alpha_0(T_v)$, as shown in Figure 5.1. For this reason, other values of α_i, in addition to α_0 and α_1, are maintained by the algorithm.

It remains to ensure that the ranges (range for initialization of $\alpha_i(T_1)$, as well as the ranges of for-loops in Steps 2.2 and 2.3) is necessary and sufficient to correctly compute $\alpha_0(T_v)$ and $\alpha_1(T_v)$. As child u_j is attached to v, there can be at most j attackers and at most j defenders of v, so we may bound the for-loops in Steps 2.2 and 2.3 by the range $[-j, j]$. Next, note that the value of $\alpha_i(T_{12})$ depends on $\alpha_{i-1}(T_1)$, $\alpha_i(T_1)$ and $\alpha_{i+1}(T_1)$. With reference to Figure

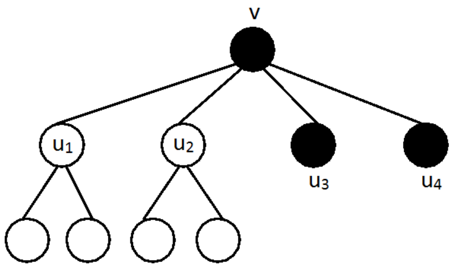

Figure 5.1: A minimum rooted secure set and a valid α_0 configuration.

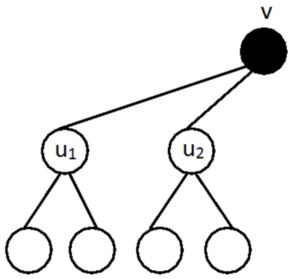

Figure 5.2: A partial solution and an α_2 configuration.

5.3, $\alpha_0(T_v)$ and $\alpha_1(T_v)$ (on row c_v, columns 0 and 1) depend on the cells within the range $[j - c_v, c_v + 1 - j]$ on row j. In Figure 5.3, the cells for which $\alpha_0(T_v)$ and $\alpha_1(T_v)$ depend are marked with vertical bars, and the cells whose values are non-trivial are marked with horizontal bars. The algorithm proceeds and computes values for exactly those cells marked by both vertical and horizontal bars, which are the cells with non-trivial values and affect results $\alpha_0(T_v)$ or $\alpha_1(T_v)$. The range for initialization is then derived from the ranges of the loops and the indexes being referenced: $\left(-\left\lfloor \frac{c_v}{2} \right\rfloor - 1\right) \leq (i-1)$ and $(i+1) \leq \left(\left\lceil \frac{c_v}{2} \right\rceil + 1\right)$.

The above discussion verifies the correctness of Algorithm 5.2.3.

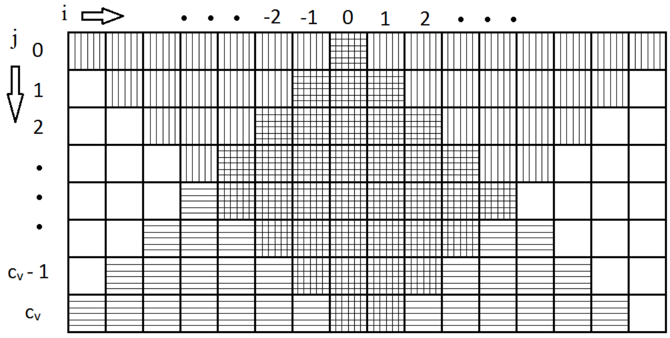

Figure 5.3: Solution Table

Lemma 5.2.4. Algorithm 5.2.3 has time complexity $O(n\Delta)$.

Proof. For each vertex $v \in V(T)$, the algorithm is invoked with T_v. Step 2 is executed c_v times, once for each child of v. Steps 2.2 and 2.3 are executed at most $c_v + 1$ times in each iteration of Step 2. There are, in total, $O\left((c_v)^2\right)$ operations for each vertex $v \in V(T)$. The total number of operations required for solving $\texttt{RootedSecure}(T_r)$ is then proportional to $\displaystyle\sum_{v \in T}(c_v)^2 \leq \sum_{v \in T}(c_v \times \Delta) = (n-1) \times \Delta \in O(n\Delta)$, where Δ is the maximum degree of T. $\square$

The next two sections describe an $O(n \lg(\Delta))$ algorithm, using a strategy that deviates from Wimer's method.

5.3 Feasible partitions and feasibility preserving rule set

Algorithm 5.2.3 uses Wimer's method ([WHL85, Wim87]), which calculates the result of subtree T_v by appending the children subtrees of v, $\{T_{u_1}, T_{u_2}, \ldots, T_{u_k}\}$, to v one at a time.

An alternate strategy, which we describe in this and the next section, constructs the result for T_v by considering the results of $\{T_{u_1}, T_{u_2}, \ldots, T_{u_k}\}$ all at once. This requires more extensive analysis and sophisticated data structures, but results in an $O(n \lg(\Delta))$ algorithm.

Let S be a minimum rooted secure set containing the root r of a tree T. Let $v \in V(T)$ be an arbitrary vertex and let T_v be the subtree of T rooted at v with respect to r. Recall that $S_v = S \cap V(T_v)$ is the set of vertices of S within the subtree T_v. Let $U = \{u_1, u_2, \ldots, u_k\}$ be the children of v. As shown in Section 5.2, there are three cases to be considered for each child u_j of v.

1. $u_j \in S_v$ and u_j defends v. In this case, $|S_{u_j}| = \alpha_0(T_{u_j})$ and u_j is a *defender* of v.
2. $u_j \in S_v$ and u_j defends itself. In this case, $|S_{u_j}| = \alpha_1(T_{u_j})$ and u_j is a *neutral* vertex with respect to v. The vertex u_j is included in S so that it does not attack v.
3. $u_j \notin S_v$. In this case, $|S_{u_j}| = 0$ and u_j is an *attacker* of v.

Among the three choices, at least one will result in an optimal solution for S_v. Let D, N, A be a partition of U and be defined as follows.

Definition 5.3.1.

$$D = \{u_j : u_j \in S_v \text{ and } u_j \text{ defends } v.\}$$

$$N = \{u_j : u_j \in S_v \text{ and } u_j \text{ defends itself.}\}$$

$$A = \{u_j : u_j \notin S_v.\}$$

So, D contains the defenders of v, N contains the vertices that are neutral to v, and A contains the attackers of v.

Definition 5.3.2. The *value* of a partition (D, N, A) of $\{u_1, u_2, \ldots, u_k\}$ is

$$f(D, N, A) = 1 + \sum_{u \in D} \alpha_0(T_u) + \sum_{u \in N} \alpha_1(T_u)$$

The value of a partition is the number of vertices in the corresponding S_v as dictated by the given partition. When computing $\alpha_0(T_v)$, we seek a partition with $|D| = |A|$, and when computing $\alpha_1(T_v)$, a partition with $|D| + 1 = |A|$ is required. The partitions which satisfy the cardinality requirements are *feasible* partitions for the $\alpha_i(T_v)$ in question.

Definition 5.3.3. Let $i \in \{0, 1\}$. A partition (D, N, A) of $\{u_1, \ldots, u_k\}$ is *feasible* for $\alpha_i(T_v)$ if $|D| + i = |A|$. A partition is *optimal* for $\alpha_i(T_v)$ if it is feasible and its value $f(D, N, A) = 1 + \sum_{u \in D} \alpha_0(T_u) + \sum_{u \in N} \alpha_1(T_u)$ is minimum.

Note that according to Definition 5.2.2, a valid α_i configuration must satisfy $|D|+i \geq |A|$. But, when $|D| + i > |A|$, we may move some vertices in D to N and obtain another valid α_i configuration whose value is no worse, since $\alpha_0(T_{u_j}) \geq \alpha_1(T_{u_j})$. Thus, we can only consider partitions with $|D| + i = |A|$.

Our goal is to find optimal partitions for $\alpha_0(T_v)$ and $\alpha_1(T_v)$. This will be done by first obtaining a feasible partition and then transforming it into an optimal one using a set of *exchange* rules. The elementary rule set (Definition 5.3.4) and the feasibility preserving rule set (Definition 5.3.5) are designed for this purpose.

The following six *elementary rules* (E-rules) can be used to transform one partition into another.

Definition 5.3.4. Elementary Rule Set

$E_1 : A \to D$.Move an element from A to D. That is, let $x \in A$ and modify

$A \leftarrow (A - \{x\})$ and $D \leftarrow (D \cup \{x\})$. Rules $E_2, \ldots, E_6$ are defined similarly.

$E_2 : D \to A$.

$E_3 : A \to N$.

$E_4 : N \to A$.

$E_5 : D \to N$.

$E_6 : N \to D$.

The E-rules allow any element from D, N or A to be moved to any other set. Using the E-rules, we can transform any partition into any other one. Hence, given any feasible partition and an optimal partition, we may apply the E-rules to transform the feasible partition into the optimal partition. This can be done by moving any element that is misplaced and put it in the correct set, using the optimal partition as a reference. Note that each element in $\{u_1, \ldots, u_k\}$ is moved at most once, and as a result the order of applications is irrelevant.

Since the order of applications is irrelevant, the E-rules can be grouped together into several groups such that each group of E-rules transform one feasible partition into another, improved, feasible partition. In Definition 5.3.5, we provide a set of these groups (R-rules) and follow with a justification that these are sufficient for transforming any feasible partition into a known optimal partition (Theorem 5.3.10). Then, when an optimal partition is *not* given, the difficulty lies in deciding which of the rules should be used, to which elements, and in which order. Section 5.4 will demonstrate how the R-rules may be applied to a feasible partition and produce an optimal partition, without foreknowledge of an optimal solution.

Definition 5.3.5. Feasibility Preserving Rule Set

$R_1 : D \to A, A \to D$. Exchange elements between A and D. Let $x \in A, y \in D$ and modify $A \leftarrow (A - \{x\}) \cup \{y\}$, $D \leftarrow (D - \{y\}) \cup \{x\}$. The other R-rules are similarly defined.

$R_2 : D \to N, N \to D.$

$R_3 : N \to A, A \to N.$

$R_4 : A \to N, D \to N.$

$R_5 : N \to A, N \to D.$

$R_6 : A \to D, D \to N, N \to A.$

$R_7 : A \to N, N \to D, D \to A.$

$R_8 : A \to D, N \to A, N \to A.$

$R_9 : D \to A, N \to D, N \to D.$

$R_{10} : A \to D, D \to N, D \to N.$

$R_{11} : D \to A, A \to N, A \to N.$

Note that for R_8 two *distinct* elements in N are moved to A (and an element in A is moved to D). The case is similar for R_9, R_{10} and R_{11}. In R_5, the element that moves from $N \rightarrow A$ must be different from the one moving from $N \rightarrow D$.

Each R-rule is composed of 2 or 3 E-rules. For example, R_1 is E_2 followed by E_1. In addition, if (D, N, A) is a feasible partition for $\alpha_i(T_v)$, then (D, N, A) remains feasible for $\alpha_i(T_v)$ after any number of applications of R_1 through R_{11}.

In the following, let (D, N, A) be an arbitrary feasible partition for $\alpha_i(T_v)$ and let (D_o, N_o, A_o) be an optimal one, for $i \in \{0, 1\}$. Theorem 5.3.10 shows that starting from any feasible partition (D, N, A), there exists a sequence of R-rule applications that transforms (D, N, A) into an optimal partition, given foreknowledge of such a partition, namely (D_o, N_o, A_o). First, we introduce some definitions and lemmas needed in the proof of Theorem 5.3.10. The lemmas will help determine whether a sequence of R-rule applications has resulted in the optimal partition.

Given a feasible partition (D, N, A) and an optimal partition (D_o, N_o, A_o), if the two partitions are not equal, then some elements are misplaced in (D, N, A) and must be moved to the correct set with respect to (D_o, N_o, A_o). For example, an element may be in $D \cap A_o$, which means it is currently in D of the partition (D, N, A), and must be moved to A in order to transform (D, N, A) into (D_o, N_o, A_o). Definition 5.3.6 introduces a short hand notation for denoting the sets of elements that need to be moved in this fashion, based on which set of the partition the elements are currently in (with respect to (D, N, A)) and which set they must be moved to, with respect to (D_o, N_o, A_o).

Definition 5.3.6. Let (D, N, A) and (D_o, N_o, A_o) be given as described above. Let X_Y denote the set of elements that must be moved from X to Y in order to transform (D, N, A) into (D_o, N_o, A_o), for $X, Y \in \{D, N, A\}$. Elements that do not need to be moved are denoted X_X. With this notation, for example, $D_A = \{u : u \in D \text{ and } u \in A_o\} = D \cap A_o$, which are the elements currently in D that must be moved to A. $D = (D_D \cup D_N \cup D_A)$ and $D_o = (D_D \cup N_D \cup A_D)$.

Lemma 5.3.7 presents a characterization for determining when two feasible partitions are equal. Then, Lemma 5.3.8 presents two sufficient conditions for determining when (D, N, A) has been transformed into (D_o, N_o, A_o), based on the notation given in Definition 5.3.6.

Lemma 5.3.7. Let (D_1, N_1, A_1) and (D_2, N_2, A_2) be arbitrary feasible partitions for $\alpha_i(T_v)$. Then, $(D_1, N_1, A_1) = (D_2, N_2, A_2)$ if and only if $(D_1 \subseteq D_2)$ and $(A_2 \subseteq A_1)$.

Proof. If $(D_1, N_1, A_1) = (D_2, N_2, A_2)$, then $(D_1 = D_2)$, $(A_1 = A_2)$ and the claim holds.

Conversely, suppose $(D_1 \subseteq D_2)$ and $(A_2 \subseteq A_1)$. Then, $(|D_2| + i) = |A_2|$ because (D_2, N_2, A_2) is a feasible partition for $\alpha_i(T_v)$. $|D_1| \leq |D_2|$ and $|A_2| \leq |A_1|$ due to set inclusion. So, $(|D_1| + i) \leq (|D_2| + i) = |A_2| \leq |A_1|$. But, $(|D_1| + i) = |A_1|$ because (D_1, N_1, A_1) is also a feasible partition for $\alpha_i(T_v)$. So $(|D_1| + i) = (|D_2| + i) = |A_2| = |A_1|$. Then, $|D_1| = |D_2|$ and $|A_1| = |A_2|$, and so $D_1 = D_2$ and $A_1 = A_2$. $\square$

Lemma 5.3.8. The following are each sufficient conditions indicating that (D, N, A) has been transformed into (D_o, N_o, A_o).

(1) $N_D = A_D = A_N = \emptyset$, or

(2) $D_A = D_N = N_A = \emptyset$

Proof.

(1) Recall that $D_o = (D \cap D_o) \cup (N \cap D_o) \cup (A \cap D_o) = (D_D \cup N_D \cup A_D)$ and $A = (A_D \cup A_N \cup A_A)$. Then, $(N_D = A_D = \emptyset) \to (D_o = D_D = D \cap D_o) \to (D_o \subseteq D)$. Next, $(A_D = A_N = \emptyset) \to (A = A_A = A \cap A_o) \to (A \subseteq A_o)$. So, $(D_o \subseteq D)$ and $(A \subseteq A_o)$. The conclusion follows from Lemma 5.3.7.

(2) Similarly, $(D_A = D_N = \emptyset) \to (D = D_D = D \cap D_o) \to (D \subseteq D_o)$. Next, $(D_A = N_A = \emptyset) \to (A_o = A_A = A \cap A_o) \to (A_o \subseteq A)$. So, $(D \subseteq D_o)$ and $(A_o \subseteq A)$. The conclusion follows from Lemma 5.3.7.

□

Lemma 5.3.9 describes four situations that cannot occur in a feasible partition (D, N, A) with respect to an optimal partition (D_o, N_o, A_o). Then, Theorem 5.3.10 will show that any feasible partition (D, N, A) for $\alpha_i(T_v)$ can be transformed into an optimal one using the R-rules in Definition 5.3.5, given foreknowledge of such an optimal partition.

Lemma 5.3.9. Let (D, N, A) be an arbitrary feasible partition for $\alpha_i(T_v)$ and let (D_o, N_o, A_o) be an optimal one. Then, *none* of the following is true.

(1) $N_D = D_N = A_N = \emptyset$ and $|N_A| = 1$.

(2) $A_N = N_A = N_D = \emptyset$ and $|D_N| = 1$.

(3) $A_N = D_N = N_A = \emptyset$ and $|N_D| = 1$.

(4) $N_D = N_A = D_N = \emptyset$ and $|A_N| = 1$.

Proof. Since both (D, N, A) and (D_o, N_o, A_o) are feasible partitions of $\{u_1, u_2, \ldots, u_k\}$ for $\alpha_i(T_v)$, $(|D| + i) = |A|$, $(|D_o| + i) = |A_o|$ and $|D| + |N| + |A| = |D_o| + |N_o| + |A_o|$. Then,

$$\begin{aligned} |D| + |A| + |N| &\equiv |D_o| + |A_o| + |N_o| \pmod 2 \\ 2|D| + i + |N| &\equiv 2|D_o| + i + |N_o| \pmod 2 \\ |N| &\equiv |N_o| \pmod 2 \end{aligned} \tag{5.3}$$

We treat each case next. Recall that $N = (N_D \cup N_N \cup N_A)$ and $N_o = (D_N \cup N_N \cup A_N)$.

(1) Assume $N_D = D_N = A_N = \emptyset$ and $|N_A| = 1$. Then, $(D_N = A_N = \emptyset) \to (N_o = N_N)$. So,

$$\begin{aligned} |N| &= |N_D \cup N_N \cup N_A| \\ &= |N_D| + |N_N| + |N_A| \quad && \text{(By } (D_o, N_o, A_o) \text{ is a partition)} \\ &= 0 + |N_o| + 1 && \text{(By } N_D = \emptyset,\ N_o = N_N \text{ and } |N_A| = 1) \\ &= |N_o| + 1 \end{aligned}$$

This is a contradiction to equation (5.3).

(2) Assume $A_N = N_A = N_D = \emptyset$ and $|D_N| = 1$. Then, $(N_D = N_A = \emptyset) \rightarrow (N = N_N)$. So,

$$\begin{aligned}
|N_o| &= |D_N \cup N_N \cup A_N| \\
&= |D_N| + |N_N| + |A_N| && \text{(By } (D, N, A) \text{ is a partition)} \\
&= 1 + |N| + 0 && \text{(By } |D_N| = 1,\ N = N_N \text{ and } A_N = \emptyset) \\
&= |N| + 1
\end{aligned}$$

This is a contradiction to equation (5.3).

(3) Assume $A_N = D_N = N_A = \emptyset$ and $|N_D| = 1$. Then, $(D_N = A_N = \emptyset) \rightarrow (N_o = N_N)$. So,

$$\begin{aligned}
|N| &= |N_D \cup N_N \cup N_A| \\
&= |N_D| + |N_N| + |N_A| && \text{(By } (D_o, N_o, A_o) \text{ is a partition)} \\
&= 1 + |N_o| + 0 && \text{(By } |N_D| = 1,\ N_o = N_N \text{ and } N_A = \emptyset) \\
&= |N_o| + 1
\end{aligned}$$

This is a contradiction to equation (5.3).

(4) Assume $N_D = N_A = D_N = \emptyset$ and $|A_N| = 1$. Then, $(N_D = N_A = \emptyset) \rightarrow (N = N_N)$. So,

$$\begin{aligned}
|N_o| &= |D_N \cup N_N \cup A_N| \\
&= |D_N| + |N_N| + |A_N| && \text{(By } (D, N, A) \text{ is a partition)} \\
&= 0 + |N| + 1 && \text{(By } D_N = \emptyset,\ N = N_N \text{ and } |A_N| = 1) \\
&= |N| + 1
\end{aligned}$$

This is a contradiction to equation (5.3).

□

Next, we show that with foreknowledge of an optimal partition, the R-rules can transform any feasible partition into the optimal partition.

Theorem 5.3.10. Let (D, N, A) be a feasible partition for $\alpha_i(T_v)$ and let (D_o, N_o, A_o) be a known optimal partition. There exists a sequence of R-rule applications that transforms (D, N, A) to (D_o, N_o, A_o). In addition, each element in $\{u_1, \ldots, u_k\}$ is moved at most once.

Proof. Given (D, N, A) and knowing an optimal partition (D_o, N_o, A_o), check if any R-rules may be applied on any 2 or 3 elements of $\{u_1, \ldots, u_k\}$ in a way that the rule brings the corresponding elements to the correct set with respect to (D_o, N_o, A_o). So, if an element is misplaced, then moving it once places it in the correct set with respect to (D_o, N_o, A_o), and the element will not be moved again. When none of the R-rules apply, we show that (D, N, A) has been transformed into (D_o, N_o, A_o) (denoted by $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$).

First, apply R_1, R_2 and R_3 whenever possible. Recall that the contents of (D, N, A) are changed as each R-rule is applied. As a result the sets X_Y for $X, Y \in \{D, N, A\}$ also change.

1. While D_A and A_D are both non-empty, apply R_1.

2. While D_N and N_D are both non-empty, apply R_2.

3. While N_A and A_N are both non-empty, apply R_3.

This results in at least one of the following 8 possible situations, depending on which of the two sets becomes empty in each of the three steps.

(1) $D_A = D_N = N_A = \emptyset$.

(2) $D_A = D_N = A_N = \emptyset$.

(3) $D_A = N_D = N_A = \emptyset$.

(4) $D_A = N_D = A_N = \emptyset$.

(5) $A_D = D_N = N_A = \emptyset$.

(6) $A_D = D_N = A_N = \emptyset$.

(7) $A_D = N_D = N_A = \emptyset$.

(8) $A_D = N_D = A_N = \emptyset$.

We treat the above 8 cases next.

(1) $D_A = D_N = N_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(2) $D_A = D_N = A_N = \emptyset$. At this point, the only applicable rules are R_5 and R_8. Apply R_5 repeatedly until either $N_A = \emptyset$ or $N_D = \emptyset$.

  (2.1) $N_A = \emptyset$. So, $D_A = D_N = N_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(2.2) $N_D = \emptyset$. The only applicable rule is R_8. Apply R_8 repeatedly until either $A_D = \emptyset$ or $|N_A| \leq 1$.

(2.2.1) $A_D = \emptyset$. So, $A_N = N_D = A_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(2.2.2) $N_A = \emptyset$. So, $D_A = D_N = N_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(2.2.3) $|N_A| = 1$. So, $D_N = A_N = N_D = \emptyset$ and $|N_A| = 1$. By Lemma 5.3.9 (Part 1), we cannot arrive at this situation.

(3) $D_A = N_D = N_A = \emptyset$. The only applicable rules are R_4 and R_{10}. Apply R_4 repeatedly until either $A_N = \emptyset$ or $D_N = \emptyset$.

(3.1) $A_N = \emptyset$ The only applicable rule is R_{10}. Apply R_{10} repeatedly until either $A_D = \emptyset$ or $|D_N| \leq 1$.

(3.1.1) $A_D = \emptyset$. So, $N_D = A_N = A_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(3.1.2) $D_N = \emptyset$. So, $D_A = N_A = D_N = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(3.1.3) $|D_N| = 1$. So, $N_D = N_A = A_N = \emptyset$ and $|D_N| = 1$. By Lemma 5.3.9 (Part 2), we cannot arrive at this situation.

(3.2) $D_N = \emptyset$. So, $D_A = N_A = D_N = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(4) $D_A = N_D = A_N = \emptyset$. The only applicable rules are R_6, R_8 and R_{10}. Apply R_6 repeatedly until either $A_D = \emptyset$, $D_N = \emptyset$ or $N_A = \emptyset$.

(4.1) $A_D = \emptyset$. So, $N_D = A_N = A_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(4.2) $D_N = \emptyset$. The only applicable rule is R_8. Apply R_8 repeatedly until either $A_D = \emptyset$ or $|N_A| \leq 1$.

(4.2.1) $A_D = \emptyset$. So, $N_D = A_N = A_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(4.2.2) $N_A = \emptyset$. So, $D_A = D_N = N_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(4.2.3) $|N_A| = 1$. So, $N_D = A_N = D_N = \emptyset$ and $|N_A| = 1$. By Lemma 5.3.9 (Part 1), we cannot arrive at this situation.

(4.3) $N_A = \emptyset$. The only applicable rule is R_{10}. Apply R_{10} repeatedly until either $A_D = \emptyset$ or $|D_N| \leq 1$.

(4.3.1) $A_D = \emptyset$. So, $N_D = A_N = A_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(4.3.2) $D_N = \emptyset$. So, $D_A = N_A = D_N = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(4.3.3) $|D_N| = 1$. So, $N_D = A_N = N_A = \emptyset$ and $|D_N| = 1$. By Lemma 5.3.9 (Part 2), we cannot arrive at this situation.

(5) $A_D = D_N = N_A = \emptyset$. The only applicable rules are R_7, R_9 and R_{11}. Apply R_7 repeatedly until either $A_N = \emptyset$, $N_D = \emptyset$ or $D_A = \emptyset$.

(5.1) $A_N = \emptyset$. The only applicable rule is R_9. Apply R_9 repeatedly until either $D_A = \emptyset$ or $|N_D| \leq 1$.

(5.1.1) $D_A = \emptyset$. So, $D_N = N_A = D_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(5.1.2) $N_D = \emptyset$. So, $A_D = A_N = N_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(5.1.3) $|N_D| = 1$. So, $D_N = N_A = A_N = \emptyset$ and $|N_D| = 1$. By Lemma 5.3.9 (Part 3), we cannot arrive at this situation.

(5.2) $N_D = \emptyset$. The only applicable rule is R_{11}. Apply R_{11} repeatedly until either $D_A = \emptyset$ or $|A_N| \leq 1$.

(5.2.1) $D_A = \emptyset$. So, $D_N = N_A = D_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(5.2.2) $A_N = \emptyset$. So, $A_D = N_D = A_N = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(5.2.3) $|A_N| = 1$. So, $D_N = N_A = N_D = \emptyset$ and $|A_N| = 1$. By Lemma 5.3.9 (Part 4), we cannot arrive at this situation.

(5.3) $D_A = \emptyset$. So, $D_N = N_A = D_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(6) $A_D = D_N = A_N = \emptyset$. The only applicable rules are R_5 and R_9. Apply R_5 repeatedly until either $N_A = \emptyset$ or $N_D = \emptyset$.

(6.1) $N_A = \emptyset$. The only applicable rule is R_9. Apply R_9 repeatedly until either $D_A = \emptyset$ or $|N_D| \leq 1$.

(6.1.1) $D_A = \emptyset$. So, $D_N = N_A = D_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(6.1.2) $N_D = \emptyset$. So, $A_D = A_N = N_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(6.1.3) $|N_D| = 1$. So, $D_N = A_N = N_A = \emptyset$ and $|N_D| = 1$. By Lemma 5.3.9 (Part 3), we cannot arrive at this situation.

(6.2) $N_D = \emptyset$. So, $A_D = A_N = N_D = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(7) $A_D = N_D = N_A = \emptyset$. The only applicable rules are R_4 and R_{11}. Apply R_4 repeatedly until either $A_N = \emptyset$ or $D_N = \emptyset$.

(7.1) $A_N = \emptyset$. So, $A_D = N_D = A_N = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(7.2) $D_N = \emptyset$. The only applicable rule is R_{11}. Apply R_{11} repeatedly until either $D_A = \emptyset$ or $|A_N| \leq 1$.

(7.2.1) $D_A = \emptyset$. So, $N_A = D_N = D_A = \emptyset$. By Lemma 5.3.8 condition 2, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(7.2.2) $A_N = \emptyset$. So, $A_D = N_D = A_N = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

(7.2.3) $|A_N| = 1$. So, $N_D = N_A = D_N = \emptyset$ and $|A_N| = 1$. By Lemma 5.3.9 (Part 4), we cannot arrive at this situation.

(8) $A_D = N_D = A_N = \emptyset$. By Lemma 5.3.8 condition 1, $(D, N, A) \Rightarrow^* (D_o, N_o, A_o)$.

This finishes the analyses for all 8 cases and completes the proof of the theorem. □

In the justification of Theorem 5.3.10, each element is moved at most once to the correct set with respect to (D_o, N_o, A_o). As a result, the order in which the R-rules are applied is not important.

Note that Theorem 5.3.10 requires knowing an optimal partition in advance and thus cannot be directly used in an algorithm that computes an optimal partition. But, we have obtained the following important corollary.

Corollary 5.3.11. Let (D, N, A) be an arbitrary feasible partition for $\alpha_i(T_v)$. There exists a sequence of R-rule applications that transforms (D, N, A) to an optimal partition, such that each element of $\{u_1, \ldots, u_k\}$ is moved at most once.

To conclude this section, we summarize the results developed so far with regard to obtaining an optimal partition of $\{u_1, \ldots, u_k\}$.

1. The E-rules are given in Definition 5.3.4. If we know the content of an arbitrary feasible partition (D, N, A) (for $\alpha_i(T_v)$) and the content of an optimal partition (D_o, N_o, A_o),

then we can apply the E-rules, one at a time, to (D, N, A) and transform it into (D_o, N_o, A_o).

2. Every element that is misplaced in the initial partition (D, N, A) will be moved, exactly once, to the correct set in (D_o, N_o, A_o). The order of E-rule applications is not important.

3. A set of R-rules are given in Definition 5.3.5. The R-rules are designed in a way so that applying any R-rule to a feasible partition for $\alpha_i(T_v)$ will result in another feasible partition for $\alpha_i(T_v)$.

4. In Theorem 5.3.10 and Corollary 5.3.11, we established that the R-rules are sufficient for transforming an arbitrary feasible partition into an optimal partition, given foreknowledge of such an optimal partition. Every element that is misplaced is moved exactly once, and the order of applications of R-rules is not important.

Section 5.4 will make use of the developments so far and create an $O(n \lg(\Delta))$ algorithm for solving `Rooted Secure Set` for trees.

5.4 An $O(n \lg(\Delta))$ algorithm

This section presents an $O(n \lg(\Delta))$ algorithm for solving `Rooted Secure Set` (Problem 5.1.1) on trees. We will make use of the feasibility preserving rule set (R-rules) given in

Definition 5.3.5 in the last section, and the results of Theorem 5.3.10 and Corollary 5.3.11. An outline of the section is presented below.

1. Develop a set of evaluation functions for the R-rules, reflecting the change in value (given in Definition 5.3.2) of a partition as a R-rule is applied to the partition.

2. Show that given an arbitrary feasible partition, there is a sequence of R-rule applications where successive applications strictly decrease the value of the partition, and eventually will transform it into an optimal partition.

3. Present an algorithm for solving `Rooted Secure Set` on trees, and analyze its time complexity.

In the next step, we develop evaluation functions for R_1 through R_{11}.

Definition 5.4.1. When an element is moved from X to Y, for $X, Y \in \{D, N, A\}$, the value of the partition as given in Definition 5.3.2 changes. Given a partition (D, N, A), define $e : (X \to Y) \to \mathbb{Z}$ to be the evaluation function of a single transformation $X \to Y$ applied to (D, N, A). When applying the rule $X \to Y$, any element in X may be moved to Y, and $e(X \to Y)$ is the minimum change that may result, among all choices in X. More specifically,

$$e(A \to D) = \min_{u \in A}\{\alpha_0(T_u)\}$$

$$e(D \to A) = \min_{u \in D}\{-\alpha_0(T_u)\}$$

$$e(A \to N) = \min_{u \in A}\{\alpha_1(T_u)\}$$

$$e(N \to A) = \min_{u \in N}\{-\alpha_1(T_u)\}$$

$$e(D \to N) = \min_{u \in D}\{-\alpha_0(T_u) + \alpha_1(T_u)\}$$

$$e(N \to D) = \min_{u \in N}\{-\alpha_1(T_u) + \alpha_0(T_u)\}$$

Let $r : \{R_1, \ldots, R_{11}\} \to \mathbb{Z}$ be the evaluation function of the set of transformations as indicated by $R_1, \ldots, R_{11}$ in Definition 5.3.5. More specifically,

$$r(R_1) = e(D \to A) + e(A \to D)$$

$$r(R_2) = e(D \to N) + e(N \to D)$$

$$r(R_3) = e(N \to A) + e(A \to N)$$

$$r(R_4) = e(A \to N) + e(D \to N)$$

$$r(R_5) = e(N \to A) + e(N \to D)$$

$$r(R_6) = e(A \to D) + e(D \to N) + e(N \to A)$$

$$r(R_7) = e(A \to N) + e(N \to D) + e(D \to A)$$

$$r(R_8) = e(A \to D) + e(N \to A) + e(N \to A)$$

$$r(R_9) = e(D \to A) + e(N \to D) + e(N \to D)$$

$$r(R_{10}) = e(A \to D) + e(D \to N) + e(D \to N)$$

$$r(R_{11}) = e(D \to A) + e(A \to N) + e(A \to N)$$

Note that the e functions, and hence the r functions, are dependent only upon the current partition.

Similar to Definition 5.3.5, R_5, R_8, R_9, R_{10} and R_{11} require special treatment. For example, in R_8, two distinct elements are moved from $N \to A$. Thus, in the evaluation of

$r(R_8)$, we select an element from A and move it to D, and two *distinct* elements from N and move them to A, in a way so that the value of the new partition is minimum. More precisely, $r(R_8) = \min\{\alpha_0(T_u) - \alpha_1(T_w) - \alpha_1(T_{w'}) : (u \in A), (w, w' \in N) \text{ and } (w \neq w')\}$; the sum of the smallest element in $\{\alpha_0(T_u) : u \in A\}$ and the smallest two elements in $\{-\alpha_1(T_w) : w \in N\}$. The case is similar for R_9, R_{10} and R_{11}.

In R_5, the element moved from N to A must be different from the one moved from N to D. To correctly evaluate $r(R_5)$, one must examine the result of $e(N \to A)$ and $e(N \to D)$. If the minimum values of both functions are achieved by the same element in N, then one of the movements needs to use a different element of N. The value of $r(R_5)$ is $\min\{-\alpha_1(T_w) - \alpha_1(T_{w'}) + \alpha_0(T_{w'}) : (w, w' \in N \text{ and } w \neq w')\}$.

These evaluations are not trivial, but nonetheless they can be computed in $O(\lg(k))$ time using priority queues implemented with a heap data structure (cf. [CLR01], Chapter 6), where k is the total number of elements being partitioned.

Recall from Corollary 5.3.11 that when given a feasible partition (D, N, A) for $\alpha_i(T_v)$, there exists a sequence of R-rule applications that transforms (D, N, A) to an optimal partition, such that every element in $\{u_1, u_2, \ldots, u_k\}$ is moved at most once. Lemma 5.4.2 improves further on this result, by asserting that successive R-rule applications in the sequence will always strictly decrease the value of the current partition (D, N, A).

Lemma 5.4.2. Let (D, N, A) be an arbitrary feasible partition for $\alpha_i(T_v)$. There exists a sequence, $R_{x_1}, R_{x_2}, \ldots, R_{x_\ell}$ of R-rule applications that transforms (D, N, A) to an optimal partition, where $r(R_{x_t}) < 0$ for $1 \leq t \leq \ell$ and each element in $\{u_1, \ldots, u_k\}$ is moved at most once.

Proof. By Corollary 5.3.11, there exists a sequence of R-rule applications that transforms (D, N, A) to an optimal partition, where each element of $\{u_1, \ldots, u_k\}$ is moved at most once. Let $R_{x_1}, R_{x_2}, \ldots, R_{x_\ell}$ be such a sequence with the minimum number of R-rule applications (minimum ℓ).

If there exists $t \in \{1, 2, \ldots, \ell\}$ where $r(R_{x_t}) \geq 0$, then removing R_{x_t} from the sequence will yield a new, shorter sequence that transforms (D, N, A) to another feasible solution that is no worse than the optimal (i.e., it transforms (D, N, A) to a different optimal solution). This is not possible since $R_{x_1}, R_{x_2}, \ldots, R_{x_\ell}$ is shortest, so $r(R_{x_t}) < 0$ for $1 \leq t \leq \ell$. $\square$

Next, we present Algorithm 5.4.3, which computes the value of $\alpha_i(T_v)$ ($i \in \{0, 1\}$) by finding an optimal partition for $\alpha_i(T_v)$. Then, Lemma 5.4.4 will apply Lemma 5.4.2 and prove the correctness of Algorithm 5.4.3. After that, Lemma 5.4.5 will assert that the while-loop in Step 4.2 of Algorithm 5.4.3 will execute at most once in a single iteration of the for-loop in Step 4, and so the while-loop may be replaced by an if-statement. Changing the while-loop to an if-statement yields Algorithm 5.4.6, our final result of this section.

Algorithm 5.4.3.

Input: $i \in \{0, 1\}$ and $v \in V(T)$.

Output: $\alpha_i(T_v)$.

1. If v is a leaf, then return 1. (This is the base case, and the algorithm exits.)

2. Let $\{u_1, u_2, \ldots, u_k\}$ be the children of v and compute $\{\alpha_0(T_{u_1}), \ldots, \alpha_0(T_{u_k})\}$ and $\{\alpha_1(T_{u_1}), \ldots, \alpha_1(T_{u_k})\}$ recursively.

3. (Initialize partition (D, N, A) for the appropriate α_i.)

   If $i = 0$, then $(D, N, A) \leftarrow (\{\}, \{\}, \{\})$, $s \leftarrow 1$.

   ($i = 0$, and (D, N, A) is an optimal partition of $\{\}$ for α_0.)

   else $(D, N, A) \leftarrow (\{\}, \{\}, \{u_1\})$, $s \leftarrow 2$.

   ($i = 1$, and (D, N, A) is an optimal partition of $\{u_1\}$ for α_1.)

4. For $j = s \ldots k$

   4.1. $N \leftarrow N \cup \{u_j\}$

   4.2. **While** $\min_{1 \leq x \leq 11} r(R_x) < 0$

   4.2.1. Let x' be such that $r(R_{x'}) = \min_{1 \leq x \leq 11} r(R_x)$

   4.2.2. $(D, N, A) \leftarrow R_{x'}(D, N, A)$ (Apply $R_{x'}$ on (D, N, A).)

5. Return $f(D, N, A)$. (see Definition 5.3.2.)

   ($f(D, N, A) = \alpha_i(T_v)$ since (D, N, A) is an optimal partition.)

Lemma 5.4.4. Algorithm 5.4.3 computes $\alpha_i(T_v)$ correctly.

Proof. The algorithm computes $\alpha_i(T_v)$ by finding an optimal partition of $\{u_1, \ldots, u_k\}$. In Step 3, (D, N, A) is initialized to an optimal partition of $\{\}$ if $i = 0$, and an optimal partition of $\{u_1\}$ if $i = 1$. Then, at the start of the j-th iteration of the for-loop in Step 4, (D, N, A) is an optimal partition of $\{u_1, \ldots, u_{j-1}\}$. By adding u_j to N in Step 4.1, (D, N, A) becomes a feasible partition for $\{u_1, \ldots, u_j\}$. Then, in Step 4.2, each iteration of the while-loop selects a R-rule and apply it on (D, N, A), in a way that $f(D, N, A)$ is decreased the most after the application. Note that in each iteration of the while-loop in Step 4.2, the value of the current partition (D, N, A) strictly decreases. Since the value of the partition is at least 1, it cannot decrease indefinitely. Thus, the while-loop must terminate.

Next, assume that at the end of the j-th iteration of the for-loop in Step 4, (D, N, A) is not an optimal partition of $\{u_1, \ldots, u_j\}$. By Lemma 5.4.2, there exists a sequence of R-rule applications that transforms (D, N, A) to an optimal partition, where for any rule $R_{x'}$ in that sequence, $r(R_{x'}) < 0$. That is, one or more rules may be applied to (D, N, A) and further decrease its value. This is a contradiction to the termination of the while-loop. So, at the end of the j-th iteration of the for-loop in Step 4, (D, N, A) is an optimal partition of $\{u_1, u_2, \ldots, u_j\}$, and when the for-loop terminates, (D, N, A) is an optimal partition of $\{u_1, u_2, \ldots, u_k\}$. □

Lemma 5.4.5. In Algorithm 5.4.3, the while-loop in Step 4.2 is executed at most once in each iteration of the for-loop in Step 4.

Proof. Recall from Definition 5.3.2 that the value of a partition (D, N, A) is $1 + \sum_{u \in D} \alpha_0(T_u) + \sum_{u \in N} \alpha_1(T_u)$. Consider the j-th iteration of the for-loop in Step 4. Let OPT_j be the value of an optimal partition for $\{u_1, \ldots, u_j\}$, let $(D_{j-1}, N_{j-1}, A_{j-1})$ be an arbitrary optimal partition of $\{u_1, \ldots, u_{j-1}\}$, and let (D, N, A) be $(D_{j-1}, N_{j-1} \cup \{u_j\}, A_{j-1})$. Note that (D, N, A) is a feasible partition of $\{u_1, \ldots, u_j\}$, which we constructed in Step 4.1, before the execution of the while-loop in Step 4.2. The value of (D, N, A) is

$$f(D, N, A) = f(D_{j-1}, N_{j-1}, A_{j-1}) + \alpha_1(T_{u_j}). \tag{5.4}$$

By Lemma 5.4.2, there exists a (possibly empty) sequence of R-rule applications $R_{x_1}, \ldots,$ R_{x_ℓ} that transforms (D, N, A) to an optimal partition, such that $r(R_{x_t}) < 0$ for $1 \leq t \leq \ell$. Let (D_j, N_j, A_j) be such an optimal partition. Two cases follow.

1. $u_j \in N_j$. In this case, we claim that $f(D, N, A) = f(D_j, N_j, A_j) = \mathrm{OPT}_j$, and the while-loop will execute for 0 iterations. Suppose not, and $f(D, N, A) > f(D_j, N_j, A_j)$. Let (D_t, N_t, A_t) be $(D_j, N_j - \{u_j\}, A_j)$. Then, (D_t, N_t, A_t) is a feasible partition of $\{u_1, \ldots,$ $u_{j-1}\}$ with value $f(D_t, N_t, A_t) = f(D_j, N_j, A_j) - \alpha_1(T_{u_j})$. Now,

$$\begin{aligned} f(D, N, A) &> f(D_j, N_j, A_j) \\ f(D_{j-1}, N_{j-1}, A_{j-1}) + \alpha_1(T_{u_j}) &> f(D_j, N_j, A_j) && \text{(By Eq. (5.4))} \\ f(D_{j-1}, N_{j-1}, A_{j-1}) &> f(D_j, N_j, A_j) - \alpha_1(T_{u_j}) \\ f(D_{j-1}, N_{j-1}, A_{j-1}) &> f(D_t, N_t, A_t) \end{aligned}$$

   We have obtained a feasible partition of $\{u_1, \ldots, u_{j-1}\}$, namely (D_t, N_t, A_t), with value $f(D_t, N_t, A_t) < f(D_{j-1}, N_{j-1}, A_{j-1})$. This is a contradiction since $(D_{j-1}, N_{j-1}, A_{j-1})$ is an optimal partition of $\{u_1, \ldots, u_{j-1}\}$.

2. $u_j \notin N_j$. Consider the sequence of applications $R_{x_1}, \ldots, R_{x_\ell}$ that transforms (D, N, A) to (D_j, N_j, A_j). We claim that $\ell = 1$. Assume $\ell > 1$. Recall that each element is moved at most once when applying $R_{x_1}, \ldots, R_{x_\ell}$, so the element u_j is moved at most once. Since the order of applications does not matter, let u_j be moved in the last rule R_{x_ℓ}. Then, R_{x_1} does not move u_j. Let (D_t, N_t, A_t) be a feasible partition of $\{u_1, u_2, \ldots, u_{j-1}\}$ constructed as follows. Apply R_{x_1} on (D, N, A), then remove u_j from N. This is valid since R_{x_1} does not move u_j, and removing $u_j \in N$ does not affect the feasibility of the new partition. Now,

$$\begin{aligned} f(D_t, N_t, A_t) &= f(D, N, A) + r(R_{x_1}) - \alpha_1(T_{u_j}) \\ &= f(D_{j-1}, N_{j-1}, A_{j-1}) + \alpha_1(T_{u_j}) + r(R_{x_1}) - \alpha_1(T_{u_j}) \quad \text{(By Eq. (5.4))} \\ &= f(D_{j-1}, N_{j-1}, A_{j-1}) + r(R_{x_1}) \\ &< f(D_{j-1}, N_{j-1}, A_{j-1}) \quad \text{(By } r(R_{x_1}) < 0\text{)} \end{aligned}$$

This is a contradiction, since $(D_{j-1}, N_{j-1}, A_{j-1})$ is an optimal partition for $\{u_1, \ldots, u_{j-1}\}$. So, $\ell = 1$, and there is a single rule R_{x_1} that may be applied to (D, N, A) and transform it to (D_j, N_j, A_j). A rule which decreases the value of (D, N, A) the most will then transform (D, N, A) to an optimal partition of $\{u_1, u_2, \ldots, u_j\}$ because $\min\limits_{1 \leq x \leq 11} r(R_x) \leq r(R_{x_1})$.

Thus, in both cases, the while-loop in Step 4.2 executes at most once in each iteration of the for-loop in Step 4. $\square$

Lemma 5.4.5 allows us to replace the while-loop in Step 4.2 of Algorithm 5.4.3 with an if-statement. This results in Algorithm 5.4.6 below.

Algorithm 5.4.6.

Input: $i \in \{0, 1\}$ and $v \in V(T)$.

Output: $\alpha_i(T_v)$.

1. If v is a leaf, then return 1. (This is the base case, and the algorithm exits.)

2. Let $\{u_1, u_2, \ldots, u_k\}$ be the children of v and compute $\{\alpha_0(T_{u_1}), \ldots, \alpha_0(T_{u_k})\}$ and $\{\alpha_1(T_{u_1}), \ldots, \alpha_1(T_{u_k})\}$ recursively.

3. (Initialize partition (D, N, A) for the appropriate α_i.)

   If $i = 0$, then $(D, N, A) \leftarrow (\{\}, \{\}, \{\})$, $s \leftarrow 1$.

   ($i = 0$, and (D, N, A) is an optimal partition of $\{\}$ for α_0.)

   else $(D, N, A) \leftarrow (\{\}, \{\}, \{u_1\})$, $s \leftarrow 2$.

   ($i = 1$, and (D, N, A) is an optimal partition of $\{u_1\}$ for α_1.)

4. For $j = s \ldots k$

   4.1. $N \leftarrow N \cup \{u_j\}$

   4.2. **If** $\min_{1 \le x \le 11} r(R_x) < 0$

      4.2.1. Let x' be such that $r(R_{x'}) = \min_{1 \le x \le 11} r(R_x)$

      4.2.2. $(D, N, A) \leftarrow R_{x'}(D, N, A)$ (Apply $R_{x'}$ on (D, N, A).)

5. Return $f(D, N, A)$. (see Definition 5.3.2.)

   ($f(D, N, A) = \alpha_i(T_v)$ since (D, N, A) is an optimal partition.)

Theorem 5.4.7. Algorithm 5.4.6 correctly solves `Rooted Secure Set` on a tree in $O(n \lg(\Delta))$ time.

Proof. Let T be a tree of order n with root r, where r must be included in a minimum rooted secure set. The cardinality of a minimum rooted secure set of T is $\alpha_1(T_r)$. By Lemmas 5.4.4 and 5.4.5, Algorithm 5.4.6 can correctly compute $\alpha_1(T_r)$.

When solving for $\alpha_1(T_r)$, the algorithm is called exactly $2n - 1$ times because we do not need to solve for $\alpha_0(T_r)$, but we will need to solve for both $\alpha_0(T_v)$ and $\alpha_1(T_v)$ for every vertex in $V(T) - \{r\}$. Let $v \in V(T)$ and $i \in \{0, 1\}$ be arbitrary, let $\{u_1, u_2, \ldots, u_k\}$ be the children of v and let $c_v = k$ (we use them interchangeably for convenience). Step 2 gathers the values $\{\alpha_0(T_{u_1}), \alpha_0(T_{u_2}), \ldots, \alpha_0(T_{u_k})\}$ and $\{\alpha_1(T_{u_1}), \alpha_1(T_{u_2}), \ldots, \alpha_1(T_{u_k})\}$ recursively, which takes $O(c_v)$ time. Note that this only includes the time for gathering the results and does not include the time for computing each $\alpha_i(T_{u_j})$. The for-loop in Step 4 will be executed at most c_v times, and in each iteration, the algorithm evaluates all the R-rules according to Definition 5.4.1. The evaluation of a R-rule involves finding the smallest and second smallest elements of two or three priority queues. With appropriate data structures (e.g., priority queues implemented with binary heap, cf. [CLR01], Chapter 6), the evaluations can be done in $O(\lg(c_v))$ time. Note that we may also maintain the partition (D, N, A) with balanced trees (cf. [Knu98], 6.2.3), which take $O(\lg(c_v))$ time per modification and access.

Then, the overall running time of Algorithm 5.4.6 for computing $\alpha_1(T_r)$ is proportional to $\sum_{v \in V(T)} (c_v \times \lg(c_v)) \leq \sum_{v \in V(T)} (c_v \times \lg(\Delta)) = (n-1) \times \lg(\Delta) \in O(n \lg(\Delta))$. $\square$

This concludes the development of the $O(n \lg(\Delta))$ algorithm for solving `Rooted Secure Set` on trees. Several extensions of the results presented in this chapter are possible. In the next chapter, we will employ ideas similar to the ones presented in Sections 5.1 and 5.2 and develop an $O(n\Delta)$ algorithm for computing the global security number of a tree. Other interesting questions related to `Rooted Secure Set` will be presented in Chapter 9.

CHAPTER 6

GLOBAL SECURE SETS OF TREES

This chapter discusses the global security numbers of trees. Recall from Section 1.3 that a global secure set of a graph G is a dominating set and a secure set of G. The global security number of G is the cardinality of a minimum global secure set of G, denoted $\gamma_s(G)$. Section 6.1 presents an $O(n\Delta)$ algorithm for finding the global security number of a tree. The algorithm uses Wimer's method ([WHL85, Wim87]) and employs ideas developed in Sections 5.1 and 5.2. Section 6.2 presents upper and lower bounds on $\gamma_s(T)$ for a tree T. Finally, Section 6.3 presents results on the global security number of a connected graph in relation to that of its spanning trees.

6.1 An $O(n\Delta)$ algorithm

In Chapter 5, the problem `Rooted Secure Set` (Problem 5.1.1) is introduced, and algorithms are provided for finding the cardinality of a minimum rooted secure set of a tree. These algorithms have polynomial time complexity. As mentioned in Observation 5.1.2, `Rooted Secure Set` is as difficult as `Secure Set` (Problem 1.4.1) in the context of existence of polynomial solutions. The key factor that enabled a polynomial solution for trees is Lemma 5.1.3, which states that every attacker can attack exactly one vertex of a minimum rooted secure set in a tree. This observation greatly simplifies the situation because when there is only one possible attack, an algorithm only needs to construct a feasible defense for that attack to ensure security.

When finding a minimum global secure set of a tree, the situation is similar but more complex. As shown in Figures 6.1, 6.2 and 6.3, a connected minimum global secure set may not always exist for some trees. So, Lemma 5.1.3 does not apply to minimum global secure sets. But, as stated in Observation 6.1.1 below, Lemma 5.1.3 can be applied to each connected component of a global secure set.

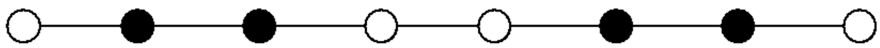

Figure 6.1: The path P_8 and a minimum global secure set marked in black. All minimum global secure sets of P_8 are disconnected.

Observation 6.1.1. Let S be a dominating set of a tree T. Let $C_1, C_2, \ldots, C_k$ be the connected components of $T[S]$. Then, S is a global secure set if and only if each C_i is a secure set of T. Furthermore, since each $V(C_i)$ is a connected subset of vertices, Lemma 5.1.3 applies to C_i and there is an unique attack upon C_i, $1 \leq i \leq k$.

By Observation 6.1.1, when finding a minimum global secure set of a tree, an algorithm can ensure the security of the set if it can identify a feasible defense for each connected component of the set individually. Similar to the algorithm given in Section 5.2, we present

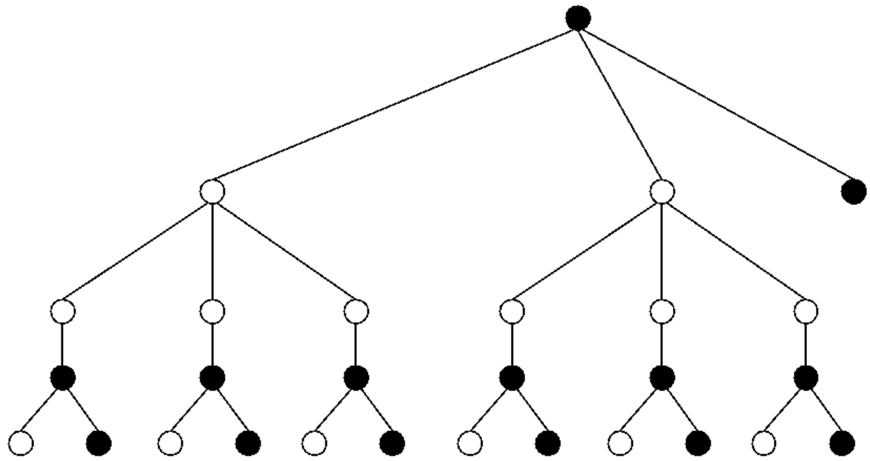

Figure 6.2: A tree and a minimum global secure set marked in black. All minimum global secure sets of this tree are disconnected.

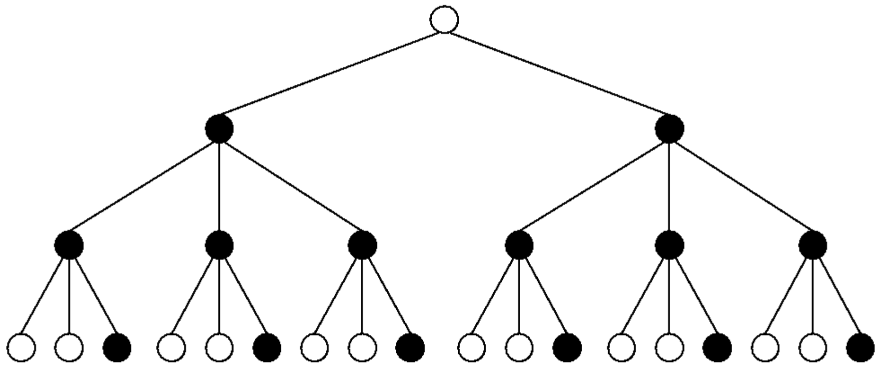

Figure 6.3: A tree and a minimum global secure set marked in black. All minimum global secure sets of this tree are disconnected.

an $O(n\Delta)$ algorithm for finding the cardinality of a minimum global secure set (i.e., the global security number) of a tree.

Let T be a tree and let r be an *arbitrary* vertex of T. Consider T as a rooted tree with root r. Then, for $v \in V(T)$, let T_v denote the subtree of T rooted at v with respect to r, and let c_v denote the number of children of v in T_v. When $v \neq r$, let p_v denote the parent of v, with respect to r. If S is a subset of $V(T)$, then let $S_v = S \cap V(T_v)$ denote the vertices of T_v that belong to S. Following Wimer's method, associate with each vertex $v \in V(T)$ a set of *auxiliary invariants*, $\alpha(T_v) = \{\alpha_i(T_v) : -c_v \leq i \leq c_v\}$, $\beta(T_v) = \{\beta_i(T_v) : -(c_v+1) \leq i \leq (c_v+1)\}$ and $\sigma(T_v) = \{\sigma_0(T_v), \sigma_1(T_v)\}$. Each of $\alpha(T_v)$, $\beta(T_v)$ and $\sigma(T_v)$ is an array of integers. The semantics of each entry of the arrays is given in Definition 6.1.3.

Definition 6.1.2. Let T be a tree and let S be a global secure set of T. For $x \in S$, let $A_x = N[x] - S$ be the possible attackers of x, and let D_x be the set of defenders of x in a defense $\mathscr{D}$.

Definition 6.1.2 is similar to Definition 5.1.4, with one distinction. Since a global secure set S may not be connected, there is no unique attack for S. For $x \in S$, let C_x be the connected component of $T[S]$ that contains x. To ensure S is a secure set, an algorithm must verify that each C_x is a secure set in T. Since C_x is connected, it has a unique attack $\mathscr{A}$, and the attackers of x in $\mathscr{A}$ is the set $A_x = (N[x] - C_x) = (N[x] - S)$. If x and y are two vertices of S which belong to different connected components in $T[S]$, $A_x \cap A_y$ may be non-empty. But, since an algorithm verifies the security of C_x and C_y separately, we may still let $A_x = (N[x] - C_x) = (N[x] - S)$ and $A_y = (N[y] - C_y) = (N[y] - S)$. So, A_x is *not* the attackers of x in any specific attack on S, but rather the attackers of x in the unique attack on C_x.

Definition 6.1.3.

1. The entry $\alpha_i(T_v)$ is an integer representing the cardinality of a minimum set S_v such that

   (i) $v \in S_v$,

   (ii) $|D_x| \geq |A_x|$ for all $x \in S_v - \{v\}$,

   (iii) $|D_v| + i \geq |A_v|$,

   (iv) $v \notin D_x$ for all $x \in S_v$, and

   (v) S_v dominates T_v.

   That is, $v \in S_v$ and S_v is a dominating set of T_v. Every vertex in $S_v - \{v\}$ is protected by a sufficient number of defenders. If $i > 0$, v has at most i more attackers than defenders, and will be protected if it receives an additional i defenders. If $i \leq 0$, v is currently protected, and will remain protected if it receives an additional $|i|$ attackers. Furthermore, v has *not* been assigned to a vertex for which v should defend. Thus, an $\alpha_i(T_v)$ configuration can be used in the composition of a larger solution, where v is free to defend another vertex.

2. The entry $\beta_i(T_v)$ is an integer representing the cardinality of a minimum set S_v such that

   (i) $v \in S_v$,

   (ii) $|D_x| \geq |A_x|$ for all $x \in S_v - \{v\}$,

(iii) $|D_v| + i \geq |A_v|$,

(iv) $v \in D_x$ for a $x \in S_v$, and

(v) S_v dominates T_v.

A $\beta_i(T_v)$ configuration is similar to an $\alpha_i(T_v)$ configuration, with the only distinction that v has been assigned to a vertex for which v defends. When a $\beta_i(T_v)$ configuration is used in the composition of a larger solution, v cannot be used to defend another vertex. Note that $\beta_0(T_v)$ is the minimum cardinality among the global secure sets of T_v that contain v.

3. The entry $\sigma_0(T_v)$ is an integer representing the cardinality of a minimum set S_v such that

   (i) $v \notin S_v$,

   (ii) $|D_x| \geq |A_x|$ for all $x \in S_v$, and

   (iii) S_v dominates T_v.

   That is, $v \notin S_v$ and S_v is a global secure set of T_v. In this case, v is dominated by one of its children in S_v. So, $\sigma_0(T_v)$ is the minimum cardinality among the global secure sets of T_v that do not contain v.

4. The entry $\sigma_1(T_v)$ is an integer representing the cardinality of a minimum set S_v such that

   (i) $v \notin S_v$,

(ii) $|D_x| \geq |A_x|$ for all $x \in S_v$, and

(iii) S_v dominates $V(T_v) - \{v\}$, but not v.

So, $v \notin S_v$, S_v is a secure set in T_v, and S_v dominates every vertex of T_v except for v. When a $\sigma_1(T_v)$ configuration is used in the composition of a larger solution, v must be dominated by one of its neighbors in the larger tree if the resulting set is to be a dominating set.

This completes Definition 6.1.3. The global security number of T is $\min\{\sigma_0(T_r), \beta_0(T_r)\}$. A $\sigma_0(T_r)$ configuration corresponds to a minimum set $S \subseteq V(T)$ such that S is a global secure set of T and $r \notin S$. A $\beta_0(T_r)$ configuration corresponds to a minimum set $S \subseteq V(T)$ such that S is a global secure set of T and $r \in S$.

The auxiliary invariants proposed above is noticeably more complex than the one used for computing the cardinality of a minimum rooted secure set of a tree, given in Section 5.2 (Definition 5.2.2). Only one array is required for computing the cardinality of a minimum rooted secure set of a tree. We give two reasons why a simpler formulation for the global security number of a tree may not be possible.

1. Let S be a minimum global secure set of a tree T. If a vertex $v \in V(T)$ is not in S and is not a leaf, then the subtree T_v must contain vertices in S, for otherwise $V(T_v) - \{v\}$ is not dominated. In other words, if $v \notin S$ and T_v contains more than one vertex, S_v must not be empty. This is unlike the case of rooted secure sets. If a vertex v is not in

a minimum rooted secure set, then T_v does not contain any vertices in the set, since a minimum rooted secure set is always a connected set that contains the root.

Since S_v may not be empty when $v \notin S$, we must introduce auxiliary invariants that record the possible cardinalities of S_v in such situations. The values $\{\sigma_0(T_v), \sigma_1(T_v)\}$ are designed for this purpose. In particular, if $v \notin S$, then the subtree T_v is either dominated by S_v ($\sigma_0(T_v)$), or the vertices in $(V(T_v) - \{v\})$ are dominated by S_v and v is dominated by its neighbors outside T_v ($\sigma_1(T_v)$).

2. In Lemma 5.1.5 (and the remark that followed it), we showed that in a feasible defense of a minimum rooted secure set, a parent will never defend any of its children. As a result, each vertex in the set either defends its parent or defends itself. The result of Lemma 5.1.5 cannot be (trivially) extended to global secure sets of trees. Consider Figure 6.4 as an example. The minimum global secure set shown in Figure 6.4 is connected and it has one unique attack. In a feasible defense for this attack, the root r must defend its child u. In fact, every minimum global secure set of the tree shown in Figure 6.4 must include r and have r defend one of its children. Note that we may decide to let t be the root instead, in which case no parent will have to defend a child. But, in the lack of any intelligent ways to decide which vertex should be the root, an algorithm that selects a root arbitrarily has to consider cases where a parent may defend a child. The array $\beta(T_v)$ is designed for this purpose. This will be discussed further when a recursive formulation for $\beta(T_v)$ is given.

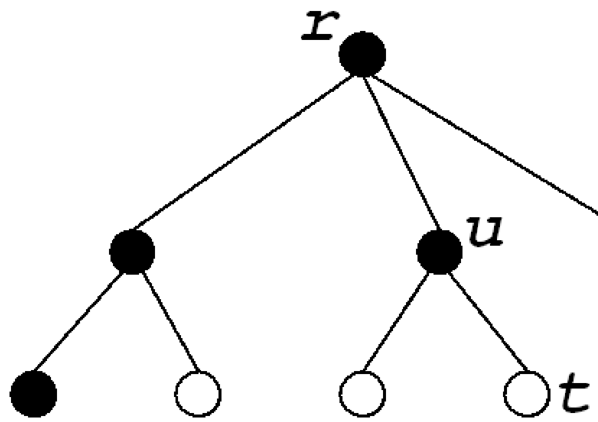

Figure 6.4: A tree and a minimum global secure set marked in black. If r is selected as the root, then in every minimum global secure set, a parent must defend one of its children in every feasible defense.

Recall from Definition 5.2.1 that if T_1 is a rooted tree with root r_1 and T_2 is a rooted tree with root r_2, then $T_1 \circ T_2$ is a rooted tree constructed by taking the disjoint union of T_1 and T_2, then adding an edge between r_1 and r_2, and designating r_1 to be the root of $T_1 \circ T_2$. As seen in Section 5.2, when applying Wimer's method it suffices to construct the auxiliary invariants for $T_1 \circ T_2$, given the auxiliary invariants of T_1 and T_2. Then, the auxiliary invariants for T_v ($v \in V(T)$) may be computed by first considering $\{v\}$ as a rooted tree with a single vertex v and attaching each of its child subtrees $\{T_{u_1}, T_{u_2}, \ldots, T_{u_k}\}$ to v, one at a time, using the $\circ$ operator.

Next, we present the base case values for each of the auxiliary invariants, and the recursive formulation for the auxiliary invariants of $T_1 \circ T_2$ using the auxiliary invariants of T_1 and T_2.

Let T_1 be a rooted tree of order one. Then, the initial (base case) values for each auxiliary invariant associated with T_1 is as follows.

1. $\alpha_i(T_1) = \begin{cases} 1 & \text{if } i \geq 0 \\ \infty & \text{otherwise} \end{cases}$

2. $\beta_i(T_1) = \begin{cases} 1 & \text{if } i \geq -1 \\ \infty & \text{otherwise} \end{cases}$

3. $\sigma_0(T_1) = \infty$.

4. $\sigma_1(T_1) = 0$.

In the base case, let r_1 be the only vertex in T_1. The vertex r_1 is the root of T_1. Let S be a subset of $V(T_1)$ and, if $r_1 \in S$, let $|D_1|$ and $|A_1|$ be the number of defenders and attackers of r_1, respectively. Note that when $r_1 \in S$, $|A_1| = 0$. In an $\alpha_i(T_1)$ configuration, $r_1 \in S$ and r_1 is not utilized in any defense. In particular, r_1 is not defending itself. So, $|D_1| = |A_1| = 0$, and the condition $|D_1| + i \geq |A_1|$ is satisfied whenever $i \geq 0$. Thus, if $i \geq 0$ then $\alpha_i(T_1) = 1$ (since $S = \{r_1\}$), and otherwise $\alpha_i(T_1) = \infty$ since there is no valid $\alpha_i(T_1)$ configuration if $i < 0$. In a $\beta_i(T_1)$ configuration, $r_1 \in S$ and r_1 is utilized in a defense. Since r_1 is the only vertex in S, r_1 defends itself, and so $|D_1| = 1$ and $|A_1| = 0$. Then, the condition $|D_1| + i \geq |A_1|$ is satisfied whenever $(1 + i) \geq 0$, or $i \geq -1$. So, $\beta_i(T_1) = 1$ if $i \geq -1$, and otherwise $\beta_i(T_1) = \infty$. In a $\sigma_0(T_1)$ configuration, $r_1 \notin S$ and S is empty, so $\sigma_0(T_1) = \infty$ since a valid $\sigma_0(T_1)$ configuration requires S dominates T_1. Finally, in a $\sigma_1(T_1)$ configuration, $r_1 \notin S$, but in this case $(V(T_1) - \{r_1\}) = \emptyset$ is considered dominated, and so $\sigma_1(T_1) = 0$ (since $S = \emptyset$).

Next, suppose T_1 and T_2 are two rooted trees with known auxiliary invariants values. Let $T_{12} = T_1 \circ T_2$. The auxiliary invariants for T_{12} may be computed as follows.

1. $$\alpha_i(T_{12}) = \min \begin{cases} \alpha_{i-1}(T_1) + \min\{\sigma_0(T_2), \sigma_1(T_2)\} & \text{(Type A.1)} \\ \alpha_i(T_1) + \beta_0(T_2) & \text{(Type A.2)} \\ \alpha_{i+1}(T_1) + \alpha_0(T_2) & \text{(Type A.3)} \end{cases}$$

2. $$\beta_i(T_{12}) = \min \begin{cases} \beta_{i-1}(T_1) + \min\{\sigma_0(T_2), \sigma_1(T_2)\} & \text{(Type B.1)} \\ \beta_i(T_1) + \beta_0(T_2) & \text{(Type B.2)} \\ \beta_{i+1}(T_1) + \alpha_0(T_2) & \text{(Type B.3)} \\ \alpha_i(T_1) + \beta_1(T_2) & \text{(Type B.4)} \end{cases}$$

3. $$\sigma_0(T_{12}) = \min \begin{cases} \sigma_0(T_1) + \sigma_0(T_2) & \text{(Type S.1)} \\ \min\{\sigma_0(T_1), \sigma_1(T_1)\} + \beta_{-1}(T_2) & \text{(Type S.2)} \end{cases}$$

4. $$\sigma_1(T_{12}) = \sigma_1(T_1) + \sigma_0(T_2) \qquad \text{(Type S.3)}$$

We first provide an explanation of the recursive formulations given above, then present the pseudocode of the algorithm. Let r_1 be the root of T_1 and let r_2 be the root of T_2. Note that r_1 is also the root of T_{12}. Let S be a subset of $V(T_{12})$. If $r_1 \in S$, then let $|D_1|$ and $|A_1|$ be, respectively, the number of defenders and attackers of r_1 among the vertices of T_1, and let $|D_{12}|$ and $|A_{12}|$ be, respectively, the number of defenders and attackers of r_1 among the vertices of T_{12}. If $r_2 \in S$, then let $|D_2|$ and $|A_2|$ be, respectively, the number of defenders and attackers of r_2 among the vertices of T_2.

In an $\alpha_i(T_{12})$ configuration, $r_1 \in S$ and r_1 is not defending any vertex of T_{12}. So, in T_1, $r_1 \in S$ and r_1 does not defend any vertex in T_1. That is, we must use an $\alpha_j(T_1)$ configuration for T_1. Then, consider the role of r_2 in the new tree T_{12}.

A.1 $r_2 \notin S$. In this case, r_2 is an attacker of r_1. Since $r_1 \in S$, r_2 may be dominated by r_1 in T_{12} if it is not already dominated within T_2. Thus, in T_2 we may use either a $\sigma_0(T_2)$ or $\sigma_1(T_2)$ configuration. Here, $|D_{12}| = |D_1|$ and $|A_{12}| = |A_1| + 1$. In an $\alpha_i(T_{12})$ configuration, $|D_{12}| + i \geq |A_{12}|$, so $|D_1| + i \geq |A_1| + 1$, or $|D_1| + (i-1) \geq |A_1|$. Thus, in T_1 we use an $\alpha_{i-1}(T_1)$ configuration.

A.2 $r_2 \in S$, but r_2 is not defending r_1. Since, in an $\alpha_i(T_{12})$ configuration, r_1 is not defending any vertex, r_1 is not defending r_2. In this case, r_1 and r_2 are *neutral* with respect to each other. The vertex r_2 is assigned a vertex in T_2 to defend. So, we use a $\beta_0(T_2)$ configuration for T_2. In other words, $S \cap V(T_2)$ is a global secure set of T_2 for which $r_2 \in S$. Here, $|D_{12}| = |D_1|$ and $|A_{12}| = |A_1|$, so we use an $\alpha_i(T_1)$ configuration for T_1.

A.3 $r_2 \in S$ and r_2 defends r_1. In this case, r_2 is a defender of r_1. So, r_2 does not defend any vertex in T_2 (not even itself), and we use an $\alpha_0(T_2)$ configuration for T_2. In other words, $S \cap V(T_2)$ is a global secure set of T_2, for which r_2 is in S, but is not utilized in a defense within T_2. Then, r_2 may be used in T_{12} to defend r_1. Here, $|D_{12}| = |D_1| + 1$ and $|A_{12}| = |A_1|$. In an $\alpha_i(T_{12})$ configuration, $|D_{12}| + i \geq |A_{12}|$, so $|D_1| + (i+1) \geq |A_1|$. Thus, we use an $\alpha_{i+1}(T_1)$ configuration for T_1.

In a $\beta_i(T_{12})$ configuration, $r_1 \in S$ and r_1 is defending a vertex of T_{12}. If r_1 is not defending r_2 in T_{12}, then r_1 is defending a vertex within T_1. In that case, we use a $\beta_j(T_1)$ configuration for T_1, and the scenarios that may follow are similar to those given for $\alpha_i(T_{12})$. That is, the compositions B.1, B.2 and B.3 correspond to A.1, A.2 and A.3 respectively, with the distinction that r_1 is defending a vertex in T_1 in a β configuration, but is not defending any vertex in an α configuration. An interesting case is when r_1 defends r_2 in T_{12}. This case is handled by composition B.4. In B.4, r_1 is a defender of r_2, and r_1 is not defending any vertex within T_1. Thus, we use an $\alpha_i(T_1)$ configuration in T_1. Since r_1 is an additional defender of r_2 outside T_2, from within T_2 we require $|D_2| + 1 \geq |A_2|$. So, we use a $\beta_1(T_2)$ configuration for T_2. Composition B.4 is used in the case where a parent may defend one of its children, and is the primary function of the β array.

In a $\sigma_0(T_{12})$ configuration, $r_1 \notin S$, and S dominates T_{12}. There are two cases based on the role of r_2 in T_{12}.

S.1 $r_2 \notin S$. In this case, both r_1 and r_2 are not in the set. So, r_1 must be dominated by a vertex in T_1, and r_2 must be dominated by a vertex in T_2. Thus, we use a $\sigma_0(T_1)$ configuration for T_1 and a $\sigma_0(T_2)$ configuration for T_2.

S.2 $r_2 \in S$. In this case, r_1 is an attacker of r_2. So, in T_2, we must require $|D_2| \geq |A_2|+1$, or $|D_2| - 1 \geq |A_2|$. Thus, we use a $\beta_{-1}(T_2)$ configuration for T_2. In T_{12}, r_1 is dominated by r_2 if it is not already dominated in T_1. So, for T_1, either a $\sigma_0(T_1)$ or $\sigma_1(T_1)$ configuration is valid.

Finally, in a $\sigma_1(T_{12})$ configuration, r_1 is not in S and is not dominated by S. So, r_1 must not be dominated within T_1, and r_2 must not be in S. Thus, we use a $\sigma_1(T_1)$ configuration for T_1. Since $r_1 \notin S$, r_2 must be dominated from within T_2. Thus, we use a $\sigma_0(T_2)$ configuration for T_2.

This completes the explanation of the recursive formulas. Note that in the formulation, for subtree T_2 the only referenced values are $\{\alpha_0(T_2), \beta_{-1}(T_2), \beta_0(T_2), \beta_1(T_2), \sigma_0(T_2), \sigma_1(T_2)\}$. The global security number of a tree T is $\min\{\sigma_0(T_r), \beta_0(T_r)\}$, where r is an arbitrarily selected root. Thus, for a vertex $v \in V(T)$, an algorithm does not need to compute all the entries of $\alpha(T_v)$ and $\beta(T_v)$, but only those referenced during the construction of its parent's auxiliary invariants, namely $\{\alpha_0(T_v), \beta_{-1}(T_v), \beta_0(T_v), \beta_1(T_v), \sigma_0(T_v), \sigma_1(T_v)\}$. But, similar to Algorithm 5.2.3 given in Section 5.2, other entries of α and β must be kept track of during the construction of T_v (i.e., when attaching each child subtree of v to v), whereas only those aforementioned entries are necessary in the final result after T_v is completely constructed. The ranges for $\alpha(T_v)$ and $\beta(T_v)$ are then derived from the ranges being referenced during the construction of T_v, which are the values for which the final result $\{\alpha_0(T_v), \beta_{-1}(T_v), \beta_0(T_v), \beta_1(T_v)\}$ depends on.

The pseudocode of the algorithm, implemented according to the developments so far, is given on the next page.

Algorithm 6.1.4.

Input: A rooted tree T_v.

Output: $\{\alpha_0(T_v), \beta_{-1}(T_v), \beta_0(T_v), \beta_1(T_v), \sigma_0(T_v), \sigma_1(T_v)\}$

`GlobalSecure`(T_v)

1. Let c_v be the number of children of v.

2. (Initialize T_1)

   2.1 For $i = -c_v \dots c_v$
   If $i \geq 0$, then $\alpha_i(T_1) = 1$,
   else $\alpha_i(T_1) = \infty$.

   2.2 For $i = -(c_v + 1) \dots (c_v + 1)$
   If $i \geq -1$, then $\beta_i(T_1) = 1$,
   else $\beta_i(T_1) = \infty$.

   2.3 $\sigma_0(T_1) = \infty$.

   2.4 $\sigma_1(T_1) = 0$.

3. Let $\{u_1, u_2, \dots u_k\}$ be the children of v. (i.e., $k = c_v$)

4. For $j = 1 \dots c_v$,

   4.1 $\{\alpha_0(T_2), \beta_{-1}(T_2), \beta_0(T_2), \beta_1(T_2), \sigma_0(T_2), \sigma_1(T_2)\} \leftarrow$ `GlobalSecure`(T_{u_j}).

   (Steps 4.2 to 4.5 attach T_{u_j} to v. The new tree is T_{12}.)

   4.2 For $i = (j - c_v) \dots (c_v - j)$

   $$\alpha_i(T_{12}) \leftarrow \min\{\alpha_{i-1}(T_1) + \min\{\sigma_0(T_2), \sigma_1(T_2)\}, \alpha_i(T_1) + \beta_0(T_2), \alpha_{i+1}(T_1) + \alpha_0(T_2)\}$$

4.3 For $i = j - (c_v + 1) \ldots (c_v + 1) - j$

$$\beta_i(T_{12}) \leftarrow \min\{\beta_{i-1}(T_1) + \min\{\sigma_0(T_2), \sigma_1(T_2)\}, \beta_i(T_1) + \beta_0(T_2), \beta_{i+1}(T_1) + \alpha_0(T_2),$$
$$\alpha_i(T_1) + \beta_1(T_2)\}$$

4.4 $\sigma_0(T_{12}) \leftarrow \min\{\sigma_0(T_1) + \sigma_0(T_2), \min\{\sigma_0(T_1), \sigma_1(T_1)\} + \beta_{-1}(T_2)\}$

4.5 $\sigma_1(T_{12}) \leftarrow \sigma_1(T_1) + \sigma_0(T_2)$

(Steps 4.6 to 4.8 copy T_{12} back into T_1.)

4.6 For $i = (j - c_v) \ldots (c_v - j)$

$$\alpha_i(T_1) \leftarrow \alpha_i(T_{12})$$

4.7 For $i = j - (c_v + 1) \ldots (c_v + 1) - j$

$$\beta_i(T_1) \leftarrow \beta_i(T_{12})$$

4.8 $\{\sigma_0(T_1), \sigma_1(T_1)\} \leftarrow \{\sigma_0(T_{12}), \sigma_1(T_{12})\}$

5. Return $\{\alpha_0(T_1), \beta_{-1}(T_1), \beta_0(T_1), \beta_1(T_1), \sigma_0(T_1), \sigma_1(T_1)\}$

Lemma 6.1.5. Algorithm 6.1.4 has time complexity $O(n\Delta)$, where Δ is the maximum degree of the input tree T.

Proof. Let T be a tree with maximum degree Δ. Let r be an arbitrarily selected root of T. For each $v \in V(T)$, the algorithm is invoked with argument T_v. In Step 2, the algorithm performs $O(c_v)$ work, in particular at Steps 2.1 and 2.2. Then, Step 4 is executed exactly c_v times, and in each iteration of Step 4, the algorithm performs $O(c_v)$ work, in particular at Steps 4.2, 4.3, 4.6 and 4.7. Thus, the algorithm performs a total of $O\left((c_v)^2\right)$ work for each vertex $v \in V(T)$. The overall runtime of the algorithm for solving $\texttt{GlobalSecure}(T_r)$ is then proportional to $\sum_{v \in V(T)} (c_v)^2 \leq \sum_{v \in V(T)} (c_v)\Delta = (n-1)\Delta \in O(n\Delta)$. □

6.2 Upper and lower bounds

In this section, we present upper and lower bounds on the global security number of a tree. A general lower bound on the global security number of an arbitrary graph in terms of its order may be obtained using Theorem 1.2.3.

Lemma 6.2.1. Let G be a graph of order n. Then, $\gamma_s(G) \geq \lceil n/2 \rceil$.

Proof. Let S be a minimum global secure set of G. Since S is a dominating set, $N[S] = V(G)$ and $|S| + |N[S] - S| = |V(G)| = n$. So, $|N[S] - S| = n - |S|$. Since S is a secure set, by Theorem 1.2.3, $|S| = |N[S] \cap S| \geq |N[S] - S| = n - |S|$. Thus, $2|S| \geq n$ and $|S| \geq \lceil n/2 \rceil$. □

The lower bound given in Lemma 6.2.1 is a sharp lower bound for trees. For example, in Chapter 7 we show that $\gamma_s(P_n) = \lceil n/2 \rceil$ (Theorem 7.2.1), where P_n denotes a path on $n \geq 2$ vertices. There are many other examples for the sharpness of the lower bound given in Lemma 6.2.1. Figures 6.2, 6.3 and 6.4 show three such examples.

The remainder of this section studies an upper bound on the global security number of a tree. Lemma 6.2.8 shows that $\gamma_s(T) \leq 2n/3$ for a tree T of order $n \geq 2$. First, Definition 6.2.2 presents two classes of rooted trees, $\mathbb{T}_1$ and $\mathbb{T}_2$. A minimum global secure set of any rooted tree in these two classes can be used to construct global secure sets of larger trees. This is discussed in more detail in Lemma 6.2.3.

Definition 6.2.2. Let T be a rooted tree with root r.

$\mathbb{T}_1$: T is in the class $\mathbb{T}_1$ if there exist two minimum global secure sets of T, denoted S_1 and S_2, such that $r \in S_1$ and $r \notin S_2$.

$\mathbb{T}_2$: T is in the class $\mathbb{T}_2$ if there exists a minimum global secure set of T, denoted S, such that $r \in S$, and S is a global secure set of $T \cup \{r\ell\}$, where ℓ is a new vertex added to T. Note that S is *not* necessarily a minimum global secure set of $T \cup \{r\ell\}$.

The set of rooted trees that belong to either $\mathbb{T}_1$ or $\mathbb{T}_2$ is denoted $\mathbb{T}_1 \cup \mathbb{T}_2$.

If T is in the class $\mathbb{T}_1$, then there are two minimum global secure sets of T such that one set includes r and the other excludes r. With reference to the auxiliary invariants defined in Section 6.1 (Definition 6.1.3), T is in $\mathbb{T}_1$ whenever $\sigma_0(T_r) = \beta_0(T_r) = \gamma_s(T)$. If T is in the class $\mathbb{T}_2$, then there is a minimum global secure set of T such that the set contains r, and the set remains a secure set even if r receives an additional attacker. With reference to the auxiliary invariants in Section 6.1, T is in $\mathbb{T}_2$ whenever $\beta_{-1}(T_r) = \gamma_s(T)$. Note that the classes $\mathbb{T}_1$ and $\mathbb{T}_2$ are not disjoint, and there are some trees that belong to both sets. Furthermore, there are trees that belong to neither $\mathbb{T}_1$ nor $\mathbb{T}_2$.

The significance of $\mathbb{T}_1 \cup \mathbb{T}_2$ is that a minimum global secure set of a tree in $\mathbb{T}_1 \cup \mathbb{T}_2$ can be combined with a minimum global secure set of another tree (not necessarily in $\mathbb{T}_1 \cup \mathbb{T}_2$) to form a global secure set of a larger tree. The constructed global secure set is not necessarily a minimum one, but the composition provides an upper bound on the global security number of the larger tree. This property is described more precisely in Lemma 6.2.3.

Lemma 6.2.3. Let T_x be a rooted tree with root x and let T_y be a rooted tree with root y. Let T be the tree constructed by taking the disjoint union of T_x and T_y, and adding an edge between x and y. If T_x (or T_y) is in $\mathbb{T}_1 \cup \mathbb{T}_2$, then $\gamma_s(T) \leq \gamma_s(T_x) + \gamma_s(T_y)$.

Proof. Without loss of generality, let $T_x \in \mathbb{T}_1 \cup \mathbb{T}_2$. Consider two cases based on whether $T_x \in \mathbb{T}_1$ or $T_x \in \mathbb{T}_2$.

1. $T_x \in \mathbb{T}_1$. Then, let S_1 and S_2 be two minimum global secure sets of T_x, such that $x \in S_1$ and $x \notin S_2$. Let S_y be a minimum global secure set of T_y. Consider two sub-cases based on whether $y \in S_y$.

   1.1 If $y \in S_y$, then $S_1 \cup S_y$ is a global secure set of T with cardinality $|S_1| + |S_y| = \gamma_s(T_x) + \gamma_s(T_y)$. This is true because during the construction of T, the edge xy is added between two vertices that are both in the set $S_1 \cup S_y$. Adding an edge between two vertices in a global secure set allows more defenses, but does not increase the number of possible attacks. Thus, $S_1 \cup S_y$ is a global secure set of T.

   1.2 If $y \notin S_y$, then $S_2 \cup S_y$ is a global secure set of T with cardinality $|S_2| + |S_y| = \gamma_s(T_x) + \gamma_s(T_y)$. This is true because during the construction of T, the edge xy is added between two vertices that are both outside the set $S_2 \cup S_y$. Adding an edge between two vertices outside a global secure set does not affect the number of defenses or attacks. Thus, $S_2 \cup S_y$ is a global secure set of T.

In both cases, there exists a global secure set of T of cardinality $\gamma_s(T_x) + \gamma_s(T_y)$, so $\gamma_s(T) \leq \gamma_s(T_x) + \gamma_s(T_y)$.

2. $T_x \in \mathbb{T}_2$. Let S_x be a minimum global secure set of T_x such that $x \in S_x$ and S_x is also a global secure set of $T_x \cup \{x\ell\}$, for a new vertex ℓ. In other words, S_x remains a secure set if an additional attacker of x is to exist. Let S_y be a minimum global secure set of T_y. Consider two sub-cases based on whether $y \in S_y$.

   2.1 If $y \in S_y$, then $S_x \cup S_y$ is a global secure set of T. The justification is similar to case 1.1. That is, during the construction of T, the edge xy is added between two vertices that are both in the set $S_x \cup S_y$.

   2.2 If $y \notin S_y$, then $S_x \cup S_y$ is a global secure set of T. Here, during the construction of T, as the edge xy is added, the vertex y becomes an additional attacker of x. But, S_x is still a secure set in T since by its definition it remains secure even if x receives an additional attacker. Thus, $S_x \cup S_y$ is a global secure set of T.

   In both cases, $S_x \cup S_y$ is a global secure set of T, with cardinality $|S_x| + |S_y| = \gamma_s(T_x) + \gamma_s(T_y)$, so $\gamma_s(T) \leq \gamma_s(T_x) + \gamma_s(T_y)$.

$\square$

Note that Lemma 6.2.3 requires only one of the rooted trees, T_x or T_y, to be in $\mathbb{T}_1 \cup \mathbb{T}_2$.

In cases 1.1, 1.2 and 2.1 of the proof of Lemma 6.2.3, a global secure set for the larger tree T is formed using minimum global secure sets of two smaller rooted trees T_x and T_y. T is constructed by taking the disjoint union of T_x and T_y, along with their respective minimum global secure sets, and then adding an edge between vertices that are either both in the global secure set, or both outside. This strategy is also used in Chapter 7 for constructing minimum global secure sets of grid-like graphs.

Next, Definition 6.2.4 presents the class of rooted trees $\mathbb{L}$, to be used in the proof of Lemma 6.2.8.

Definition 6.2.4. The set of rooted trees $\mathbb{L}$ is defined as follows.

1. The graph $K_{1,t}$ with t being *odd*, $t \geq 3$, and treating the degree t vertex as the root, is a rooted tree in $\mathbb{L}$.

2. Let T_x be a rooted tree with root x, where x has exactly two children ℓ and y. The vertex ℓ is a leaf and the vertex y is the root of a subtree $T_y \in \mathbb{L}$. Then, $T_x \in \mathbb{L}$.

3. The only rooted trees in $\mathbb{L}$ are those defined by Rule 1 or Rule 2.

Figure 6.5 shows three examples of rooted trees in $\mathbb{L}$. Note that the number of vertices of a rooted tree in $\mathbb{L}$ must be even, and at least four.

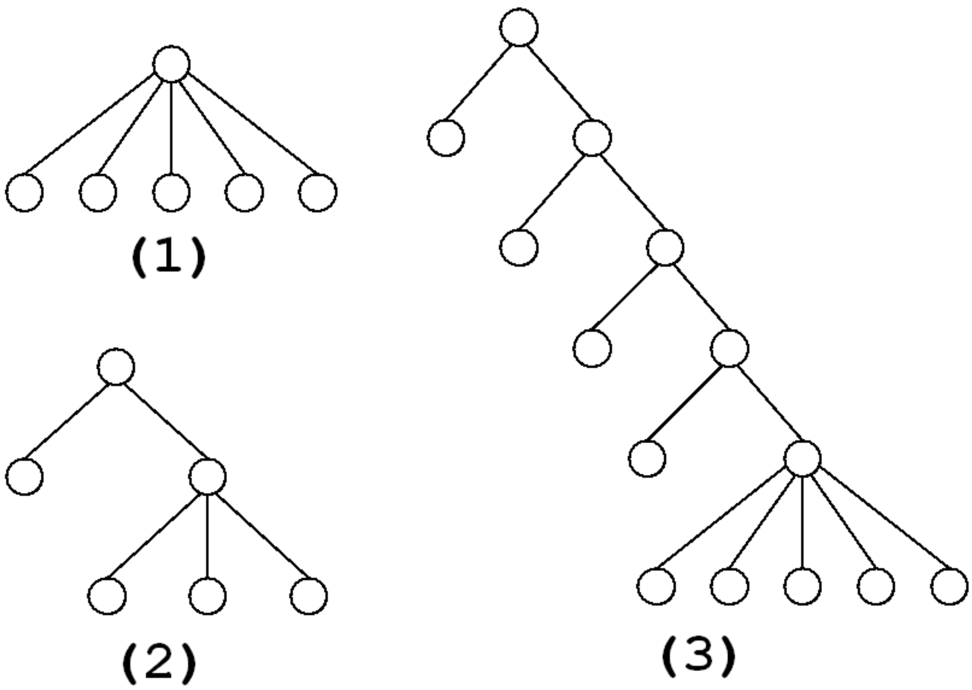

Figure 6.5: Three examples of rooted trees in $\mathbb{L}$. The root of each tree is the topmost vertex.

Lemma 6.2.5. If T is a tree of order n in $\mathbb{L}$, then $\gamma_s(T) = n/2$.

Proof. By Lemma 6.2.1, $\gamma_s(T) \geq n/2$. A global secure set of cardinality exactly $n/2$ includes all internal vertices of T, and $\lfloor k/2 \rfloor$ leaves of the bottom most level, where k is the number of leaves at that level. □

Definition 6.2.6. Let T be a rooted tree with root r. The *depth* of T is the maximum number of vertices in a path between r and a leaf.

Lemma 6.2.7. Let T be a rooted tree of order n and let r be its root. With reference to the auxiliary invariants introduced in Section 6.1 (Definition 6.1.3), if $T \in \mathbb{L}$, then $\min\{\beta_{-1}(T), \sigma_0(T)\} = n/2 + 1$.

Proof. Let $S \subseteq V(T)$ and proceed by induction on the depth of T. The minimum depth of any tree in $\mathbb{L}$ is two. In the base case, suppose T has depth two. Then, $T = K_{1,t}$, where t is odd, $t \geq 3$ and root r is the degree t vertex of $K_{1,t}$. Let $t = 2k+1$, where $k \geq 1$. Note that $n = t+1 = 2k+2$. If S is a σ_0 configuration, $r \notin S$ and so all t children of r must be in S. Then, $\sigma_0(T) = t = 2k+1$. If S is a β_{-1} configuration, $r \in S$, and S must include enough children of r such that $|D_r| - 1 \geq |A_r|$, or $|D_r| \geq |A_r| + 1$[1]. In this case, S must include at least $k+1$ children of r. So, $\beta_{-1}(T) = 1 + (k+1) = k+2$. Then, $\min\{\beta_{-1}(T), \sigma_0(T)\} = \min\{k+2, 2k+1\} = k+2 = n/2+1$.

In the inductive step, suppose T has depth $d > 2$. Then, according to Definition 6.2.4, r has two children, a leaf ℓ and a vertex y, where y is the root of subtree $T_y \in \mathbb{L}$, of depth $d-1$. Let $S_y = S \cap V(T_y)$ and let $n_y = |V(T_y)|$. Note that $n = n_y + 2$. If S is a σ_0 configuration, then $r \notin S$, ℓ must be in S, and S_y is either a $\sigma_0(T_y)$ (in case $y \notin S_y$) or $\beta_{-1}(T_y)$ (in case $y \in S_y$) configuration. So, $\sigma_0(T) = 1 + \min\{\sigma_0(T_y), \beta_{-1}(T_y)\} = 1 + (n_y/2+1) = n/2+1$. If S is a β_{-1} configuration, then $r \in S$ and, if only $n/2$ vertices are included in S, there will be exactly $n/2$ attackers within T, and makes it impossible for S to be a $\beta_{-1}(T)$ configuration, since a valid $\beta_{-1}(T)$ configuration must have $|D_r| \geq |A_r| + 1$. This shows $\beta_{-1}(T) \geq n/2+1$. A valid $\beta_{-1}(T)$ configuration includes all the internal vertices, $\lfloor k/2 \rfloor$ leaves at the bottom most level (k is the number of vertices at the bottom most level), and the vertex ℓ. Thus, $\beta_{-1}(T) \leq n/2+1$, and in conclusion $\sigma_0(T) = \beta_{-1}(T) = n/2+1$.

□

[1]Recall from Definition 6.1.2 that $|D_r|$ and $|A_r|$ denote, respectively, the number of defenders and attackers of r.

Lemma 6.2.7 considers a situation when a rooted tree $T \in \mathbb{L}$ is used in the construction of a larger tree T'. In particular, when the neighbor of r in $T' - T$ is not included in a global secure set S, then either $r \notin S$, in which case $S \cap V(T)$ must correspond to a σ_0 configuration, or $r \in S$, in which case r must be able to defend the additional attacker coming from $T' - T$, and a β_{-1} configuration is required for $S \cap V(T)$. Lemma 6.2.7 states that in this situation $|S \cap V(T)| \geq |V(T)|/2 + 1 > |V(T)|/2 = \gamma_s(T)$ (Lemma 6.2.5).

We now present Lemma 6.2.8. Let T be a tree of order n. Then, for two adjacent vertices $x, y \in V(T)$, let $T - xy$ be the forest obtained by removing edge xy from T. In $T - xy$, there are two components (two trees) T_x and T_y. Let n_x and n_y be, respectively, the order of T_x and T_y. Define $\mathcal{C}(T) = \{xy \in E(T) : n_x \geq 2 \text{ and } n_y \geq 2\}$. So, removing from T a single edge in $\mathcal{C}(T)$ results in a graph without isolated vertices.

Lemma 6.2.8. Let T be a tree of order n. If $n \geq 2$, then $\gamma_s(T) \leq 2n/3$.

Proof. Consider a minimum counter example, a tree T of order n, where $n \geq 2$ and $\gamma_s(T) > 2n/3$. Then, by examining trees of small orders we may conclude $n \geq 4$. Let r be an arbitrarily selected root of T. We want to show that $T \in \mathbb{L}$, by induction in a bottom up fashion. For each level of T, we show that a vertex on that level is either a leaf or the root of a subtree in $\mathbb{L}$.

The base cases are the lowest two levels. In the lowest level, all vertices are leaves. In the next to last level, a vertex is either a leaf or has several children, where each child is a leaf. Let x be such an internal node and let c be the number of children of x. Let T_x be the subtree rooted at x. If $N[x] = V(T)$, then $T = K_{1,(n-1)}$. In this case, $\gamma_s(T) = \lceil n/2 \rceil \leq 2n/3$, a contradiction. So, $N[x] \neq V(T)$. Let p_x be the parent of x in T with respect to r. Since $N[x] \neq V(T)$, $xp_x \in \mathcal{C}(T)$. Consider three cases based on the value of c.

1. If $c = 1$, then $T_x = P_2$ and $T_x \in \mathbb{T}_1$. By Lemma 6.2.3, $\gamma_s(T) \leq \gamma_s(T_x) + \gamma_s(T - T_x)$. Let $n_x = |V(T_x)|$. Then $|V(T - T_x)| = n - n_x$. Since $xp_x \in \mathcal{C}(T)$, $n > n_x \geq 2$ and $n > n - n_x \geq 2$. Since T is a minimum counter example, $\gamma_s(T_x) \leq 2n_x/3$ and $\gamma_s(T - T_x) \leq 2(n - n_x)/3$. Then, $\gamma_s(T) \leq \gamma_s(T_x) + \gamma_s(T - T_x) \leq 2n/3$, a contradiction.

2. If c is even, then $T_x \in \mathbb{T}_2$, and the argument follows similarly to case 1.

3. If c is odd and $c \geq 3$, then $T_x \in \mathbb{L}$.

This completes the base cases. By the way of induction, suppose every vertex on level $i + 1$ or below is either a leaf or the root of a subtree in $\mathbb{L}$. Then, let x be an internal vertex on level i and let c be the number of children of x. We want to show that $T_x \in \mathbb{L}$. By the inductive hypothesis, each child of x is either a leaf or the root of a subtree in $\mathbb{L}$. Let c_1 be the number of subtrees in $\mathbb{L}$ and let c_2 be the number of leaves, among the children of x. First, we claim that $c_1 \leq 1$. Suppose $c_1 \geq 2$ and let Y be the set of children of x that are the roots of their respective subtrees in $\mathbb{L}$. So, $|Y| = c_1 \geq 2$. Let $y \in Y$ and consider $T - T_y$. Let S be a minimum global secure set of $T - T_y$. If $x \notin S$, then (1) every leaf child of x must be in S and (2) $|V(T_{y'}) \cap S| = |V(T_{y'})|/2 + 1$ for all $y' \in Y - \{y\}$. That is, S must include an additional vertex in each $T_{y'}$, compare to a minimum global secure set of $T_{y'}$ alone (Lemma 6.2.7). Modify S by including x and removing the extra vertex from each $T_{y'}$. This new set is of the same (or less) cardinality. The new set is also a secure set since x can defend the attack coming from p_x (if p_x exists and $p_x \notin S$) and every children of x is in this new set. This shows that there exists a minimum global secure set S_x of $T - T_y$ such that x is in the set. Then, since y is in a minimum global secure set S_y of T_y (Lemma 6.2.5), $S_x \cup S_y$ is a global secure set of T. This implies $\gamma_s(T) \leq \gamma_s(T_y) + \gamma_s(T - T_y) \leq (2|V(T_y)|/3) + (2|V(T - T_y)|/3) = 2n/3$, a contradiction. Thus, $c_1 \leq 1$. Now consider two cases.

1. $c_1 = 0$. In this case all children of x are leaves, and by an argument similar to the base case, $c_2 \geq 3$ and is odd, and $T_x \in \mathbb{L}$.

2. $c_1 = 1$. In this case, exactly one child of x is the root of a subtree in $\mathbb{L}$, and all other children of x are leaves. Let y be the root of the child subtree of x in $\mathbb{L}$ and let $n_y = |V(T_y)|$ be the number of vertices in T_y. We claim that $c_2 = 1$. Suppose $c_2 \geq 2$ and consider $T - T_y$. Let S be a minimum global secure set of $T - T_y$. If $x \notin S$, then all c_2 leaf children of x are in S. Including x in S can then allow the exclusion of $\left\lfloor \dfrac{c_2}{2} \right\rfloor$ leaves and retain a global secure set. Since $c_2 \geq 2$, such sets are also minimum (or in fact smaller if $c_2 \geq 4$). This shows that there exists a minimum global secure set S_x of $T - T_y$ such that $x \in S_x$. Similar to the analysis above, since y is in a minimum global secure set of T_y, $\gamma_s(T) \leq \gamma_s(T_y) + \gamma_s(T - T_y) \leq 2n/3$, a contradiction. Thus, $c_2 \leq 1$. Assume $c_2 = 0$ and consider three cases, depending upon whether x is the root of T, and if not, whether $T - xp_x$ has isolated vertices (i.e., whether $xp_x \notin \mathcal{C}(T)$).

2.1 $x = r$. Then, a global secure set of T may be constructed by including x and $n_y/2$ vertices in T_y (Lemma 6.2.5). So, $\gamma_s(T) \leq 1 + \frac{n_y}{2} = 1 + \frac{n-1}{2} = \frac{n+1}{2} \leq \frac{2n}{3}$, a contradiction.

2.2 $x \neq r$ and $xp_x \notin \mathcal{C}(T)$. In this case, $\deg(p_x) = 1$ and $p_x = r$. Treating x as the root of T makes T a rooted tree in $\mathbb{L}$, and by Lemma 6.2.5, $\gamma_s(T) = n/2 \leq 2n/3$, a contradiction.

2.3 $x \neq r$ and $xp_x \in \mathcal{C}(T)$. Then, in $T - xp_x$ there are two components, each of order at least 2. Let S' be a minimum global secure set of the component which contains p_x. Then, the set S', along with vertex x and $n_y/2$ vertices of T_y forms a global secure set of T. In this set, x may defend the attack coming from p_x if $p_x \notin S'$. This set has cardinality $|S'|+1+n_y/2 \leq 2(n-n_x)/3+(n_x+1)/2 \leq 2(n-n_x)/3+2n_x/3 = 2n/3$, where $n_x = |V(T_x)| = n_y + 1$. Then, $\gamma_s(T) \leq 2n/3$, a contradiction.

Thus, $c_2 = 1$ and $T_x \in \mathbb{L}$.

This completes the inductive step and so $T \in \mathbb{L}$. By Lemma 6.2.5, $\gamma_s(T) = n/2 \leq 2n/3$, the final contradiction to the counter example. □

Recall from Theorem 2.1.16 that $(n+2)/4 \leq \gamma_a(T) \leq (3n)/5$ are sharp bounds on the global defensive alliance number of a tree of order $n \geq 4$. Theorem 6.2.9 presents a similar result regarding the global security number of a tree of order $n \geq 2$.

Theorem 6.2.9. If T is a tree of order $n \geq 2$, then $\lceil n/2 \rceil \leq \gamma_s(T) \leq \lfloor 2n/3 \rfloor$, and both bounds are sharp.

Proof. By Lemma 6.2.1, $\gamma_s(T) \geq \lceil n/2 \rceil$ and by Lemma 6.2.8, $\gamma_s(T) \leq 2n/3$. Since $\gamma_s(T)$ is an integer, $\gamma_s(T) \leq \lfloor 2n/3 \rfloor$. Sharpness of the lower bound are realized by the set of paths P_n for $n \geq 2$ (and other examples discussed in the remark following Lemma 6.2.1). For sharpness of the upper bound, let $n = 3t + 1$ and construct a rooted tree T of order n. Let r be the root of T. The vertex r has t children, where each child has in addition two more children. Then, in a global secure set of T each child subtree must include at least two vertices, with possible exception for one of the subtrees when r is in the set. So, $\gamma_s(T) = 2t = \lfloor 2n/3 \rfloor$. □

There is no known sharp upper bound on the global security number of an arbitrary graph in terms of only its order. This is posted as Open Problem 9.2.2.

6.3 Global security numbers and spanning trees

In this section, we will investigate and present results on the relationship between the global security number of a connected graph and that of its spanning trees, with Lemma 6.3.3 being the primary result.

Definition 6.3.1. An edge e of graph G is a *bridge* if removing e from G increases the number of components of G.

Observation 6.3.2. Let G be a graph and let S be a (not necessarily global) secure set of G. For $uv \in G$, if $u \in V(G) - S$, then S is a secure set of $G - uv$.

Lemma 6.3.3. For any connected graph G, there exists a spanning tree T of G such that $\gamma_s(G) \geq \gamma_s(T)$.

Proof. Let $G = (V, E)$ be an arbitrary connected graph and let S be a global secure set of G. S is not necessarily a minimum global secure set of G. Consider a partition of E into three sets E_1, E_2 and E_3, depending on where the endpoints of an edge lies, with respect to $V - S$ and S, as follows.

1. E_1 contains those edges in E where both endpoints are in $V - S$.

2. E_2 contains those edges in E where both endpoints are in S.

3. E_3 contains those edges in E where exactly one endpoint is in S (and the other endpoint is in $V - S$).

Apply the following operations on G, let the result be H.

1. Let $H \leftarrow G$.

2. For each $e \in E_1 \cap E(H)$, if e is not a bridge of H, let $H \leftarrow (H - e)$.

3. For each $e \in E_3 \cap E(H)$, if e is not a bridge of H, let $H \leftarrow (H - e)$.

By Observation 6.3.2, S is a secure set of H. We show next that S is a dominating set of H, and hence S is a global secure set of H. Clearly, S is a dominating set of H after operation 2, and any edge remaining in $E_1 \cap E(H)$ after operation 2 must be a bridge. If operation 3 did not remove any edges, S is a dominating set of H. Otherwise, assume that

$u \in V - S$ is a vertex which is not dominated after operation 3. Let uv be the last edge removed in operation 3, among all edges incident to u. Then, $uv \in E_3$, $v \in S$ and uv is not a bridge prior to its removal. Let C be a cycle containing uv, and let $w \neq v$ be the other neighbor of u in C. Since uv is the last edge removed among all edges incident to u, edge uw remains in H after operation 3. Since u is not dominated after operation 3, $w \in V - S$ and $uw \in E_1$. But, uw is in the cycle C, therefore is not a bridge. This is a contradiction to the fact that uw was not removed by operation 2. Thus, S is a global secure set of H. Note that every edge being removed is not be a bridge of H, and so its removal does not disconnect H. Therefore, after the operations H is a connected subgraph of G.

After operations $1 - 3$, every edge in $(E_1 \cup E_3) \cap E(H)$ is a bridge. In H, any vertex $u \in V - S$ has at most one neighbor in any component of $H[S]$. That is, if u has multiple neighbors in S, then they must belong to different components of $H[S]$, for otherwise an edge in E_3 remains in H and is not a bridge. Then, for any component C' of $H[S]$, an attacker in $V - S$ can attack at most one vertex of C', and C' is secure if there exists a feasible defense for this *unique* attack (with respect to C'). But S is a secure set of H, so a feasible defense D exists. Let $E_D \subseteq E_2$ be the edges used by D. The edges in E_D induce a forest. Apply the following operations on H and let the result be T.

1. $T \leftarrow H$.

2. For each $e \in (E_2 - E_D) \cap E(T)$, if e is not a bridge of T, let $T \leftarrow (T - e)$.

Note that T is connected since any edge removed from T is not a bridge. If T contains a cycle C, then none of the edges of C belong to $E_1 \cup E_3$, since $E_1 \cup E_3$ contains only bridges. So, $E(C) \subseteq E_2$. If $E(C) \subseteq E_D$, then E_D contains a cycle, a contradiction. Thus, there is an edge in C which belongs to $E_2 - E_D$. But, this is also impossible since this edge should have been removed by the above operation. Therefore, T is a spanning tree of G, and S is a global secure set of T. S is a dominating set because edges in E_3 are preserved, and S is secure because edges in E_D are preserved.

In conclusion, if S is a global secure set of G, then S is a global secure set of some spanning tree of G. The result of this lemma follows by letting S be a minimum global secure set of G. $\square$

Lemma 6.3.4. There is a graph G such that for any spanning tree T of G, $\gamma_s(G) > \gamma_s(T)$.

Proof. Let $G = C_6$, any spanning tree of C_6 is a P_6, and $\gamma_s(C_6) = 4 > 3 = \gamma_s(P_6)$. $\square$

A more general example for Lemma 6.3.4 is C_{4k+2}, where $\gamma_s(C_{4k+2}) = 2k + 2$ (Theorem 8.1.2), but $\gamma_s(P_{4k+2}) = 2k + 1$ (Theorem 7.2.1).

Lemma 6.3.5. There is a graph G such that for some spanning tree T of G, $\gamma_s(G) < \gamma_s(T)$.

Proof. Consider K_{3k+1}, the complete graph on $3k + 1$ vertices. Then, let T be a rooted tree where the root has exactly k children, and each child of the root has exactly 2 children. T has $3k+1$ vertices and is a spanning tree of K_{3k+1}. $\gamma_s(K_{3k+1}) = \lceil (3k + 1)/2 \rceil$ and $\gamma_s(T) = 2k$. $\square$

CHAPTER 7

CONSTRUCTIONS OF GLOBAL SECURE SETS OF GRID-LIKE GRAPHS

7.1 Introduction

This chapter and the next present results on the global security numbers of grid-like graphs. Recall from Section 1.4 the definitions of Cartesian product (Definition 1.4.3) and grid-like graphs (Definition 1.4.4), presented again below.

The *Cartesian product* of two graphs G and H is a graph denoted $G \times H$, where $V(G \times H) = V(G) \times V(H)$ and $E(G \times H) = \{(v_i, u_i)(v_j, u_j) : (v_i = v_j \text{ and } u_i u_j \in E(H)) \text{ or } (v_i v_j \in E(G) \text{ and } u_i = u_j)\}$.

A *path* on $n \geq 2$ vertices is a graph P_n where $V(P_n) = \{v_1, v_2, \ldots, v_n\}$ and $E(P_n) = \{v_i v_{i+1} : 1 \leq i \leq n-1\}$. A *cycle* on $n \geq 3$ vertices is a graph C_n where $V(C_n) = \{v_1, v_2, \ldots, v_n\}$ and $E(C_n) = \{v_i v_{i+1} : 1 \leq i \leq n-1\} \cup \{v_1 v_n\}$. A *two-dimensional grid* $P_n \times P_m$ is the Cartesian product of two paths P_n and P_m. A *two-dimensional cylinder* $P_n \times C_m$ is the Cartesian product of a path P_n and a cycle C_m. A *two-dimensional torus* $C_n \times C_m$ is the Cartesian product of two cycles C_n and C_m.

The class of graphs which contains exactly all paths, cycles and two-dimensional grids, cylinders and tori is the class of *grid-like* graphs. Note that the order of each path is at least 2 and the order of each cycle is at least 3. For example, the smallest two-dimensional torus is $C_3 \times C_3$, which has 9 vertices.

Let G be a grid-like graph of order n. The security number of G is given in Theorem 2.3.4, which states that $s(G) \leq 12$. On the other hand, by Lemma 6.2.1 in Chapter 6, the global security number of G is at least $\lceil n/2 \rceil$. Theorem 7.1.1 presents our main result regarding the global security numbers of grid-like graphs.

Theorem 7.1.1. Let G be a grid-like graph of order n. Then, $\gamma_s(G) = \lceil n/2 \rceil$, unless G is isomorphic to C_{4k+2} or $C_3 \times C_{4k+2}$, in which case $\gamma_s(G) = n/2 + 1$.

Theorem 7.1.1 states that the general lower bound $\lceil n/2 \rceil$ is realized for all grid-like graphs, with the curious exceptions C_{4k+2} and $C_3 \times C_{4k+2}$. This chapter and the next develop in detail the result stated in Theorem 7.1.1.

Let G be a grid-like graph of order n. In this chapter, an upper bound on $\gamma_s(G)$ is established by exhibiting a global secure set of the desired cardinality. In most cases, a global secure set of cardinality $\lceil n/2 \rceil$ is given for G, proving $\gamma_s(G) \leq \lceil n/2 \rceil$. Along with Lemma 6.2.1, the result is $\gamma_s(G) = \lceil n/2 \rceil$. The only exceptions are when G is isomorphic to C_{4k+2} or $C_3 \times C_{4k+2}$. In these two cases, we exhibit a global secure set of cardinality $n/2+1$ for G, proving $\gamma_s(G) \leq n/2 + 1$. Then, in the next chapter, a lower bound of $n/2 + 1$ for these two graphs will be established through specialized analysis.

In general, global secure sets of larger graphs are constructed using global secure sets of smaller graphs. Definition 7.1.2 provides several graph operations used during the constructions of the global secure sets of interest. Then, Observation 7.1.3 gives some sufficient conditions on when global secure sets of a set of graphs $\mathscr{G}$ can be used to provide a global secure set for another graph H, where H is obtained by performing specific operations on graphs in $\mathscr{G}$.

Definition 7.1.2. Let G and H be graphs. The *disjoint union* of G and H, denoted $G \cup H$, is the graph with $V(G \cup H) = V(G) \cup V(H)$ and $E(G \cup H) = E(G) \cup E(H)$, such that $V(G)$ and $V(H)$ are disjoint sets of vertices in $G \cup H$. So, $|V(G \cup H)| = |V(G)| + |V(H)|$. For $uv \notin E(G)$, the *edge addition* of uv to G, denoted $G+uv$, is the graph with $V(G+uv) = V(G)$ and $E(G + uv) = E(G) \cup \{uv\}$. For $u, v \in V(G)$, let G' be the graph obtained from G by deleting u and v, and then adding a new vertex w such that $N_{G'}(w) = N_G(u) \cup N_G(v) - \{u, v\}$. The graph G' is obtained from G by *identifying* u and v.

Observation 7.1.3. Let S be a global secure set of graph G and let S_1 and S_2 be global secure sets of graphs G_1 and G_2, respectively. Then,

1. $S_1 \cup S_2$ is a global secure set of $G_1 \cup G_2$ ([Jes10]).

2. Let $u, v \in V(G)$ such that $uv \notin E(G)$. If $(u, v \in S)$ or $(u, v \in V(G) - S)$, then S is a global secure set of $G + uv$.

3. For $u, v \in V(G) - S$, let H be the graph obtained from G by identifying u with v. Then, S is a global secure set of H.

4. For $v \in V(G) - S$, S is a global secure set of $G - v$.

We will tacitly assume the statements of Observation 7.1.3. Next, Section 7.2 provides constructions of global secure sets for paths and cycles. Then, Sections 7.3, 7.4 and 7.5 provide constructions for two-dimensional grids ($P_n \times P_m$), cylinders ($P_n \times C_m$) and tori ($C_n \times C_m$), respectively.

7.2 Paths and cycles

In this section, we apply the methodology outlined in Section 7.1 and construct global secure sets for paths P_n and cycles C_n. Theorem 7.2.1 shows that $\gamma_s(P_n) = \lceil n/2 \rceil$. Then, Lemmas 7.2.2 and 7.2.3 present upper bounds on the global security numbers of C_n, based on whether $n \in \{4k + 2 : k \geq 1\}$. Finally, Theorem 7.2.4 summarizes the results for C_n obtained in this section.

Theorem 7.2.1. $\gamma_s(P_n) = \lceil n/2 \rceil$ for $n \geq 2$.

Proof. By Lemma 6.2.1, $\gamma_s(P_n) \geq \lceil n/2 \rceil$. We construct global secure sets of cardinality $\lceil n/2 \rceil$ for P_n. Whenever appropriate, the global secure sets provided do not include the endpoints of the paths. This property enables the application of Observation 7.1.3 (Part 2) in the constructions of larger paths.

Figure 7.1 shows global secure sets of P_n for $2 \leq n \leq 5$, each of cardinality $\lceil n/2 \rceil$. Note that for P_4 and P_5, the global secure sets do not include the endpoints of the paths.

For $n > 5$, consider four cases based on the value of $n \pmod 4$. For $n \equiv 0 \pmod 4$, the construction is done recursively using a global secure set of P_{n-4} and a copy of P_4 (in Figure 7.1). In the other cases, the constructions are done by using a global secure set of P_{4h} and a global secure set of a small path.

1. $n \in \{4k : k \geq 2\}$. First, construct recursively a global secure set of cardinality $2(k-1)$ for $P_{4(k-1)}$, where the endpoints of $P_{4(k-1)}$ are not in the set. Then, take the disjoint union of P_4 (with a global secure set given in Figure 7.1) and $P_{4(k-1)}$. Since the endpoints of both paths are outside the set, by Observation 7.1.3 (Part 2), adding an edge between an endpoint of P_4 and an endpoint of $P_{4(k-1)}$ produces P_{4k}, with a global secure set of cardinality $2 + 2(k-1) = 2k$. Note that both endpoints of P_{4k} are outside the global secure set.

2. $n \in \{4k+1 : k \geq 2\}$. Consider the disjoint union of P_5 (in Figure 7.1) and $P_{4(k-1)}$. The endpoints of both paths are outside the set. By Observation 7.1.3 (Part 2), adding an edge between an endpoint of P_5 and an endpoint of $P_{4(k-1)}$ produces P_{4k+1}, with a global secure set of cardinality $3 + 2(k-1) = 2k+1$. The endpoints of P_{4k+1} are outside the global secure set.

3. $n \in \{4k+2 : k \geq 1\}$. Consider the disjoint union of P_2 (in Figure 7.1) and P_{4k}. One endpoint of P_2 is not in the global secure set, and both endpoints of P_{4k} are not in the global secure set. By Observation 7.1.3 (Part 2), adding an edge between the endpoint of P_2 which is not in the set and an endpoint of P_{4k} produces P_{4k+2}, with a global secure

set of cardinality $2k+1$. Notice in P_{4k+2} *exactly one* endpoint is included in the global secure set.

4. $n \in \{4k+3 : k \geq 1\}$. For $k=1$, a global secure set of cardinality 4 for P_7 can be constructed by taking the disjoint union of two copies of P_4 (in Figure 7.1) and identifying one of their endpoints (Observation 7.1.3, Part 3). The result is illustrated in Figure 7.2. Note that both endpoints of P_7 are outside the global secure set.

   For $k > 1$, consider the disjoint union of P_7 and $P_{4(k-1)}$. The endpoints of P_7 and $P_{4(k-1)}$ are outside the global secure set. By Observation 7.1.3 (Part 2), adding an edge between an endpoint of P_7 and an endpoint of $P_{4(k-1)}$ produces P_{4k+3}, with a global secure set of cardinality $4+2(k-1)=2k+2$. The endpoints of P_{4k+3} are outside the specified global secure set.

□

Refer to Figure 7.3 for illustrations of global secure set patterns specified in the proof of Theorem 7.2.1.

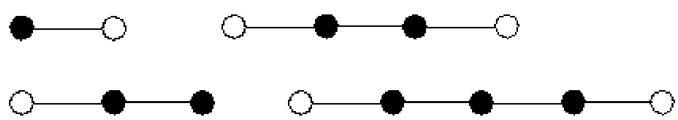

Figure 7.1: Global secure sets of P_2, P_3, P_4 and P_5.

Theorem 7.2.1 determines the global security number of P_n. In the proof of Theorem 7.2.1, we constructed a global secure set for a path P_n by considering global secure sets of

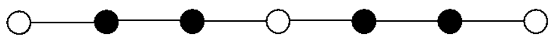

Figure 7.2: A global secure set of P_7 as constructed by the proof of Theorem 7.2.1 (Part 4).

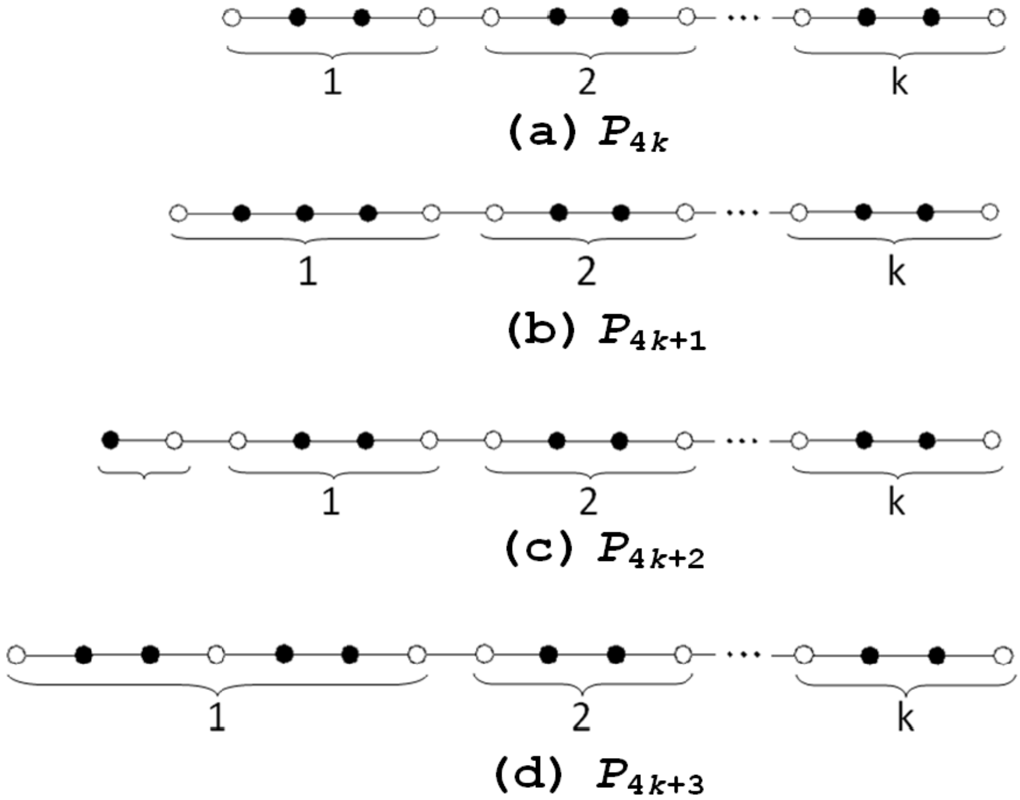

Figure 7.3: Global secure sets of P_n, as constructed by the proof of Theorem 7.2.1.

smaller paths $\{P_{n'} : n' < n\}$, and applying operations given in Observation 7.1.3 on these graphs. The same technique will be used for constructing global secure sets of $P_n \times P_m$ in Section 7.3. The remainder of this section examines the global security numbers of cycles C_n.

Lemma 7.2.2. If $n \notin \{4k + 2 : k \geq 1\}$, then $\gamma_s(C_n) = \lceil n/2 \rceil$.

Proof. By Lemma 6.2.1, $\gamma_s(C_n) \geq \lceil n/2 \rceil$. We establish equality by constructing global secure sets of cardinality $\lceil n/2 \rceil$ for C_n when $n \notin \{4k + 2 : k \geq 1\}$. Clearly, $\gamma_s(C_3) = 2$. For $n \geq 4$, consider a global secure set configuration for P_n of cardinality $\lceil n/2 \rceil$ as given by the proof of Theorem 7.2.1. Note that when $n \notin \{4k + 2 : k \geq 1\}$ (cases 1, 2 and 4 in the proof of Theorem 7.2.1), both endpoints of P_n are outside the specified global secure set. By Observation 7.1.3 (Part 2), adding an edge between the endpoints of P_n produces C_n, with a global secure set of cardinality $\lceil n/2 \rceil$. □

The proof of Lemma 7.2.2 constructed global secure sets for C_n using the global secure sets of P_n from Theorem 7.2.1. This works in all cases except when $n \in \{4k + 2 : k \geq 1\}$. In that case, one endpoint of P_{4k+2} is in the global secure set, while the other endpoint is not, and the edge addition operation given in Observation 7.1.3 (Part 2) cannot be applied. Nonetheless, we may show that $\gamma_s(C_{4k+2}) \leq 2k + 2$ by constructing global secure sets of cardinality $2k + 2$.

Lemma 7.2.3. $\gamma_s(C_{4k+2}) \leq 2k + 2$ for $k \geq 1$.

Proof. Figure 7.4 shows a global secure set of P_6 with 4 vertices. Note that this configuration has the property where both endpoints of P_6 are not in the global secure set. Then, for C_6, add an edge between the endpoints of P_6 in Figure 7.4. This results in a global secure set for C_6 with 4 vertices, and $\gamma_s(C_6) \leq 4$.

For $k > 1$, consider the disjoint union of P_6 (in Figure 7.4) and $P_{4(k-1)}$ (with a global secure set of $2(k-1)$ vertices as constructed in the proof of Theorem 7.2.1). Note that the endpoints of P_6 and $P_{4(k-1)}$ are outside the global secure set. By Observation 7.1.3 (Part 2), adding two edges between their respective endpoints produces C_{4k+2}, with a global secure set of cardinality $4 + 2(k-1) = 2k+2$. So, $\gamma_s(C_{4k+2}) \leq 2k+2$. The final configuration is illustrated in Figure 7.5. $\square$

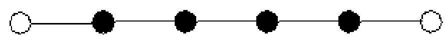

Figure 7.4: A global secure set of P_6 with 4 vertices.

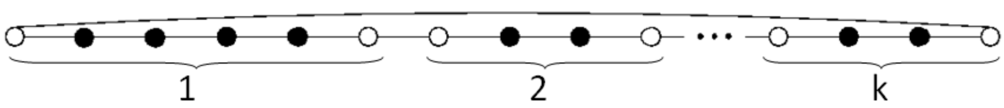

Figure 7.5: A global secure set of C_{4k+2}, as constructed by the proof of Lemma 7.2.3.

As mentioned in Section 7.1, a proof for $\gamma_s(C_{4k+2}) > 2k+1$ will be presented in Chapter 8. Along with Lemma 7.2.3, the result is $\gamma_s(C_{4k+2}) = 2k+2$ (Theorem 8.1.2). Theorem 7.2.4 summarizes the results obtained so far for C_n.

Theorem 7.2.4. $\gamma_s(C_n) = \begin{cases} 2k+1 \text{ or } 2k+2 & \text{if } n \in \{4k+2 : k \geq 1\} \\ \lceil n/2 \rceil & \text{otherwise} \end{cases}$

Proof. By Lemmas 6.2.1, 7.2.2 and 7.2.3. $\square$

7.3 Two-dimensional grids

In this section, we investigate the global security numbers of two-dimensional grids, $P_n \times P_m$. The methodology applied is analogous to the one used in the previous section. We start by providing global secure sets for $P_n \times P_m$ of small order, and global secure sets of larger graphs are obtained by taking the disjoint union of smaller graphs (along with their respective global secure sets), and adding edges between vertices that are either both in the global secure set or both outside. Note that we only need to demonstrate one global secure set whose cardinality matches the lower bound given in Lemma 6.2.1. In general, there may be many such configurations.

Next, Remark 7.3.1 introduces a program that is used for verifying the validity of global secure set configurations of several small graphs. These small graphs and their respective global secure set configurations will be used in the upcoming developments of this chapter. Then, Definition 7.3.2 and Lemma 7.3.3 specifies certain situations where a global secure set configuration for $P_n \times P_m$ may be used as a valid global secure set configuration of $P_n \times C_m$ or $C_n \times C_m$.

Remark 7.3.1. Recall from Section 3.3 that there is no known polynomial algorithm for verifying the validity of a secure set configuration. But, when the graphs under consideration are small, exhaustive search algorithms can still verify the validity of secure sets in a reasonable amount of time. A program has been developed for `Is Secure` (Problem 3.1.2) on grid-like graphs. The program accepts an input grid-like graph and a vertex subset, enumerates all possible attacks, and finds a feasible defense for each attack (or determines that none exists). A feasible defense for a fixed attack is found using the network flow formulation given in Section 3.2. The validity of global secure sets illustrated in Figures 7.13 ($P_3 \times P_6$), 7.24 ($P_6 \times P_5$), 7.31 ($P_6 \times P_6$), 7.37 ($P_7 \times P_7$), 7.40 ($C_3 \times P_7$), 7.50 ($C_7 \times P_6$) and 7.51 ($C_7 \times P_4$) was verified by this program. Notice we can verify by hand that each configuration is also a dominating set.

Definition 7.3.2. Let S be a global secure set of $P_n \times P_m$. Consider the vertices of $P_n \times P_m$ as an array with n rows and m columns, where $V(P_n \times P_m) = \{v_{i,j} : 1 \leq i \leq n, 1 \leq j \leq m\}$. With this notation, a vertex $v_{i,j}$ has at most four neighbors $\{v_{(i-1),j}, v_{(i+1),j}, v_{i,(j-1)}, v_{i,(j+1)}\}$. The *top row* of $P_n \times P_m$ is the set $\{v_{1,j} : 1 \leq j \leq m\}$ and the *bottom row* of $P_n \times P_m$ is the set $\{v_{n,j} : 1 \leq j \leq m\}$. The top and bottom rows of $P_n \times P_m$ are *identical* when $v_{1,j} \in S$ if and only if $v_{n,j} \in S$ for $1 \leq j \leq m$. That is, any vertex on the top row and its corresponding vertex on the bottom row are either both in S, or both outside. Similarly, the leftmost and rightmost columns of $P_n \times P_m$ are identical when $v_{i,1} \in S$ if and only if $v_{i,m} \in S$ for $1 \leq i \leq n$.

Whenever possible, we want to construct global secure sets of $P_n \times P_m$ with identical top and bottom rows, and identical leftmost and rightmost columns. This way, the configurations can be extended to global secure set configurations of $P_n \times C_m$, $C_n \times P_m$ and $C_n \times C_m$, as shown next in Lemma 7.3.3.

Lemma 7.3.3. Let S be a global secure set of $P_n \times P_m$ for $n, m \geq 2$. If the leftmost and rightmost columns of $P_n \times P_m$ are identical, then S is a global secure set of $P_n \times C_m$ for

$m \geq 3$. If the top and bottom rows of $P_n \times P_m$ are identical, then S is a global secure set of $C_n \times P_m$ for $n \geq 3$. If, in $P_n \times P_m$, the leftmost and rightmost columns are identical, and the top and bottom rows are identical, then S is a global secure set of $C_n \times C_m$ for $n, m \geq 3$.

Proof. If the leftmost and rightmost columns of $P_n \times P_m$ are identical, we may add edges between corresponding vertices in these two columns and, by Observation 7.1.3 (Part 2), S will be a global secure set of the resulting graph $P_n \times C_m$. Similarly, if the top and bottom rows are identical, adding edges between corresponding vertices on these two rows produces the graph $C_n \times P_m$, where S is a global secure set. If the leftmost and rightmost columns are identical, and the top and bottom rows are identical, adding the appropriate edges produces $C_n \times C_m$ with S as a global secure set. □

Observation 7.3.4. If graphs G and H are isomorphic, then $\gamma_s(G) = \gamma_s(H)$. In particular, $\gamma_s(P_n \times P_m) = \gamma_s(P_m \times P_n)$, $\gamma_s(P_n \times C_m) = \gamma_s(C_m \times P_n)$ and $\gamma_s(C_n \times C_m) = \gamma_s(C_m \times C_n)$.

We will tacitly assume the result of Observation 7.3.4.

The constructions of global secure sets for $P_n \times P_m$ are done by cases based on the values of n and m. The cases are $n = 2$, $n = 3$, and for $n, m \geq 4$, based on the values of $n \pmod 4$ and $m \pmod 4$. First, Lemma 7.3.5 treats the case $n = 2$ and Lemma 7.3.7 treats the case $m \equiv 0 \pmod 4$.

Lemma 7.3.5. $\gamma_s(P_2 \times P_m) = m$, for $m \geq 2$.

Proof. By Lemma 6.2.1, $\gamma_s(P_2 \times P_m) \geq m$. A global secure set of $P_2 \times P_m$ consists of one row of m vertices. Figure 7.6 illustrates such a set. So, $\gamma_s(P_2 \times P_m) \leq m$. □

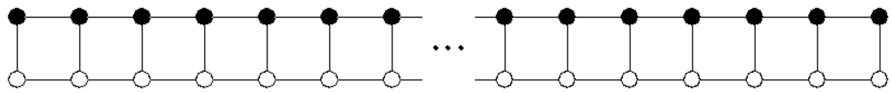

Figure 7.6: A global secure set of $P_2 \times P_m$.

Corollary 7.3.6. $\gamma_s(P_2 \times C_m) = m$, for $m \geq 3$.

Proof. By Lemma 6.2.1, $\gamma_s(P_2 \times C_m) \geq m$. Let S be the global secure set of $P_2 \times P_m$ given by the proof of Lemma 7.3.5. The leftmost and rightmost columns of $P_2 \times P_m$ are identical. By Lemma 7.3.3, S is also a global secure set of $P_2 \times C_m$. So, $\gamma_s(P_2 \times C_m) \leq m$. □

Lemma 7.3.7. $\gamma_s(P_n \times P_{4k}) = 2kn$, for $n \geq 2$ and $k \geq 1$.

Proof. By Lemma 6.2.1, $\gamma_s(P_n \times P_{4k}) \geq 2kn$. We construct global secure sets for $P_n \times P_{4k}$ using n copies of global secure sets of P_{4k} (as given by the proof of Theorem 7.2.1, Part 1).

For $n = 2$, consider the disjoint union of two copies of P_{4k} along with their global secure set configurations as given by the proof of Theorem 7.2.1 (Part 1). Since the two copies of P_{4k} are identical, we may add edges between corresponding vertices in these two copies to produce $P_2 \times P_{4k}$, with a global secure set of cardinality $(2 \times 2k) = 4k$. Figure 7.7 illustrates the result of this process.

For $n > 2$, first construct recursively a global secure set of cardinality $2k(n-1)$ for $P_{n-1} \times P_{4k}$, where each row of $P_{n-1} \times P_{4k}$ is identical to P_{4k}. Then, take the disjoint union of $P_{n-1} \times P_{4k}$ and another copy of P_{4k}. Since the last row of $P_{n-1} \times P_{4k}$ is identical to P_{4k}, adding edges between corresponding vertices in P_{4k} and the last row of $P_{n-1} \times P_{4k}$ produces $P_n \times P_{4k}$, with a global secure set of cardinality $2k(n-1) + 2k = 2kn$. So, $\gamma_s(P_n \times P_{4k}) \leq 2kn$. □

Refer to Figure 7.8 for the global secure set pattern specified in the proof of Lemma 7.3.7.

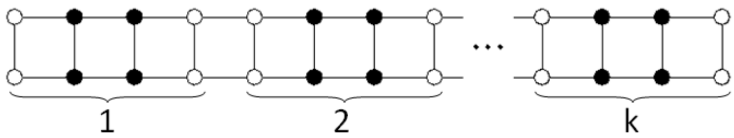

Figure 7.7: A global secure set of $P_2 \times P_{4k}$.

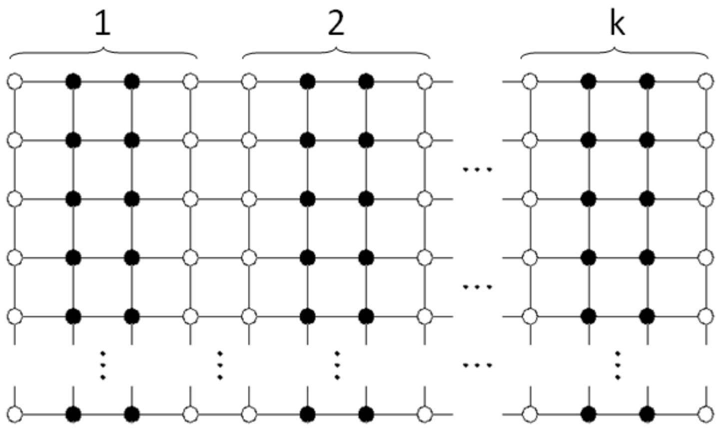

Figure 7.8: The global secure set pattern for $P_n \times P_{4k}$ as constructed by the proof of Lemma 7.3.7.

Corollary 7.3.8. Let $k \geq 1$. Then, $\gamma_s(P_n \times C_{4k}) = 2kn$ for $n \geq 2$, and $\gamma_s(C_n \times P_{4k}) = \gamma_s(C_n \times C_{4k}) = 2kn$ for $n \geq 3$.

Proof. Consider the global secure set S of $P_n \times P_{4k}$ as given by the proof of Lemma 7.3.7. In this configuration, the leftmost and rightmost columns are identical, and the top and bottom rows are identical. By Lemma 7.3.3, S is also a global secure set of $P_n \times C_{4k}$, $C_n \times P_{4k}$ and $C_n \times C_{4k}$. This shows the upper bound, and the lower bound is established by Lemma 6.2.1. $\square$

Proofs of future similar corollaries based on Lemma 7.3.3 will not be presented.

Lemma 7.3.9. The marked vertices form a global secure set of the graph shown in Figure 7.11.

Proof. Consider the disjoint union of the graphs in Figures 7.9 and 7.10. By Observation 7.1.3 (Part 2), adding appropriate edges produces the graph in Figure 7.11, with the same global secure set. □

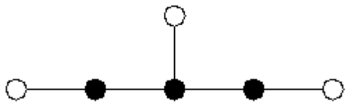

Figure 7.9: A graph and one of its global secure sets marked in black.

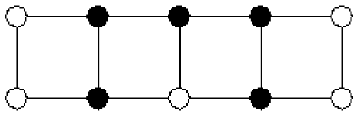

Figure 7.10: A global secure set of $P_2 \times P_5$ with 5 vertices.

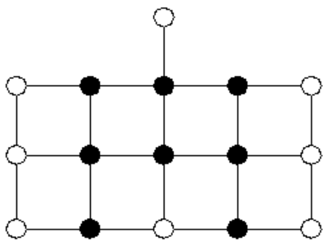

Figure 7.11: A graph and one of its global secure sets marked in black. See the proof of Lemma 7.3.9 for details.

The graph in Figure 7.11 will be used to construct global secure sets for $P_3 \times P_5$ (Lemma 7.3.10) and $P_7 \times P_5$ (Lemma 7.3.21). Next, Lemmas 7.3.10, 7.3.12 and 7.3.14 consider cases with $n = 3$ and different values of m (mod 4). Then, Corollary 7.3.16 collects the results and shows that $\gamma_s(P_3 \times P_m) = \lceil 3m/2 \rceil$.

Lemma 7.3.10. $\gamma_s(P_3 \times P_{4k+1}) = \lceil 3(4k+1)/2 \rceil$, for $k \geq 1$.

Proof. We construct global secure sets of cardinality $\lceil 3(4k+1)/2 \rceil = 6k + 2$ for $P_3 \times P_{4k+1}, k \geq 1$. The configurations are given in way such that the leftmost and rightmost columns contain no vertex in the set.

For $k = 1$, removing the degree one attacker from the graph in Figure 7.11 gives a global secure set of cardinality 8 for $P_3 \times P_5$ (Observation 7.1.3, Part 4). In this configuration, vertices of the leftmost and rightmost columns are not in the global secure set.

For $k > 1$, consider the disjoint union of $P_3 \times P_5$ and $P_3 \times P_{4(k-1)}$ (with a global secure set for the latter given by the proof of Lemma 7.3.7). The rightmost column of $P_3 \times P_5$ and the leftmost column of $P_3 \times P_{4(k-1)}$ neither contains any vertex in the set and hence are identical. By Observation 7.1.3 (Part 2), adding edges between corresponding vertices in these two columns produces $P_3 \times P_{4k+1}$, with a global secure set of cardinality $8 + 6(k-1) = 6k+2$. $\square$

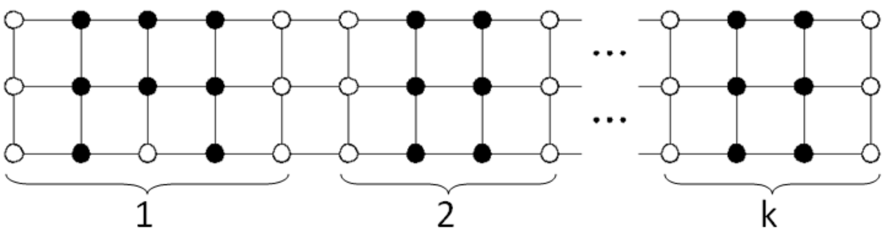

Figure 7.12: A global secure set of $P_3 \times P_{4k+1}$ as constructed by the proof of Lemma 7.3.10.

Figure 7.12 illustrates the global secure set pattern for $P_3 \times P_{4k+1}$ specified in the proof of Lemma 7.3.10. Notice in these configurations, the leftmost and rightmost columns are identical.

Corollary 7.3.11. $\gamma_s(P_3 \times C_{4k+1}) = \lceil 3(4k+1)/2 \rceil$, for $k \geq 1$.

Lemma 7.3.12. $\gamma_s(P_3 \times P_{4k+2}) = 6k+3$, for $k \geq 1$.

Proof. We construct global secure sets of cardinality $6k+3$ for $P_3 \times P_{4k+2}$. In the specified global secure set configurations, the leftmost and rightmost columns each contains exactly one vertex in the set on the middle row. When $k = 1$, Figure 7.13 gives a global secure set of cardinality 9 for $P_3 \times P_6$. The validity of this configuration has been checked by an exhaustive search computer program (see Remark 7.3.1). For $k > 1$, first construct recursively a global secure set of cardinality $6(k-1)+3$ for $P_3 \times P_{4(k-1)+2}$, such that the leftmost and rightmost columns of $P_3 \times P_{4(k-1)+2}$ each contains exactly one vertex in the set on the middle row. Then, take the disjoint union of $P_3 \times P_{4(k-1)+2}$ and $P_3 \times P_4$ (with a global secure set for the latter given by Figure 7.14). Since the leftmost and rightmost columns of $P_3 \times P_{4(k-1)+2}$ and $P_3 \times P_4$ each contains exactly one vertex in the set on the middle row, all four columns are identical. Adding edges between corresponding vertices in the rightmost column of $P_3 \times P_{4(k-1)+2}$ and the leftmost column of $P_3 \times P_4$ produces the graph $P_3 \times P_{4k+2}$, with a global secure set of cardinality $6(k-1)+3+6 = 6k+3$. The leftmost column of $P_3 \times P_{4(k-1)+2}$ becomes the leftmost column of $P_3 \times P_{4k+2}$, and the rightmost column of $P_3 \times P_4$ becomes the rightmost column of $P_3 \times P_{4k+2}$, and, as a result, each contains exactly one vertex in the set on the middle row, and are identical. $\square$

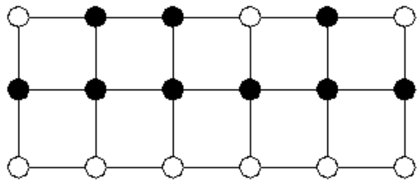

Figure 7.13: A global secure set of $P_3 \times P_6$. (See Remark 7.3.1.)

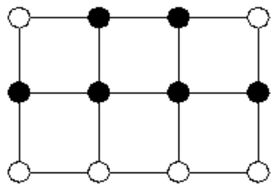

Figure 7.14: A global secure set of $P_3 \times P_4$.

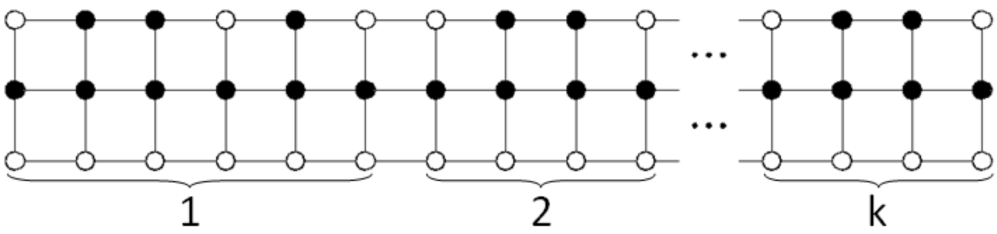

Figure 7.15: A global secure set of $P_3 \times P_{4k+2}$ as constructed in the proof of Lemma 7.3.12.

Figure 7.15 illustrates the global secure set pattern for $P_3 \times P_{4k+2}$ specified in the proof of Lemma 7.3.12. This configuration will be used in Lemma 7.4.5 for constructing global secure sets of $P_7 \times C_{4k+2}$.

Corollary 7.3.13. $\gamma_s(P_3 \times C_{4k+2}) = 6k + 3$, for $k \geq 1$.

Lemma 7.3.14. $\gamma_s(P_3 \times P_{4k+3}) = \lceil 3(4k + 3)/2 \rceil$, for $k \geq 0$.

Proof. We construct global secure sets of cardinality $\lceil 3(4k + 3)/2 \rceil = 6k + 5$ for $P_3 \times P_{4k+3}, k \geq 0$. Consider the global secure set of the graph shown in Figure 7.16. Removing the degree one attacker gives a global secure set of cardinality $5 = 6 \times 0 + 5$ for $P_3 \times P_3$ (Observation 7.1.3, Part 4). For $P_3 \times P_7$, consider the disjoint union of $P_3 \times P_4$ in Figure 7.14 and the graph in Figure 7.16. Identify the degree one attacker in Figure 7.16 with the top right attacker of $P_3 \times P_4$ (Observation 7.1.3, Part 3). This produces the graph shown in Figure 7.17, with a global secure set of cardinality $6 + 5 = 11$. Adding the appropriate edges (Observation 7.1.3, Part 2) produces $P_3 \times P_7$, with the same global secure set, as shown in Figure 7.18.

For $k > 1$, consider the disjoint union of $P_3 \times P_{4(k-1)+3}$ and $P_3 \times P_4$ in Figure 7.14. The rightmost column of $P_3 \times P_{4(k-1)+3}$ is identical to the leftmost column of $P_3 \times P_4$. Adding edges between corresponding vertices in these two columns produces $P_3 \times P_{4k+3}$, with a global secure set of cardinality $6(k - 1) + 5 + 6 = 6k + 5$. $\square$

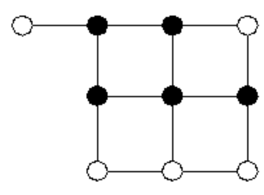

Figure 7.16: A graph and one of its global secure sets marked in black.

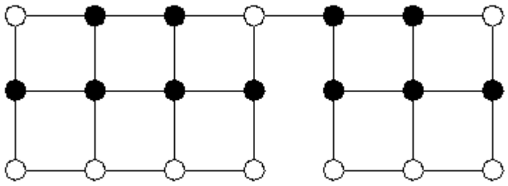

Figure 7.17: Identifying the degree one attacker in Figure 7.16 with the top right attacker in Figure 7.14.

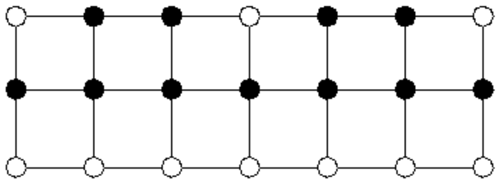

Figure 7.18: A global secure set of $P_3 \times P_7$ constructed by the proof of Lemma 7.3.14.

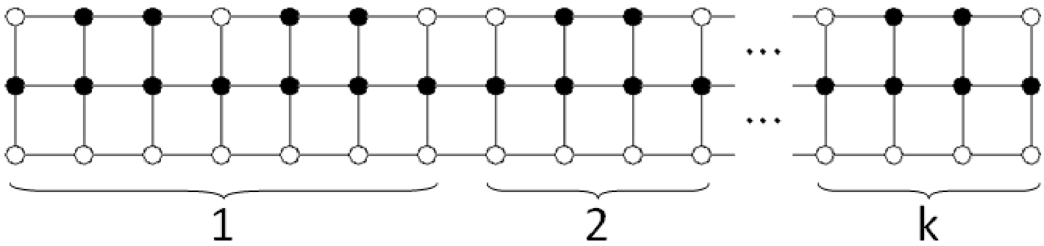

Figure 7.19: A global secure set of $P_3 \times P_{4k+3}$ as constructed by the proof of Lemma 7.3.14.

Figure 7.19 illustrates the global secure set pattern for $P_3 \times P_{4k+3}, k \geq 1$, specified in the proof of Lemma 7.3.14.

Corollary 7.3.15. $\gamma_s(P_3 \times C_{4k+3}) = \lceil 3(4k+3)/2 \rceil$, for $k \geq 1$.

Corollary 7.3.16. $\gamma_s(P_3 \times P_m) = \lceil 3m/2 \rceil$, for $m \geq 2$.

Proof. All cases based on the value of m are covered, as shown in the following list.

1. $m = 2$. By Lemma 7.3.5 and Observation 7.3.4.

2. $m \in \{4k : k \geq 1\}$. By Lemma 7.3.7.

3. $m \in \{4k + 1 : k \geq 1\}$. By Lemma 7.3.10.

4. $m \in \{4k + 2 : k \geq 1\}$. By Lemma 7.3.12.

5. $m \in \{4k + 3 : k \geq 0\}$. By Lemma 7.3.14.

□

Next, we treat cases with $m \equiv 1 \pmod 4$. Lemmas 7.3.17, 7.3.19 and 7.3.21 treat cases based on different values of $n \pmod 4$ with $m \equiv 1 \pmod 4$. Then, Corollary 7.3.23 collects the results and shows that $\gamma_s(P_n \times P_{4k+1}) = \lceil n(4k + 1)/2 \rceil$.

Lemma 7.3.17. $\gamma_s(P_{4h+1} \times P_{4k+1}) = \lceil (4h + 1)(4k + 1)/2 \rceil$, for $h, k \geq 1$.

Proof. Figure 7.10 shows a global secure set of cardinality 5 for $P_2 \times P_5$. Consider the disjoint union of two copies of $P_2 \times P_5$. The top rows of the two $P_2 \times P_5$ are identical. By Observation 7.1.3 (Part 2), adding edges between corresponding vertices on the top rows of the two $P_2 \times P_5$ produces $P_4 \times P_5$, with a global secure set of cardinality $2 \times 5 = 10$, as shown in Figure 7.20. In this $P_4 \times P_5$, the leftmost and rightmost columns contain no vertex in the set, and the top and bottom rows are identical.

Next, for $k = 1$, construct global secure sets of cardinality $\lceil (4h + 1) \cdot 5/2 \rceil = 10h + 3$ for $P_{4h+1} \times P_5$, $h \geq 1$. These global secure sets will be constructed in a way such that the leftmost and rightmost columns contain no vertex in the set, and the top and bottom rows are identical to the top row of $P_4 \times P_5$ (in Figure 7.20). For $k = 1$ and $h = 1$, consider the disjoint union of $P_2 \times P_5$ and $P_3 \times P_5$ (with a global secure set of cardinality 8 for the latter shown in Figure 7.21). The top row of $P_2 \times P_5$ is identical to the top row of $P_3 \times P_5$. By Observation 7.1.3 (Part 2), adding edges between corresponding vertices on these top rows produces $P_5 \times P_5$, with a global secure set of cardinality $5 + 8 = 13 = 10 \cdot 1 + 3$, as shown in Figure 7.22. In the $P_5 \times P_5$, the leftmost and rightmost columns contain no vertex in the set, and the top and bottom rows are identical to the top row of $P_4 \times P_5$ in Figure 7.20.

For $k = 1$ and $h > 1$, first construct recursively a global secure set of cardinality $10(h - 1) + 3$ for $P_{4(h-1)+1} \times P_5$, such that the leftmost and rightmost columns of $P_{4(h-1)+1} \times P_5$ contain no vertex in the set, and the top and bottom rows of $P_{4(h-1)+1} \times P_5$ are identical to the top row of $P_4 \times P_5$ (in Figure 7.20). Then, take the disjoint union of $P_{4(h-1)+1} \times P_5$ and $P_4 \times P_5$. The leftmost and rightmost columns of $P_{4(h-1)+1} \times P_5$ and $P_4 \times P_5$ contain no vertex in the set, and the top and bottom rows of $P_{4(h-1)+1} \times P_5$ and $P_4 \times P_5$ are identical. By Observation 7.1.3 (Part 2), adding edges between corresponding vertices on the bottom row of $P_{4(h-1)+1} \times P_5$ and the top row of $P_4 \times P_5$ produces $P_{4h+1} \times P_5$, with a global secure set of cardinality $10(h - 1) + 3 + 10 = 10h + 3$. The top row of $P_{4(h-1)+1} \times P_5$ becomes the top row of $P_{4h+1} \times P_5$, and the bottom row of $P_4 \times P_5$ becomes the bottom row of $P_{4h+1} \times P_5$. So, the top and bottom rows of $P_{4h+1} \times P_5$ are both identical to the top row of $P_4 \times P_5$. The leftmost and rightmost columns of $P_{4(h-1)+1} \times P_5$ and $P_4 \times P_5$ become the leftmost and rightmost columns of $P_{4h+1} \times P_5$, and each contains no vertex in the set.

Finally, construct global secure sets of cardinality $\lceil (4h + 1)(4k + 1)/2 \rceil = 8hk+2h+2k+1$ for $P_{4h+1} \times P_{4k+1}$, $h, k \geq 1$. For $k = 1$, global secure sets of cardinality $10h+3 = (8h \times 1)+2h+$

$(2\times1)+1$ for $P_{4h+1}\times P_5$ are given in the previous paragraphs. For $k>1$, consider the disjoint union of $P_{4h+1}\times P_5$ and $P_{4h+1}\times P_{4(k-1)}$ (with a global secure set for the latter constructed by the proof of Lemma 7.3.7). The rightmost column of $P_{4h+1}\times P_5$ and the leftmost column of $P_{4h+1}\times P_{4(k-1)}$ both contain no vertex in the set, and are identical. By Observation 7.1.3 (Part 2), adding edges between corresponding vertices in these two columns produces $P_{4h+1}\times P_{4k+1}$, with a global secure set of cardinality $10h+3+2(k-1)(4h+1)=8hk+2h+2k+1$. □

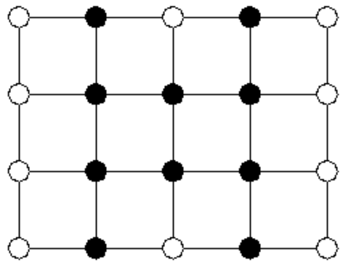

Figure 7.20: A global secure set of $P_4\times P_5$.

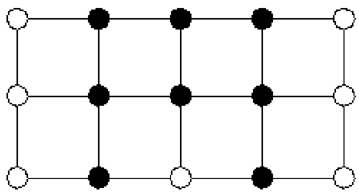

Figure 7.21: A global secure set of $P_3\times P_5$ with 8 vertices. Note that this configuration may be obtained by removing the degree one attacker from the configuration in Figure 7.11 (Observation 7.1.3, Part 4).

Refer to Figure 7.23 for an illustration of the global secure set pattern for $P_{4h+1}\times P_{4k+1}$ specified in the proof of Lemma 7.3.17.

Corollary 7.3.18. $\gamma_s(P_{4h+1}\times C_{4k+1})=\gamma_s(C_{4h+1}\times C_{4k+1})=\lceil(4h+1)(4k+1)/2\rceil$, for $h,k\geq1$.

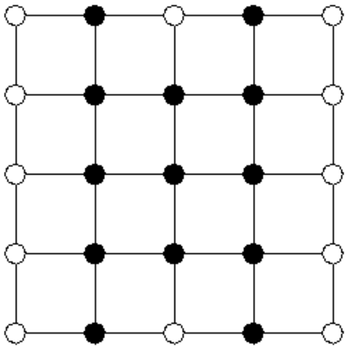

Figure 7.22: A global secure set of $P_5 \times P_5$.

Lemma 7.3.19. $\gamma_s(P_{4h+2} \times P_{4k+1}) = (2h+1)(4k+1)$, for $h, k \geq 1$.

Proof. Similar to the proof of Lemma 7.3.17, we first construct global secure sets of cardinality $(2h+1) \times 5$ for $P_{4h+2} \times P_5$, $h \geq 1$. For $h = 1$, Figure 7.24 shows a global secure set of cardinality $15 = (2 \cdot 1 + 1) \times 5$ for $P_6 \times P_5$. For $h > 1$, consider the disjoint union of $P_{4(h-1)+2} \times P_5$ and $P_4 \times P_5$ in Figure 7.20. The bottom row of $P_{4(h-1)+2} \times P_5$ is identical to the top row of $P_4 \times P_5$. Adding edges between the corresponding vertices on these two rows produces $P_{4h+2} \times P_5$, with a global secure set of cardinality $(2(h-1)+1) \times 5 + 10 = (2h+1) \times 5$.

Next, for $k = 1$, global secure sets of cardinality $(2h+1) \times 5 = (2h+1)(4 \times 1 + 1)$ for $P_{4h+2} \times P_5$ are given in the previous paragraph. For $k > 1$, consider the disjoint union of $P_{4h+2} \times P_5$ and $P_{4h+2} \times P_{4(k-1)}$ (with a global secure set for the latter constructed by the proof of Lemma 7.3.7). The rightmost column of $P_{4h+2} \times P_5$ and the leftmost column of $P_{4h+2} \times P_{4(k-1)}$ are identical. Adding edges between corresponding vertices in these two columns produces $P_{4h+2} \times P_{4k+1}$, with a global secure set of cardinality $(2h+1) \times 5 + 2(k-1)(4h+2) = (2h+1)(4k+1)$. $\square$

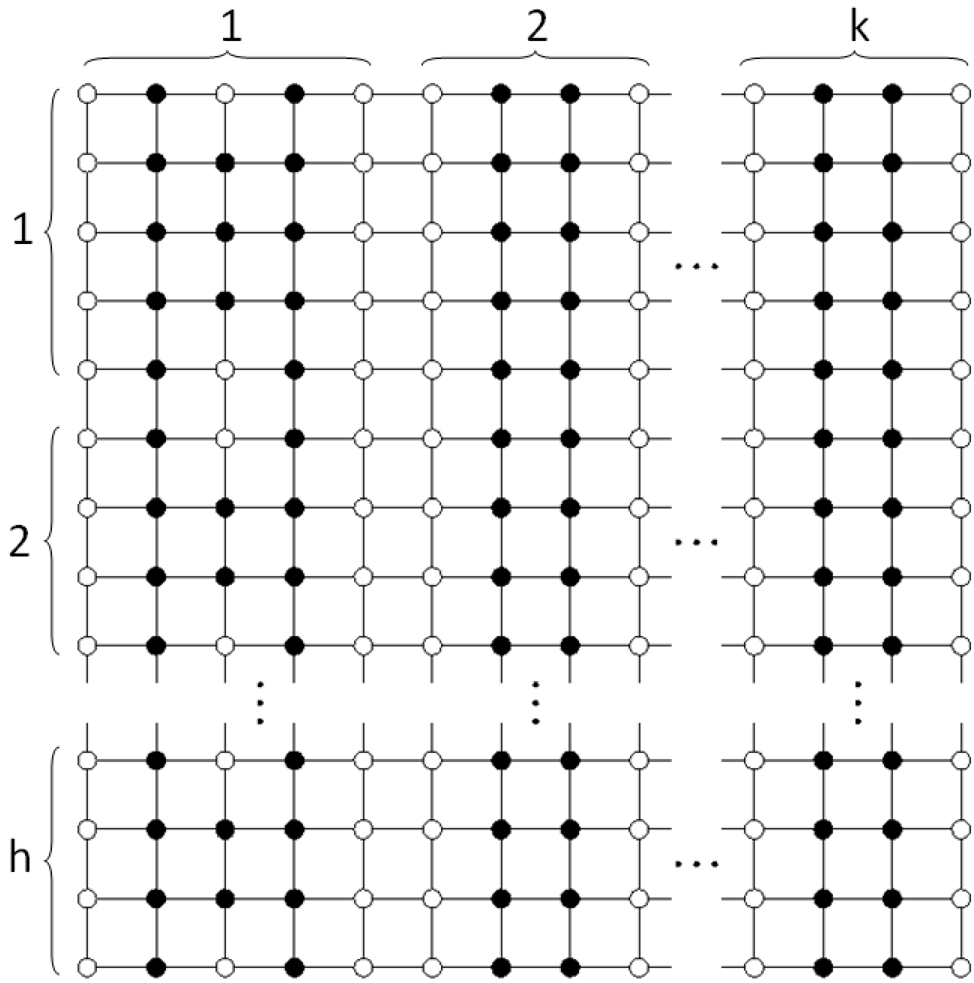

Figure 7.23: A global secure set of $P_{4h+1} \times P_{4k+1}$.

Figure 7.25 illustrates the global secure set pattern of $P_{4h+2} \times P_{4k+1}$ specified in the proof of Lemma 7.3.19.

Corollary 7.3.20. $\gamma_s(P_{4h+2} \times C_{4k+1}) = \gamma_s(C_{4h+2} \times P_{4k+1}) = \gamma_s(C_{4h+2} \times C_{4k+1}) = (2h+1)(4k+1)$, for $h, k \geq 1$.

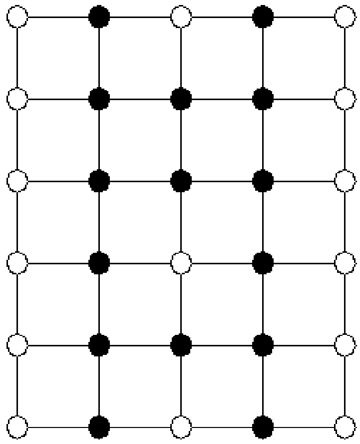

Figure 7.24: A global secure set of $P_6 \times P_5$. (See Remark 7.3.1.)

Lemma 7.3.21. $\gamma_s(P_{4h+3} \times P_{4k+1}) = \lceil (4h+3)(4k+1)/2 \rceil$, for $h, k \geq 1$.

Proof. First, consider the disjoint union of $P_4 \times P_5$ in Figure 7.20 and the graph in Figure 7.11. Identify the middle attacker on the bottom row of $P_4 \times P_5$ with the degree one attacker in Figure 7.11 (Observation 7.1.3, Part 3). This results in the graph shown in Figure 7.26, with a global secure set of cardinality $10 + 8 = 18$. Adding the appropriate edges produces $P_7 \times P_5$ with the same global secure set, as shown in Figure 7.27.

Next, similar to the proof of Lemma 7.3.17, construct global secure sets of cardinality $\lceil 5(4h+3)/2 \rceil = 10h + 8$ for $P_{4h+3} \times P_5$, $h \geq 1$. For $h = 1$, a global secure set of cardinality $18 = (10 \times 1) + 8$ for $P_7 \times P_5$ is given by the previous paragraph. For $h > 1$, consider the disjoint union of $P_{4(h-1)+3} \times P_5$ and $P_4 \times P_5$ (in Figure 7.20). The bottom row of $P_{4(h-1)+3} \times P_5$ is identical to the top row of $P_4 \times P_5$. Adding edges between corresponding vertices on these two rows produces $P_{4h+3} \times P_5$, with a global secure set of cardinality $10(h-1) + 8 + 10 = 10h + 8$.

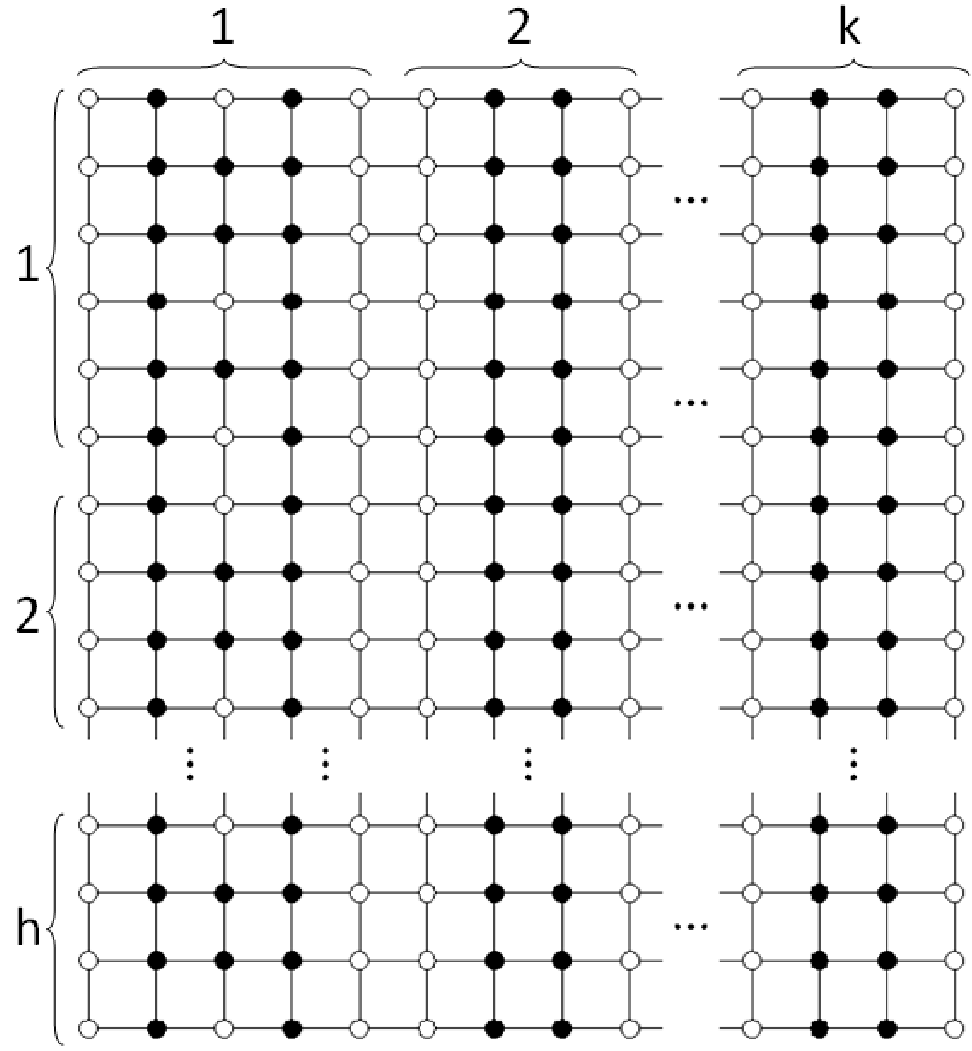

Figure 7.25: A global secure set of $P_{4h+2} \times P_{4k+1}$ as constructed by proof of Lemma 7.3.19.

Finally, construct global secure sets of cardinality $\lceil (4h+3)(4k+1)/2 \rceil = 8hk+2h+6k+2$ for $P_{4h+3} \times P_{4k+1}$, $h, k \geq 1$. For $k = 1$, global secure sets of cardinality $10h + 8 = (8h \times 1) + 2h + (6 \times 1) + 2$ for $P_{4h+3} \times P_5$ are given by the previous paragraph. For $k > 1$, consider the disjoint union of $P_{4h+3} \times P_5$ and $P_{4h+3} \times P_{4(k-1)}$ (with a global secure set for the latter constructed by the proof of Lemma 7.3.7). The rightmost column of $P_{4h+3} \times P_5$

is identical to the leftmost column of $P_{4h+3} \times P_{4(k-1)}$. Adding edges between corresponding vertices in these two columns produces $P_{4h+3} \times P_{4k+1}$, with a global secure set of cardinality $10h + 8 + 2(k-1)(4h+3) = 8hk + 2h + 6k + 2$. □

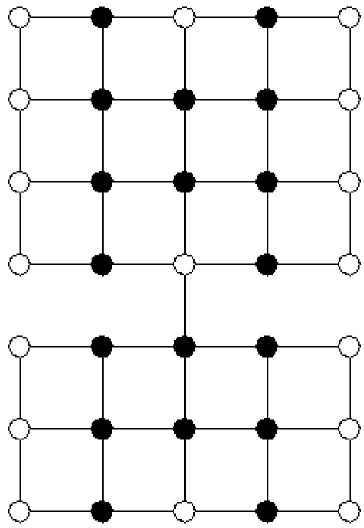

Figure 7.26: Identifying the degree one attacker in Figure 7.11 with the middle attacker on the bottom row in Figure 7.20

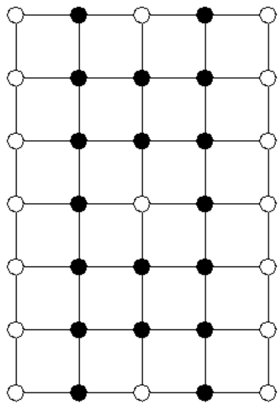

Figure 7.27: A global secure set of $P_7 \times P_5$.

Figure 7.28 illustrates the global secure set pattern of $P_{4h+3} \times P_{4k+1}$ specified in the proof of Lemma 7.3.21.

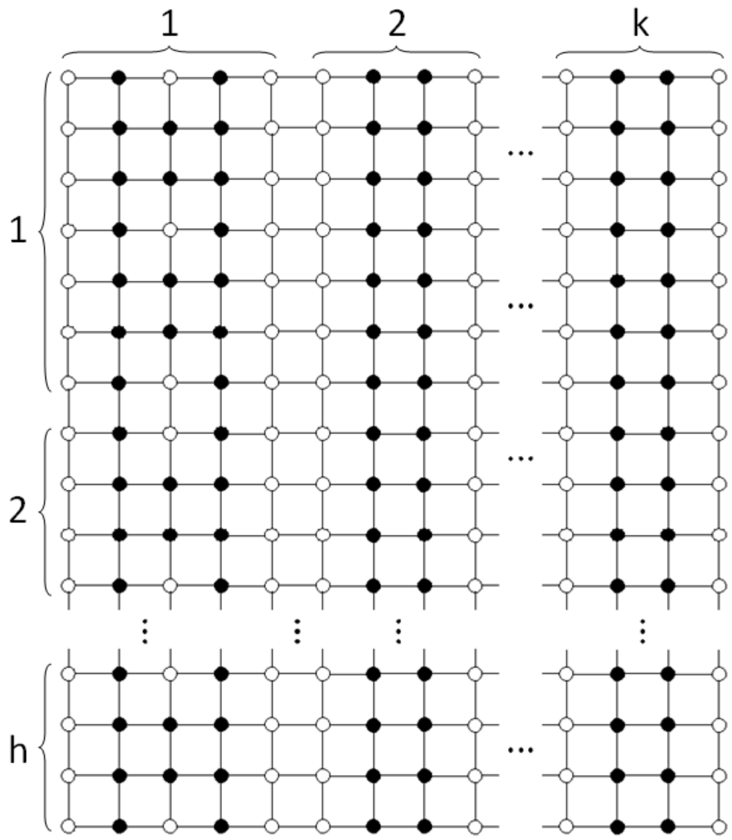

Figure 7.28: A global secure set of $P_{4h+3} \times P_{4k+1}$ as constructed by the proof of Lemma 7.3.21.

Corollary 7.3.22. Let $h, k \geq 1$. Then, $\gamma_s(P_{4h+3} \times C_{4k+1}) = \gamma_s(C_{4h+3} \times P_{4k+1}) = \gamma_s(C_{4h+3} \times C_{4k+1}) = \lceil (4h+3)(4k+1)/2 \rceil$

Corollary 7.3.23. $\gamma_s(P_n \times P_{4k+1}) = \lceil n(4k+1)/2 \rceil$, for $n \geq 2, k \geq 1$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n = 2$. By Lemma 7.3.5.

2. $n = 3$. By Corollary 7.3.16.

3. $n \in \{4h : h \geq 1\}$. By Lemma 7.3.7 and Observation 7.3.4.

4. $n \in \{4h + 1 : h \geq 1\}$. By Lemma 7.3.17.

5. $n \in \{4h + 2 : h \geq 1\}$. By Lemma 7.3.19.

6. $n \in \{4h + 3 : h \geq 1\}$. By Lemma 7.3.21.

□

The next set of cases is $m \equiv 2 \pmod 4$. Note that since we already dealt with cases $m \equiv 0 \pmod 4$ and $m \equiv 1 \pmod 4$, by Observation 7.3.4, those configurations are also valid for cases $n \equiv 0 \pmod 4$ and $n \equiv 1 \pmod 4$. Thus, we only need to treat the cases $P_{4h+2} \times P_{4k+2}$ and $P_{4h+3} \times P_{4k+2}$. Lemma 7.3.24 treats the case $P_{4h+2} \times P_{4k+2}$, and the case $P_{4h+3} \times P_{4k+2}$ is further broken down into two sub-cases, considered in Lemmas 7.3.26 ($P_{4h+2} \times P_{4k+3}$ for $k \geq 2$) and 7.3.28 ($P_{4h+2} \times P_7$). Then, Corollary 7.3.30 collects the results and shows that $\gamma_s(P_n \times P_{4k+2}) = n(2k + 1)$.

Lemma 7.3.24. $\gamma_s(P_{4h+2} \times P_{4k+2}) = 2(2h + 1)(2k + 1)$, for $h, k \geq 1$.

Proof. Consider the disjoint union of two copies of $P_2 \times P_6$ shown in Figure 7.29. The top rows of these two $P_2 \times P_6$ are identical. Adding edges between corresponding vertices on these two rows produces $P_4 \times P_6$, with a global secure set of cardinality 12, as shown in Figure 7.30.

Next, for $k = 1$, construct global secure sets of cardinality $2 \times (2h + 1) \times (2 \cdot 1 + 1) = 6 \times (2h + 1)$ for $P_{4h+2} \times P_6$, $h \geq 1$. For $k = 1$ and $h = 1$, a global secure set of cardinality

$18 = 6 \times (2 \cdot 1 + 1)$ for $P_6 \times P_6$ is shown in Figure 7.31. For $k = 1$ and $h > 1$, consider the disjoint union of $P_{4(h-1)+2} \times P_6$ and $P_4 \times P_6$ (in Figure 7.30). The bottom row of $P_{4(h-1)+2} \times P_6$ is identical to the top row of $P_4 \times P_6$. Adding edges between corresponding vertices on these two rows produces $P_{4h+2} \times P_6$, with a global secure set of cardinality $6 \times (2(h-1)+1) + 12 = 6(2h+1)$.

Finally, construct global secure sets of cardinality $2(2h+1)(2k+1)$ for $P_{4h+2} \times P_{4k+2}$, $h, k \geq 1$. For $k = 1$, global secure sets of cardinality $6(2h+1) = 2(2h+1)(2 \cdot 1 + 1)$ for $P_{4h+2} \times P_6$ are given in the previous paragraph. For $k > 1$, consider the disjoint union of $P_{4h+2} \times P_6$ and $P_{4h+2} \times P_{4(k-1)}$ (with a global secure set for the latter constructed by the proof of Lemma 7.3.7). The rightmost column of $P_{4h+2} \times P_6$ is identical to the leftmost column of $P_{4h+2} \times P_{4(k-1)}$. Adding edges between corresponding vertices in these two columns produces $P_{4h+2} \times P_{4k+2}$, with a global secure set of cardinality $6(2h+1) + 2(k-1)(4h+2) = 2(2h+1)(2k+1)$. $\square$

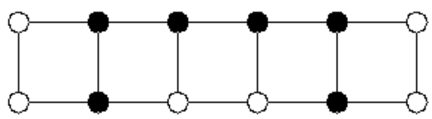

Figure 7.29: A global secure set of $P_2 \times P_6$.

Figure 7.32 illustrates the global secure set pattern of $P_{4h+2} \times P_{4k+2}$ specified in the proof of Lemma 7.3.24.

Corollary 7.3.25. $\gamma_s(P_{4h+2} \times C_{4k+2}) = \gamma_s(C_{4h+2} \times C_{4k+2}) = 2(2h+1)(2k+1)$, for $h, k \geq 1$.

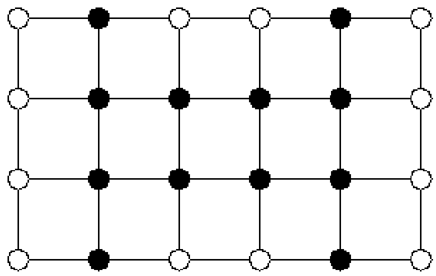

Figure 7.30: A global secure set of $P_4 \times P_6$.

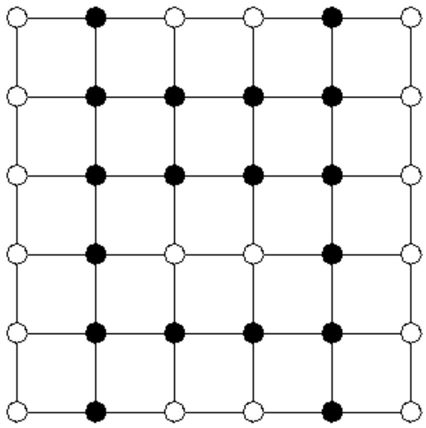

Figure 7.31: A global secure set of $P_6 \times P_6$. (See Remark 7.3.1.)

Lemma 7.3.26. $\gamma_s(P_{4h+2} \times P_{4k+3}) = (2h+1)(4k+3)$, for $h \geq 1, k \geq 2$.

Proof. For $k = 2$, consider the disjoint union of $P_{4h+2} \times P_5$ (with a global secure set constructed by the proof of Lemma 7.3.19) and $P_{4h+2} \times P_6$ (with a global secure set constructed by the proof of Lemma 7.3.24). The rightmost column of $P_{4h+2} \times P_5$ is identical to the leftmost column of $P_{4h+2} \times P_6$. Adding edges between corresponding vertices in these two columns produces $P_{4h+2} \times P_{11}$, with a global secure set of cardinality $(2h+1)(4\times1+1)+2(2h+1)(2\times1+1) = 5(2h+1)+6(2h+1) = 11(2h+1) = (2h+1)(4\times2+3)$.

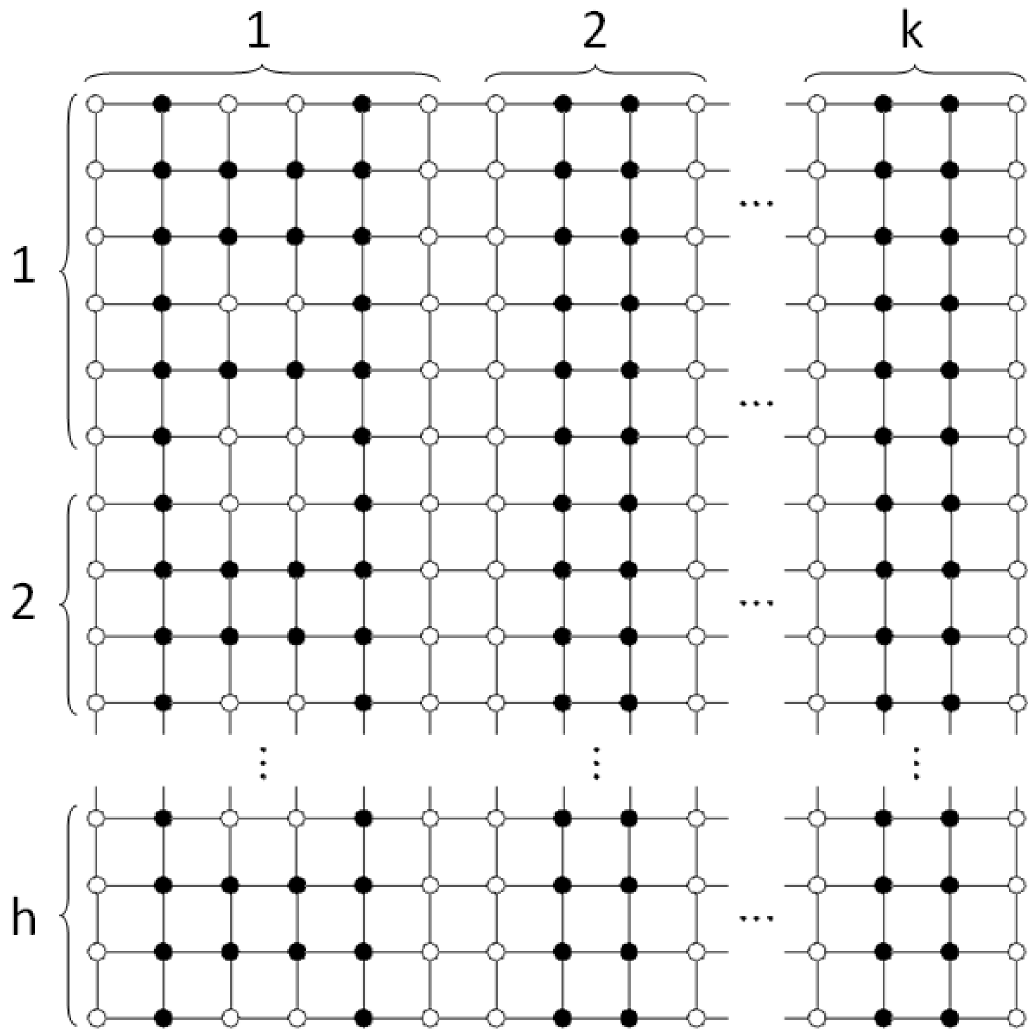

Figure 7.32: A global secure set of $P_{4h+2} \times P_{4k+2}$ as constructed by the proof of Lemma 7.3.24.

For $k > 2$, consider the disjoint union of $P_{4h+2} \times P_{11}$ and $P_{4h+2} \times P_{4(k-2)}$ (with a global secure set for the latter constructed by the proof of Lemma 7.3.7). The rightmost column of $P_{4h+2} \times P_{11}$ is identical to the leftmost column of $P_{4h+2} \times P_{4(k-2)}$. Adding edges between corresponding vertices in these two columns produces $P_{4h+2} \times P_{4k+3}$, with a global secure set of cardinality $11(2h+1) + 2(k-2)(4h+2) = (2h+1)(4k+3)$. $\square$

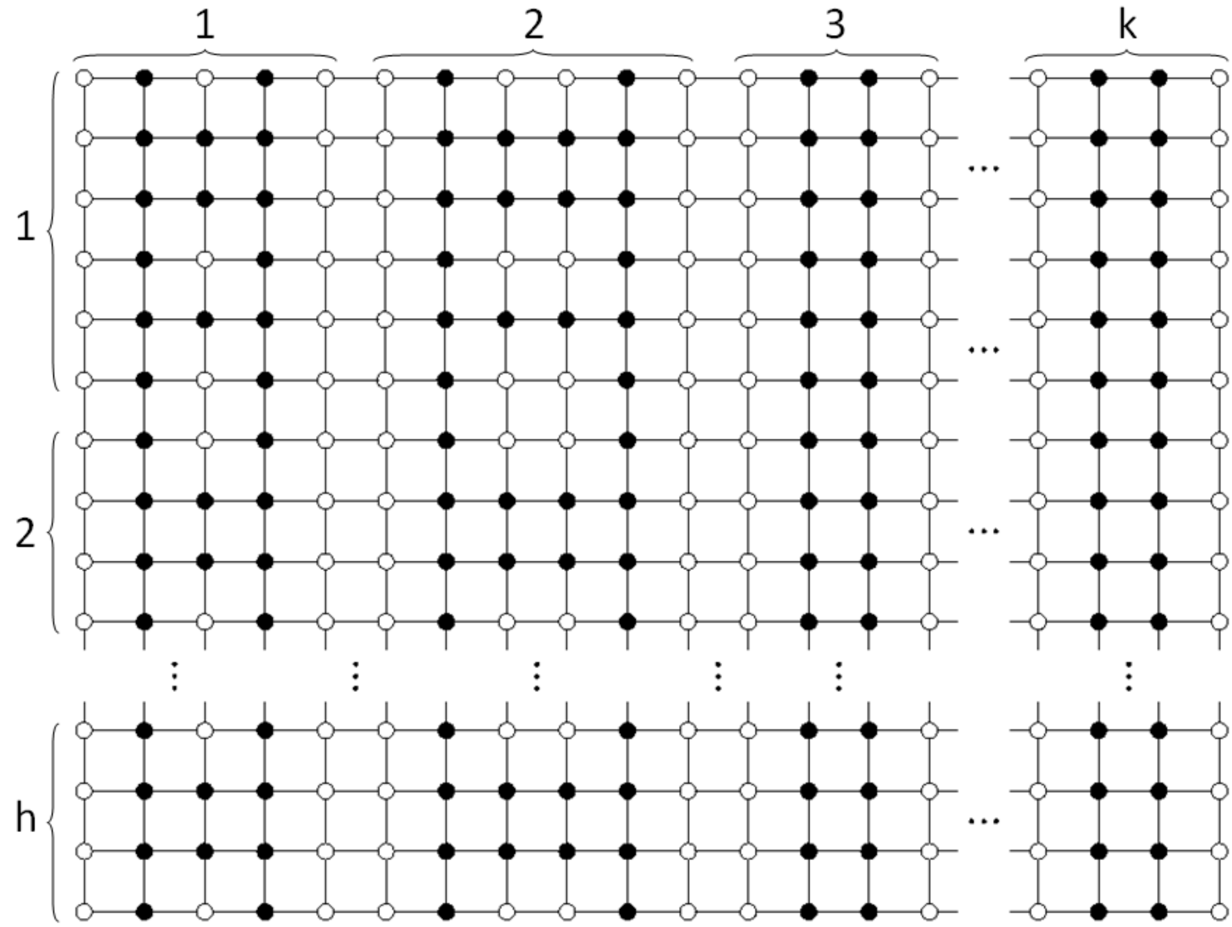

Figure 7.33: A global secure set of $P_{4h+2} \times P_{4k+3}$ as constructed by the proof of Lemma 7.3.26.

Figure 7.33 illustrates the global secure set pattern of $P_{4h+2} \times P_{4k+3}$ specified in the proof of Lemma 7.3.26.

Corollary 7.3.27.

$\gamma_s(P_{4h+2} \times C_{4k+3}) = \gamma_s(C_{4h+2} \times P_{4k+3}) = \gamma_s(C_{4h+2} \times C_{4k+3}) = (2h+1)(4k+3)$ for $h \geq 1, k \geq 2$.

Lemma 7.3.28. $\gamma_s(P_{4h+2} \times P_7) = 7(2h+1)$, for $h \geq 0$.

Proof. When $h = 0$, Figure 7.34 shows a global secure set of cardinality 7 for $P_2 \times P_7$. Consider the disjoint union of two copies of $P_2 \times P_7$. The top rows of the two $P_2 \times P_7$ are identical. Adding edges between corresponding vertices on these two rows produces $P_4 \times P_7$, with a global secure set of cardinality $2 \times 7 = 14$, as shown in Figure 7.35.

Next, for $h > 0$, consider the disjoint union of $P_{4(h-1)+2} \times P_7$ and $P_4 \times P_7$. The bottom row of $P_{4(h-1)+2} \times P_7$ is identical to the top row of $P_4 \times P_7$. Adding edges between corresponding vertices on these two rows produces $P_{4h+2} \times P_7$, with a global secure set of cardinality $7 \times (2 \cdot (h-1) + 1) + 14 = 7(2h+1)$. □

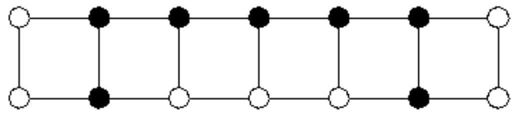

Figure 7.34: A global secure set of $P_2 \times P_7$.

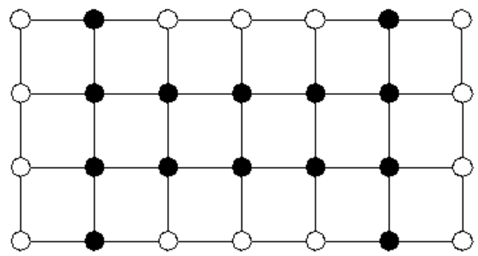

Figure 7.35: A global secure set of $P_4 \times P_7$.

Figure 7.36 illustrates the global secure set pattern of $P_{4h+2} \times P_7$ specified in the proof of Lemma 7.3.28.

Corollary 7.3.29. $\gamma_s(P_{4h+2} \times C_7) = 7(2h+1)$, for $h \geq 0$.

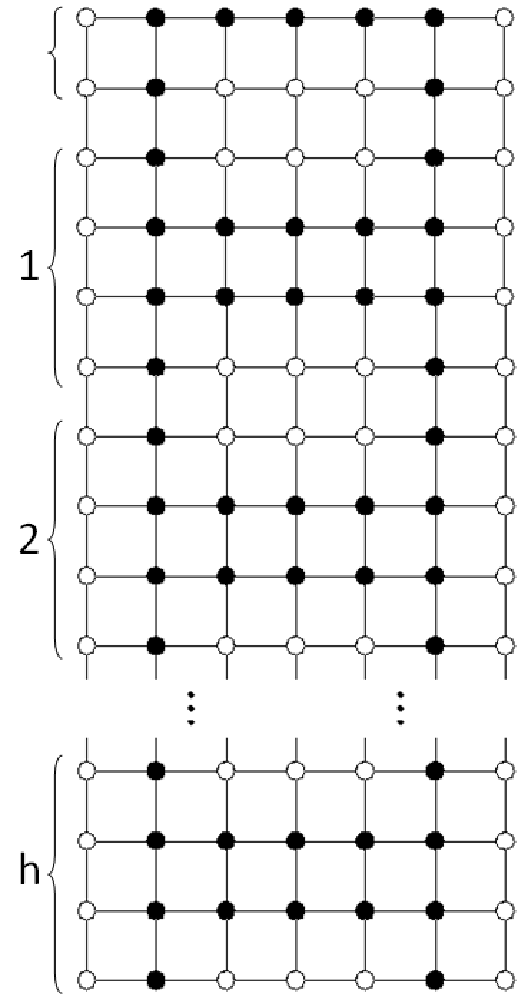

Figure 7.36: A global secure set of $P_{4h+2} \times P_7$ as constructed by the proof of Lemma 7.3.28.

Notice for the case $P_{4h+2} \times P_{4k+3}$, we considered two sub-cases $k = 1$ (Lemma 7.3.28) and $k \geq 2$ (Lemma 7.3.26). This is unlike the other dimensions we have dealt with so far. For $P_{4h+1} \times P_{4k+1}$, $P_{4h+2} \times P_{4k+1}$, $P_{4h+3} \times P_{4k+1}$ and $P_{4h+2} \times P_{4k+2}$, we presented global secure sets of small graphs with the specified dimensions ($P_5 \times P_5$ (Fig. 7.22), $P_6 \times P_5$ (Fig. 7.24), $P_7 \times P_5$ (Fig. 7.27) and $P_6 \times P_6$ (Fig. 7.31), respectively), and then constructed global secure sets for larger graphs using the small graphs as initial building blocks. These small graphs all share the properties that *the leftmost and rightmost columns contain no vertex in the global secure set, and the top and bottom rows are identical.*

But, unfortunately the dimension 6×7 does not have any global secure set configurations sharing the properties stated above. In other words, if the vertices of the leftmost and rightmost columns of $P_6 \times P_7$ (or $P_7 \times P_6$) are not included in a global secure set, and the top and bottom rows must be identical, then the global secure set would contain strictly more than half of the vertices of the graph. Consider $P_7 \times P_6$ and let $c_1, c_2, \ldots, c_6$ be its columns. If the leftmost and rightmost columns (c_1 and c_6) are not in a global secure set of $P_7 \times P_6$, then all vertices of c_2 and c_5 must be in the set in order to dominate the vertices of c_1 and c_6. If a global secure set of cardinality exactly 21 is to exist, among the vertices of c_3 and c_4, we must include exactly 7 vertices. If column c_3 contains less than 4 vertices in the global secure set, then the vertices in c_2 form a witness set (i.e., among the closed neighborhood of c_2, there are more attackers than defenders). So, c_3 must contain at least 4 vertices in the global secure set. Similarly, c_4 must contain at least 4 vertices in the set or otherwise c_5 is a witness. This means we must include at least 8 additional vertices in order to form a global secure set. So, a global secure set with cardinality 21 for $P_7 \times P_6$ cannot exist if the columns c_1 and c_6 are excluded from the set.

Similarly, consider the graph $P_6 \times P_7$ and let $c_1, c_2, \ldots, c_7$ be its columns. If vertices of the leftmost and rightmost columns (c_1 and c_7) are not in a global secure set, all vertices of c_2 and c_6 must be in the set. Then, if a global secure set of cardinality exactly 21 is to exist, we must include exactly 9 vertices among columns c_3, c_4 and c_5. A computer program was developed which enumerates all possible subsets of 9 vertices among the 18 vertices in c_3, c_4 and c_5. The program then checks whether each of the resultant configurations (including all vertices of c_2 and c_6) is a global secure set of $P_6 \times P_7$ with identical top and bottom rows. Verification of security is done by the procedure described in Remark 7.3.1 (also described at the end of Section 3.3). The program confirmed that *none* of the $\binom{18}{9}$ subsets can be added to c_2 and c_6 to form a global secure set of $P_6 \times P_7$ with identical top and bottom rows.

Since a global secure set configuration with identical top and bottom rows, and excluding the leftmost and rightmost columns, is not available, when treating the case $P_{4h+2} \times P_{4k+3}$, we considered the two sub-cases aforementioned. Furthermore, note that global secure sets for $C_{4h+2} \times C_7$ of cardinality $7(2h + 1)$ cannot be obtained as a simple extension of the $P_{4h+2} \times P_7$ configuration (in Figure 7.3.28). This is because the top and bottom rows of the $P_{4h+2} \times P_7$ configuration are not identical, and Observation 7.1.3 (Part 2) does not apply. In Section 7.5, two additional global secure set configurations (Figures 7.50 and 7.51) are discovered and used for the construction of minimum global secure sets of $C_{4h+2} \times C_7$.

Corollary 7.3.30 collects results and establishes the global security number of $P_n \times P_{4k+2}$.

Corollary 7.3.30. $\gamma_s(P_n \times P_{4k+2}) = n(2k + 1)$, for $n \geq 2, k \geq 0$.

Proof.

1. $k = 0$. By Lemma 7.3.5 and Observation 7.3.4.

2. $k \geq 1$.

(a) $n = 2$. By Lemma 7.3.5.

(b) $n = 3$. By Corollary 7.3.16.

(c) $n \in \{4h : h \geq 1\}$. By Lemma 7.3.7 and Observation 7.3.4.

(d) $n \in \{4h + 1 : h \geq 1\}$. By Corollary 7.3.23 and Observation 7.3.4.

(e) $n \in \{4h + 2 : h \geq 1\}$. By Lemma 7.3.24.

(f) $n \in \{4h + 3 : h \geq 1\}$.

i. $n = 7$. By Lemma 7.3.28 and Observation 7.3.4.

ii. $n \in \{4h + 3 : h \geq 2\}$. By Lemma 7.3.26 and Observation 7.3.4.

□

Next, Lemma 7.3.31 treats the case $P_{4h+3} \times P_{4k+3}$ and Corollary 7.3.33 collects results for $P_n \times P_{4k+3}$. Finally, Theorem 7.3.34 collects results for $P_n \times P_m$ and concludes this section.

Lemma 7.3.31. $\gamma_s(P_{4h+3} \times P_{4k+3}) = \lceil (4h + 3)(4k + 3)/2 \rceil$, for $h, k \geq 1$.

Proof. First, for $k = 1$, construct global secure sets of cardinality $\lceil 7 \cdot (4h + 3)/2 \rceil = 14h+11$ for $P_{4h+3} \times P_7$, $h \geq 1$. When $k = 1$ and $h = 1$, a global secure set of cardinality $25 = (14 \times 1) + 11$ for $P_7 \times P_7$ is given in Figure 7.37. For $h > 1$, consider the disjoint union of $P_{4(h-1)+3} \times P_7$ and $P_4 \times P_7$ in Figure 7.35. The bottom row of $P_{4(h-1)+3} \times P_7$ is identical to the top row of $P_4 \times P_7$. Adding edges between corresponding vertices on these two rows produces $P_{4h+3} \times P_7$, with a global secure set of cardinality $14(h - 1) + 11 + 14 = 14h + 11$.

Then, construct global secure sets of cardinality $\lceil (4h+3)(4k+3)/2 \rceil = 8hk+6h+6k+5$ for $P_{4h+3} \times P_{4k+3}$, $h, k \geq 1$. For $k = 1$, global secure sets of cardinality $14h + 11 = (8h \times 1) + 6h + (6 \times 1) + 5$ for $P_{4h+3} \times P_7$ are given in the previous paragraph. For $k > 1$, consider the disjoint union of $P_{4h+3} \times P_7$ and $P_{4h+3} \times P_{4(k-1)}$ (with a global secure set for the latter constructed by the proof of Lemma 7.3.7). The rightmost column of $P_{4h+3} \times P_7$ is identical to the leftmost column of $P_{4h+3} \times P_{4(k-1)}$. Adding edges between corresponding vertices in these two columns produces $P_{4h+3} \times P_{4k+3}$, with a global secure set of cardinality $14h + 11 + 2(k-1)(4h+3) = 8hk + 6h + 6k + 5$. $\square$

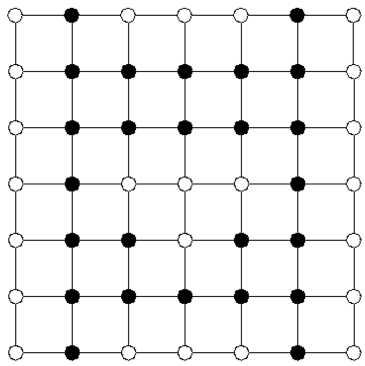

Figure 7.37: A global secure set of $P_7 \times P_7$. (See Remark 7.3.1.)

Figure 7.38 illustrates the global secure set pattern of $P_{4h+3} \times P_{4k+3}$ specified in the proof of Lemma 7.3.31.

Corollary 7.3.32. $\gamma_s(P_{4h+3} \times C_{4k+3}) = \gamma_s(C_{4h+3} \times C_{4k+3}) = \lceil (4h+3)(4k+3)/2 \rceil$, for $h, k \geq 1$.

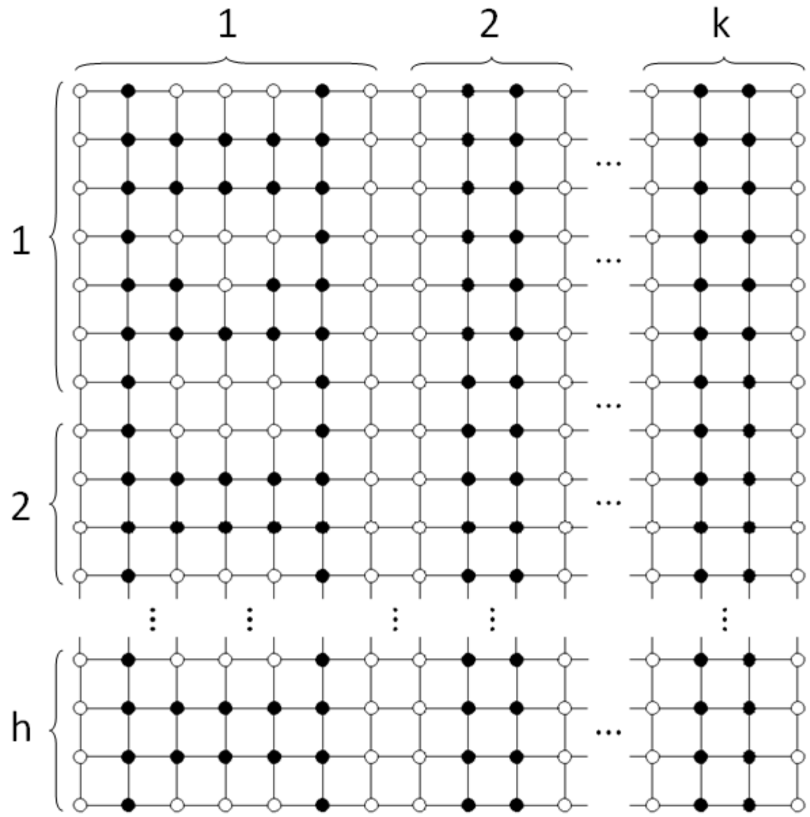

Figure 7.38: A global secure set of $P_{4h+3} \times P_{4k+3}$ as constructed by the proof of Lemma 7.3.31.

Corollary 7.3.33. $\gamma_s(P_n \times P_{4k+3}) = \lceil n(4k+3)/2 \rceil$, for $n \geq 2, k \geq 0$.

Proof.

1. $k = 0$. By Corollary 7.3.16 and Observation 7.3.4.

2. $k \geq 1$.

   (a) $n = 2$. By Lemma 7.3.5.

   (b) $n = 3$. By Corollary 7.3.16.

(c) $n \in \{4h : h \geq 1\}$. By Lemma 7.3.7 and Observation 7.3.4.

(d) $n \in \{4h+1 : h \geq 1\}$. By Corollary 7.3.23 and Observation 7.3.4.

(e) $n \in \{4h+2 : h \geq 1\}$. By Corollary 7.3.30 and Observation 7.3.4.

(f) $n \in \{4h+3 : h \geq 1\}$. By Lemma 7.3.31.

□

Theorem 7.3.34. $\gamma_s(P_n \times P_m) = \lceil nm/2 \rceil$, for $n, m \geq 2$.

Proof. All cases based on the value of m are covered, as shown in the following list.

1. $m \in \{4k : k \geq 1\}$. By Lemma 7.3.7.

2. $m \in \{4k+1 : k \geq 1\}$. By Corollary 7.3.23.

3. $m \in \{4k+2 : k \geq 0\}$. By Corollary 7.3.30.

4. $m \in \{4k+3 : k \geq 0\}$. By Corollary 7.3.33.

□

7.4 Two-dimensional cylinders

In this section, we investigate the values of $\gamma_s(P_n \times C_m)$. For many of the dimensions, global secure sets of cardinality $\lceil nm/2 \rceil$ for $P_n \times C_m$ were obtained in Section 7.3 by applying

Lemma 7.3.3 on the global secure set configurations of $P_n \times P_m$. In this section, we collect results and treat the missing cases.

Lemma 7.4.1. $\gamma_s(C_3 \times P_{4k+3}) = \lceil 3(4k+3)/2 \rceil$, for $k \geq 0$.

Proof. For $k = 0$, a global secure set of cardinality $5 = \lceil 3 \cdot (4 \times 0 + 3)/2 \rceil$ for $C_3 \times P_3$ is shown in Figure 7.39. For $k = 1$, a global secure set of cardinality $11 = (\lceil 3 \cdot (4 \times 1 + 3)/2 \rceil$ for $C_3 \times P_7$ is shown in Figure 7.40. For $k > 1$, consider the disjoint union of $C_3 \times P_{4(k-1)+3}$ and $C_3 \times P_4$ shown in Figure 7.41. The rightmost column of $C_3 \times P_{4(k-1)+3}$ is identical to the leftmost column of $C_3 \times P_4$. Adding edges between corresponding vertices in these two columns produces $C_3 \times P_{4k+3}$, with a global secure set of cardinality $\lceil 3(4(k-1)+3)/2 \rceil + 6 = 6 \cdot (k-1) + 5 + 6 = 6k + 5 = \lceil 3(4k+3)/2 \rceil$. $\square$

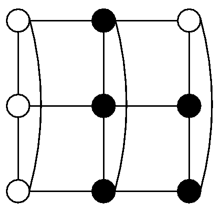

Figure 7.39: A global secure set of cardinality 5 for $C_3 \times P_3$.

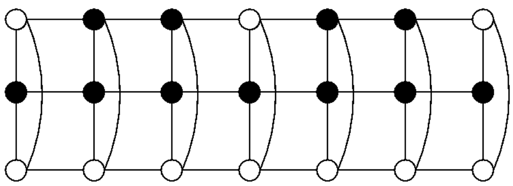

Figure 7.40: A global secure set of cardinality 11 for $C_3 \times P_7$. (See Remark 7.3.1.)

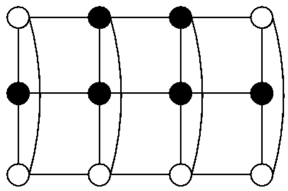

Figure 7.41: A global secure set of cardinality 6 for $C_3 \times P_4$.

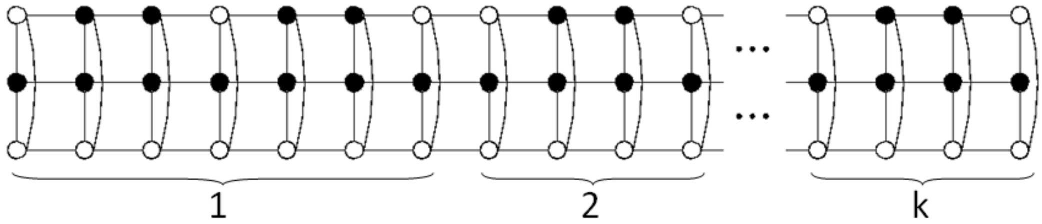

Figure 7.42: A global secure set of $C_3 \times P_{4k+3}$ as constructed by the proof of Lemma 7.4.1.

Figure 7.42 illustrates the global secure set pattern for $C_3 \times P_{4k+3}, k \geq 1$, specified in the proof of Lemma 7.4.1. The leftmost and rightmost columns of these sets are identical. By Lemma 7.3.3,

Corollary 7.4.2. $\gamma_s(C_3 \times C_{4k+3}) = \lceil 3(4k+3)/2 \rceil$ for $k \geq 1$.

Corollary 7.4.3. $\gamma_s(P_3 \times C_m) = \lceil 3m/2 \rceil$, for $m \geq 3$

Proof. All cases based on the value of m are covered, as shown in the following list.

1. $m = 3$. By Figure 7.39 and Observation 7.3.4.

2. $m \in \{4k : k \geq 1\}$. By Corollary 7.3.8.

3. $m \in \{4k+1 : k \geq 1\}$. By Corollary 7.3.11.

4. $m \in \{4k+2 : k \geq 1\}$. By Corollary 7.3.13.

5. $m \in \{4k+3 : k \geq 1\}$. By Corollary 7.3.15.

$\square$

Corollary 7.4.4. $\gamma_s(P_n \times C_{4k+1}) = \lceil n(4k+1)/2 \rceil$, for $n \geq 2, k \geq 1$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n = 2$. By Corollary 7.3.6.
2. $n = 3$. By Corollary 7.4.3.
3. $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.
4. $n \in \{4h+1 : h \geq 1\}$. By Corollary 7.3.18.
5. $n \in \{4h+2 : h \geq 1\}$. By Corollary 7.3.20.
6. $n \in \{4h+3 : h \geq 1\}$. By Corollary 7.3.22.

$\square$

Lemma 7.4.5. $\gamma_s(P_7 \times C_{4k+2}) = 7(2k+1)$, for $k \geq 1$.

Proof. Consider the disjoint union of $P_3 \times P_{4k+2}$ (with a global secure set constructed by the proof of Lemma 7.3.12) and $P_4 \times P_{4k+2}$ (with a global secure set consisting of the middle two rows of P_{4k+2}). The bottom row of $P_3 \times P_{4k+2}$ and the top row of $P_4 \times P_{4k+2}$ are identical. Adding edges between corresponding vertices on these two rows produces $P_7 \times P_{4k+2}$, with a global secure set of cardinality $6k+3+2 \times (4k+2) = 14k+7 = 7(2k+1)$. The leftmost and rightmost columns of $P_7 \times P_{4k+2}$ are identical, adding edges between corresponding vertices in these two columns produces $P_7 \times C_{4k+2}$, with the same global secure set. $\square$

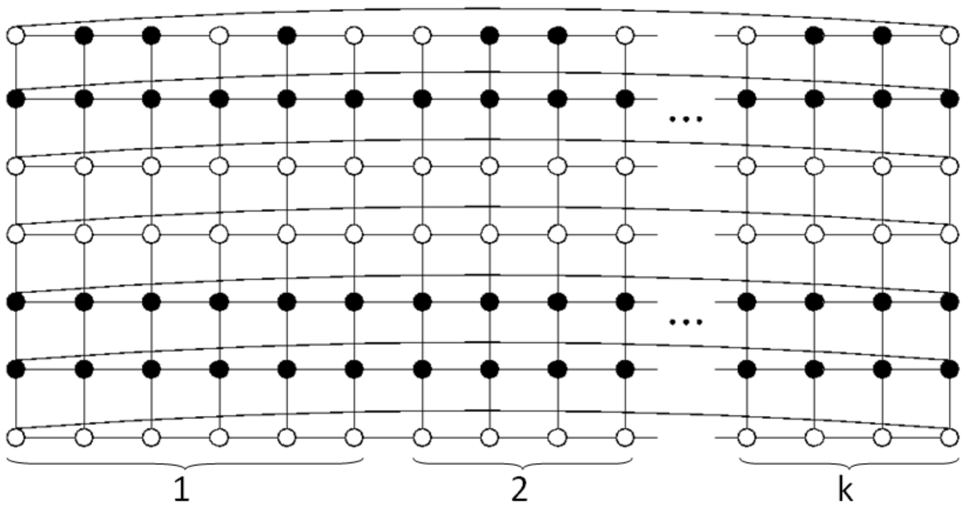

Figure 7.43: A global secure set of $P_7 \times C_{4k+2}$ as constructed by the proof of Lemma 7.4.5.

Figure 7.43 illustrates the global secure set pattern of $P_7 \times C_{4k+2}$ specified in the proof of Lemma 7.4.5.

Corollary 7.4.6. $\gamma_s(P_n \times C_{4k+2}) = n(2k+1)$, for $n \geq 2, k \geq 1$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n = 2$. By Corollary 7.3.6.

2. $n = 3$. By Corollary 7.4.3.

3. $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.

4. $n \in \{4h + 1 : h \geq 1\}$. By Corollary 7.3.20 and Observation 7.3.4.

5. $n \in \{4h + 2 : h \geq 1\}$. By Corollary 7.3.25.

6. $n \in \{4h + 3 : h \geq 1\}$.

   (a) $n = 7$. By Lemma 7.4.5.

   (b) $n \in \{4h + 3 : h \geq 2\}$. By Corollary 7.3.27 and Observation 7.3.4.

□

Lemma 7.4.7. $\gamma_s(C_3 \times P_{4k+1}) = \lceil 3(4k + 1)/2 \rceil$, for $k \geq 1$.

Proof. For $k = 1$, a global secure set for $C_3 \times P_5$ of cardinality $8 = \lceil (3 \times 5)/2 \rceil$ is given in Figure 7.44. For $k > 1$, consider the disjoint union of $C_3 \times P_{4(k-1)+1}$ and $C_3 \times P_4$ in Figure 7.41. The rightmost column of $C_3 \times P_{4(k-1)+1}$ is identical to the leftmost column of $C_3 \times P_4$. Adding edges between corresponding vertices in these two columns produces $C_3 \times P_{4k+1}$, with a global secure set of cardinality $\lceil 3 \times (4(k-1)+1)/2 \rceil + 6 = 6(k-1) + 2 + 6 = 6k + 2 = \lceil 3(4k + 1)/2 \rceil$. □

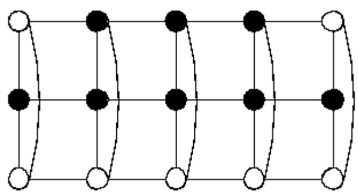

Figure 7.44: A global secure set of $C_3 \times P_5$.

Figure 7.45 illustrates the global secure set pattern of $C_3 \times P_{4k+1}$ specified in the proof of Lemma 7.4.7.

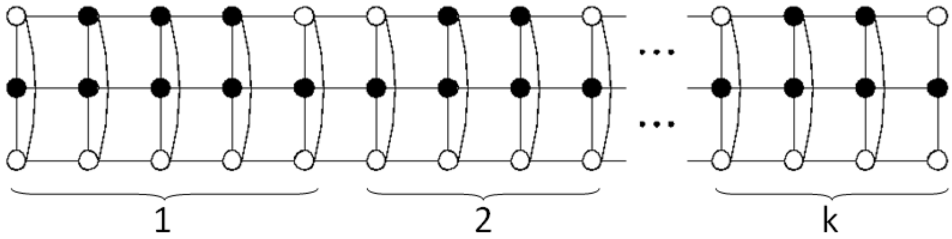

Figure 7.45: A global secure set of $C_3 \times P_{4k+1}$ as constructed by the proof of Lemma 7.4.7.

Corollary 7.4.8. $\gamma_s(C_3 \times C_{4k+1}) = \lceil 3(4k+1)/2 \rceil$, for $k \geq 1$.

Lemma 7.4.9. $\gamma_s(C_3 \times P_{4k+2}) = 3(2k+1)$, for $k \geq 0$.

Proof. For $k = 0$, a global secure set for $C_3 \times P_2$ of cardinality $3 = 3 \times (2 \cdot 0 + 1)$ is given in Figure 7.46. For $k > 0$, consider the disjoint union of $C_3 \times P_{4(k-1)+2}$ and $C_3 \times P_4$ in Figure 7.41. The rightmost column of $C_3 \times P_{4(k-1)+2}$ is identical to the leftmost column of $C_3 \times P_4$. Adding edges between corresponding vertices in these two columns produces $C_3 \times P_{4k+2}$, with a global secure set of cardinality $3 \times (2(k-1)+1) + 6 = 3(2k+1)$. $\square$

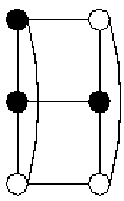

Figure 7.46: A global secure set of $C_3 \times P_2$.

Figure 7.47 illustrates the global secure set pattern of $C_3 \times P_{4k+2}$ specified in the proof of Lemma 7.4.9. Notice in this pattern the leftmost and rightmost columns are not identical, and so the configurations cannot be extended to global secure set configurations of $C_3 \times C_{4k+2}$. In Section 7.5, we provide global secure set configurations for $C_3 \times C_{4k+2}$ with $6k+4$

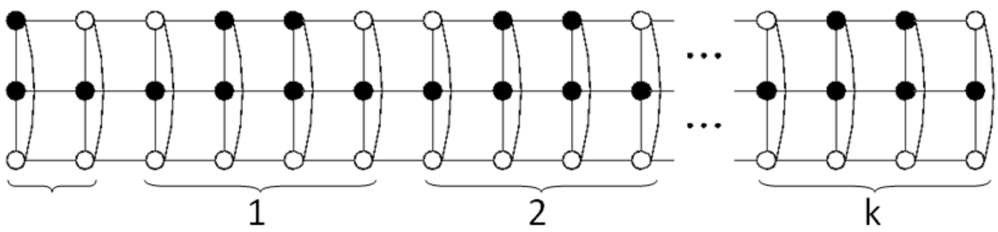

Figure 7.47: A global secure set of $C_3 \times P_{4k+2}$ as constructed by the proof of Lemma 7.4.9.

vertices, showing $\gamma_s(C_3 \times C_{4k+2}) \leq 6k + 4$. Then, Chapter 8 will show that global secure set configurations of $C_3 \times C_{4k+2}$ with cardinality $6k + 3$ cannot exist, proving $\gamma_s(C_3 \times C_{4k+2}) = 6k + 4$.

Corollary 7.4.10. $\gamma_s(P_n \times C_3) = \lceil 3n/2 \rceil$, for $n \geq 2$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.

2. $n \in \{4h + 1 : h \geq 1\}$. By Lemma 7.4.7 and Observation 7.3.4.

3. $n \in \{4h + 2 : h \geq 0\}$. By Lemma 7.4.9 and Observation 7.3.4.

4. $n \in \{4h + 3 : h \geq 0\}$. By Lemma 7.4.1 and Observation 7.3.4.

□

Corollary 7.4.11. $\gamma_s(P_n \times C_{4k+3}) = \lceil n(4k + 3)/2 \rceil$, for $n \geq 2, k \geq 0$.

Proof.

1. $k = 0$. By Corollary 7.4.10.

2. $k \geq 1$.

   (a) $n = 2$. By Corollary 7.3.6.

   (b) $n = 3$. By Corollary 7.4.3.

   (c) $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.

   (d) $n \in \{4h + 1 : h \geq 1\}$. By Corollary 7.3.22 and Observation 7.3.4.

   (e) $n \in \{4h + 2 : h \geq 1\}$.

      i. $k = 1$. By Corollary 7.3.29.

      ii. $k \geq 2$. By Corollary 7.3.27.

   (f) $n \in \{4h + 3 : h \geq 1\}$. By Corollary 7.3.32.

□

Theorem 7.4.12. $\gamma_s(P_n \times C_m) = \lceil nm/2 \rceil$, for $n \geq 2, m \geq 3$.

Proof. All cases based on the value of m are covered, as shown in the following list.

1. $m \in \{4k : k \geq 1\}$. By Corollary 7.3.8.

2. $m \in \{4k + 1 : k \geq 1\}$. By Corollary 7.4.4.

3. $m \in \{4k + 2 : k \geq 1\}$. By Corollary 7.4.6.

4. $m \in \{4k + 3 : k \geq 0\}$. By Corollary 7.4.11.

□

7.5 Two-dimensional tori

The global security numbers of $C_n \times C_m$ are more interesting. It is not the case that $\gamma_s(C_n \times C_m) = \lceil nm/2 \rceil$ for all $n, m \geq 3$, with $C_3 \times C_{4k+2}$ being the only exception in the class of two-dimensional tori. In this section, Lemma 7.5.1 constructs global secure sets for $C_3 \times C_{4k+2}$ with cardinality $6k + 4$, which is one larger than the lower bound given in Lemma 6.2.1. The result is $6k + 3 \leq \gamma_s(C_3 \times C_{4k+2}) \leq 6k + 4$. In Chapter 8, we show that $\gamma_s(C_3 \times C_{4k+2}) > 6k + 3$ using specialized analysis. Then, Lemma 7.5.2 constructs global secure sets for $C_7 \times C_{4k+2}$ with cardinality $14k + 7$. Among the class of two-dimensional tori, these are the only remaining cases whose global secure set configurations were not obtained as extensions of $P_n \times P_m$ or $P_n \times C_m$ configurations.

Lemma 7.5.1. $\gamma_s(C_3 \times C_{4k+2}) \leq 6k + 4$, for $k \geq 1$.

Proof. We first construct a global secure set of cardinality $6k + 4$ for $C_3 \times P_{4k+2}$, with the property that the leftmost and rightmost columns are identical. For $k = 1$, a global secure set of cardinality $10 = 6 \times 1 + 4$ for $C_3 \times P_6$ is given in Figure 7.48. For $k > 1$, consider the disjoint union of $C_3 \times P_{4(k-1)+2}$ and $C_3 \times P_4$ in Figure 7.41. The rightmost column of $C_3 \times P_{4(k-1)+2}$ is identical to the leftmost column of $C_3 \times P_4$. Adding edges between corresponding vertices of these two columns produces $C_3 \times P_{4k+2}$, with a global secure set of cardinality $6(k - 1) + 4 + 6 = 6k + 4$. Note that the leftmost and rightmost columns of $C_3 \times P_{4k+2}$ are identical. Adding edges between corresponding vertices of these two columns produces $C_3 \times C_{4k+2}$, with the same global secure set. □

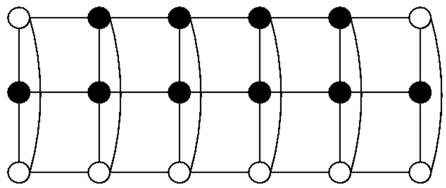

Figure 7.48: A global secure set of $C_3 \times P_6$.

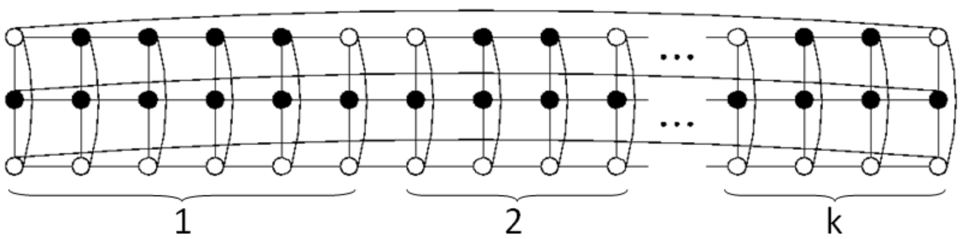

Figure 7.49: A global secure set of $C_3 \times C_{4k+2}$ as constructed by the proof of Lemma 7.5.1.

Figure 7.49 illustrates the global secure set pattern of $C_3 \times C_{4k+2}$ specified in the proof of Lemma 7.5.1.

Lemma 7.5.2. $\gamma_s(C_7 \times C_{4k+2}) = 14k + 7$ for $k \geq 1$.

Proof. By Lemma 6.2.1, $\gamma_s(C_7 \times C_{4k+2}) \geq 14k + 7$. Similar to the proof of Lemma 7.5.1, we construct global secure sets of cardinality $14k + 7$ for $C_7 \times P_{4k+2}$, with the property that the leftmost and rightmost columns of the configurations are identical.

Figures 7.50 and 7.51 show global secure set configurations for $C_7 \times P_6$ and $C_7 \times P_4$, respectively. The validity of these configurations has been checked by a computer program (see Remark 7.3.1). For $k = 1$, Figure 7.50 shows a global secure set of cardinality $21 = (14 \times 1) + 7$ for $C_7 \times P_6$. For $k > 1$, consider the disjoint union of $C_7 \times P_{4(k-1)+2}$ and $C_7 \times P_4$ shown in Figure 7.51. The rightmost column of $C_7 \times P_{4(k-1)+2}$ is identical to the leftmost column of $C_7 \times P_4$. By Observation 7.1.3 (Part 2), adding edges between corresponding vertices in these two columns produces $C_7 \times P_{4k+2}$, with a global secure set of cardinality $14(k-1)+7+14 = 14k+7$. Note that the leftmost and rightmost columns of $C_7 \times P_{4k+2}$ are also identical. Adding edges between corresponding vertices of these two columns produces $C_7 \times C_{4k+2}$, with the same global secure set. $\square$

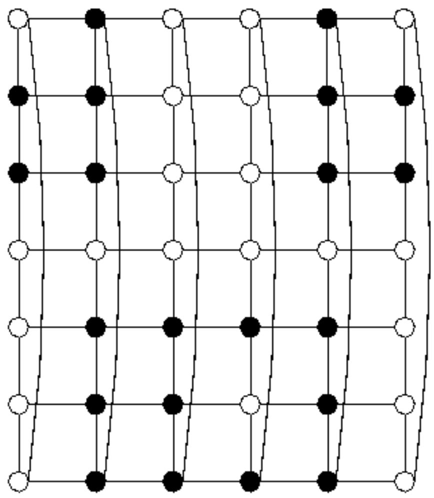

Figure 7.50: A global secure set of $C_7 \times P_6$. (See Remark 7.3.1.)

Figure 7.52 illustrates the global secure set pattern of $C_7 \times C_{4k+2}$ specified in the proof of Lemma 7.5.2.

In the remark following Corollary 7.3.29, we noted that there are no global secure set configurations for $P_7 \times P_6$ (or $P_6 \times P_7$) of cardinality exactly 21, with the properties that the leftmost and rightmost columns contain no vertex in the set, and the top and bottom rows are identical. A computer program verified that there is no global secure set configuration for $P_7 \times P_6$ (or $P_6 \times P_7$) of cardinality exactly 21, with the properties that the leftmost and

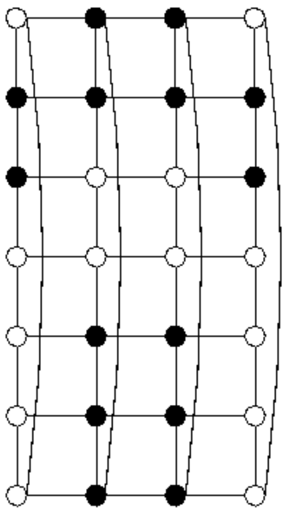

Figure 7.51: A global secure set of $C_7 \times P_4$ with 14 vertices. (See Remark 7.3.1.)

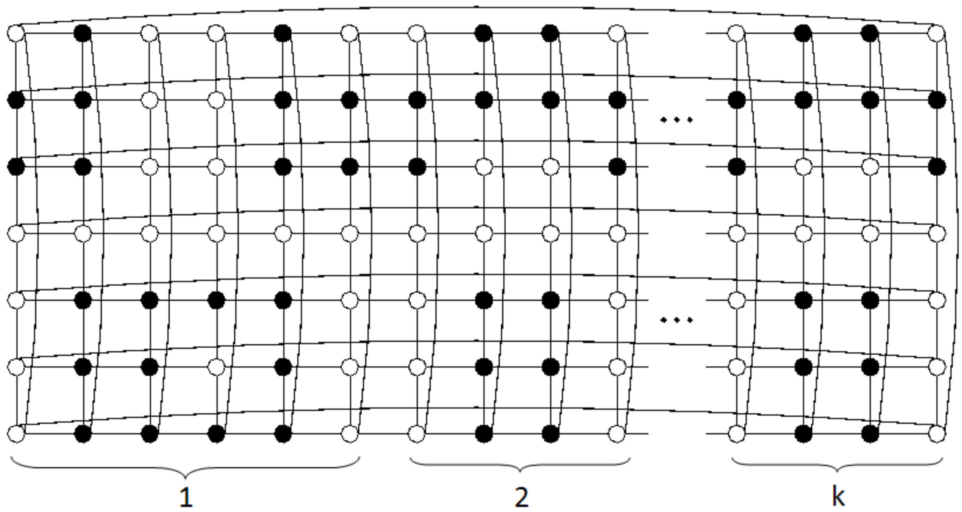

Figure 7.52: A global secure set of $C_7 \times C_{4k+2}$ as constructed by the proof of Lemma 7.5.2.

rightmost columns are identical, and the top and bottom rows are identical. This means a global secure set for $C_7 \times C_{4k+2}$ cannot be built using $P_7 \times P_6$ or $P_6 \times P_7$ configurations. Luckily, a global secure set for $C_7 \times P_6$ exists where the leftmost and rightmost columns are identical (Figure 7.50), and another configuration exists for $C_7 \times P_4$ where its leftmost and rightmost columns are identical to the leftmost and rightmost columns of $C_7 \times P_6$. These two configurations can then be used to construct global secure set configurations of $C_7 \times C_{4k+2}$, as seen in the proof of Lemma 7.5.2.

The remainder of this section collects results for $\gamma_s(C_n \times C_m)$, with the final result given in Theorem 7.5.7.

Corollary 7.5.3. $\gamma_s(C_3 \times C_m) = \lceil 3m/2 \rceil$, for $m \geq 3$ and $m \notin \{4k+2 : k \geq 1\}$

Proof. All cases based on the value of m are covered, as shown in the following list.

1. $m \in \{4k : k \geq 1\}$. By Corollary 7.3.8.
2. $m \in \{4k+1 : k \geq 1\}$. By Corollary 7.4.8.
3. $m \in \{4k+3 : k \geq 0\}$.
    (a) $k = 0$. By Figure 7.53.
    (b) $k \geq 1$. By Corollary 7.4.2.

□

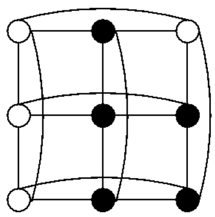

Figure 7.53: A global secure set for $C_3 \times C_3$.

Corollary 7.5.4. $\gamma_s(C_n \times C_{4k+1}) = \lceil n(4k+1)/2 \rceil$, for $n \geq 3, k \geq 1$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n = 3$. By Corollary 7.5.3.
2. $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.
3. $n \in \{4h + 1 : h \geq 1\}$. By Corollary 7.3.18.
4. $n \in \{4h + 2 : h \geq 1\}$. By Corollary 7.3.20.
5. $n \in \{4h + 3 : h \geq 1\}$. By Corollary 7.3.22.

□

Corollary 7.5.5. $\gamma_s(C_n \times C_{4k+2}) = n(2k+1)$, for $n \geq 4, k \geq 1$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.
2. $n \in \{4h + 1 : h \geq 1\}$. By Corollary 7.5.4 and Observation 7.3.4.
3. $n \in \{4h + 2 : h \geq 1\}$. By Corollary 7.3.25.
4. $n \in \{4h + 3 : h \geq 1\}$.
   (a) $n = 7$. By Lemma 7.5.2.
   (b) $n \in \{4h + 3 : h \geq 2\}$. By Corollary 7.3.27 and Observation 7.3.4.

$\square$

Corollary 7.5.6. $\gamma_s(C_n \times C_{4k+3}) = \lceil n(4k+3)/2 \rceil$, for $n \geq 3, k \geq 0$, except for the case when $n \in \{4h+2 : h \geq 1\}$ and $k = 0$.

Proof. All cases based on the value of n are covered, as shown in the following list.

1. $n = 3$. By Corollary 7.5.3.

2. $n \in \{4h : h \geq 1\}$. By Corollary 7.3.8 and Observation 7.3.4.

3. $n \in \{4h+1 : h \geq 1\}$. By Corollary 7.5.4 and Observation 7.3.4.

4. $n \in \{4h+2 : h \geq 1\}$. Then, we may let $k \geq 1$. The result follows by Corollary 7.5.5 and Observation 7.3.4.

5. $n \in \{4h+3 : h \geq 1\}$. By Corollary 7.3.32.

$\square$

Theorem 7.5.7.

1. $6k + 3 \leq \gamma_s(C_3 \times C_{4k+2}) \leq 6k + 4$, for $k \geq 1$.

2. $\gamma_s(C_n \times C_m) = \lceil nm/2 \rceil$, with the exception of $C_3 \times C_{4k+2}$.

Proof.

1. Lemma 6.2.1 establishes the lower bound and Lemma 7.5.1 establishes the upper bound.

2. All cases based on the value of m are covered, as shown in the following list.

   (a) $m \in \{4k : k \geq 1\}$. By Corollary 7.3.8.

   (b) $m \in \{4k+1 : k \geq 1\}$. By Corollary 7.5.4.

   (c) $m \in \{4k+2 : k \geq 1\}$. Then, we may let $n > 3$. The result follows from Corollary 7.5.5.

   (d) $m \in \{4k+3 : k \geq 0\}$. By Corollary 7.5.6.

□

This concludes the construction of global secure sets of grid-like graphs (P_n, C_n, $P_n \times P_m$, $P_n \times C_m$ and $C_n \times C_m$). Let G be a grid-like graph of order N. Among Theorems 7.2.1, 7.2.4, 7.3.34, 7.4.12 and 7.5.7, we showed that $\gamma_s(G) = \lceil N/2 \rceil$, unless G is isomorphic to C_{4k+2} or $C_3 \times C_{4k+2}$, in which case $\gamma_s(G) \leq N/2 + 1$ (Lemmas 7.2.3 and 7.5.1). In the next chapter, we analysis the lower bounds of $\gamma_s(C_{4k+2})$ and $\gamma_s(C_3 \times C_{4k+2})$, proving $\gamma_s(C_{4k+2}) = 2k+2$ and $\gamma_s(C_3 \times C_{4k+2}) = 6k+4$.

CHAPTER 8

GLOBAL SECURE SETS OF C_{4k+2} and $C_3 \times C_{4k+2}$

This chapter considers the global security numbers of C_{4k+2} and $C_3 \times C_{4k+2}$, the only two exceptions where $\gamma_s(G) > \lceil n/2 \rceil$ for a grid-like graph G of order n. In Chapter 7, we showed that $\gamma_s(C_{4k+2}) \leq 2k+2$ (Lemma 7.2.3) and $\gamma_s(C_3 \times C_{4k+2}) \leq 6k+4$ (Lemma 7.5.1). In this chapter, we develop lower bounds for $\gamma_s(C_{4k+2})$ and $\gamma_s(C_3 \times C_{4k+2})$ that match the given upper bounds, proving $\gamma_s(C_{4k+2}) = 2k+2$ and $\gamma_s(C_3 \times C_{4k+2}) = 6k+4$. Note that Lemma 6.2.1 does not provide a sharp lower bound for $\gamma_s(C_{4k+2})$ or $\gamma_s(C_3 \times C_{4k+2})$.

8.1 Global secure sets of C_{4k+2}

This section presents the lower bound for the global security number of C_{4k+2}, the simpler case of the two exceptions where $\gamma_s(G) > \lceil n/2 \rceil$ for a grid-like graph G of order n. Definition 8.1.1 below will be used in Theorem 8.1.2, proving $\gamma_s(C_{4k+2}) = 2k+2$.

Definition 8.1.1. Let C_n be a cycle of order n and let $S \subseteq V(C_n)$. An *attacker group* of C_n is a maximal consecutive sequence of vertices that are not in S. A *defender group* of C_n is a maximal consecutive sequence of vertices in S. Note that if the first and last vertices of

C_n are both in S, they belong to the same defender group. Similarly, two end vertices who are both outside S belong to the same attacker group. Figure 8.1 illustrates an example of attacker and defender groups for C_{14}, with each attacker group denoted by A_i and each defender group by D_i.

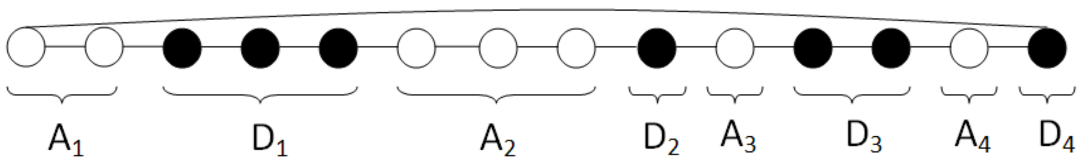

Figure 8.1: An example of attacker groups and defender groups in C_{14}.

Theorem 8.1.2. $\gamma_s(C_{4k+2}) = 2k + 2$ for $k \geq 1$.

Proof. By Lemma 6.2.1, $\gamma_s(C_{4k+2}) \geq 2k + 1$, and by Lemma 7.2.3, $\gamma_s(C_{4k+2}) \leq 2k + 2$. We show that $\gamma_s(C_{4k+2}) \neq 2k + 1$. By the way of contradiction, let S be a global secure set of C_{4k+2} with $|S| = 2k + 1$. Let V be the vertex set of C_{4k+2}. With reference to Definition 8.1.1, consider the attacker and defender groups of C_{4k+2} as given by the set S, with vertices oriented so an entire attacker group forms A_1. Since $0 < |S| < |V|$, there is at least one attacker group and at least one defender group, and they must alternate. So, the number of attacker groups must equal the number of defender groups.

Let $A_1, D_1, A_2, D_2, \ldots, A_t, D_t$ be the attacker and defender groups of C_{4k+2}. Let $|A_i|$ be the number of vertices in group A_i and $|D_i|$ be the number of vertices in group D_i, for $1 \leq i \leq t$. If $|A_i| > 2$, then some vertex of A_i is not dominated (such as the middle vertex of group A_2 in Figure 8.1), a contradiction to S being a dominating set. If $|D_i| < 2$, then

D_i consists of a single vertex and is unable to defend an attack coming from both of its neighbors (such as D_2 in Figure 8.1), a contradiction to S being a secure set. Thus, $|A_i| \leq 2$ and $|D_i| \geq 2$ for all $1 \leq i \leq t$. Notice $|S| = 2k+1 = |V - S|$. Then $2t \leq \sum_{i=1}^{t} |D_i| = |S| = |V - S| = \sum_{i=1}^{t} |A_i| \leq 2t$. So, $2t = |S| = 2k+1$, which is impossible. Thus, $\gamma_s(C_{4k+2}) = 2k+2$ for $k \geq 1$. $\square$

8.2 Global secure sets of $C_3 \times C_{4k+2}$

This section presents the lower bound for the global security number of $C_3 \times C_{4k+2}$. The analysis technique is similar to the proof of Theorem 8.1.2 in that we analyze attacker and defender groups of $C_3 \times C_{4k+2}$, with a modified definition for these groups.

Definition 8.2.1. Let S be a global secure set of $C_3 \times C_{4k+2}$. Consider $C_3 \times C_{4k+2}$ as an array with 3 rows and $4k+2$ columns. A column is an *attacker column* if it contains at most one vertex in S, otherwise it is a *defender column*. So, an attacker column contains more attackers than defenders, and a defender column contains more defenders than attackers. An *attacker group* of $C_3 \times C_{4k+2}$ is a maximal consecutive sequence of attacker columns. A *defender group* of $C_3 \times C_{4k+2}$ is a maximal consecutive sequence of defender columns. The groups are taken cyclically around the leftmost and rightmost columns. So, if the leftmost and rightmost columns are both attacker (defender) columns, the two columns belong to the same group.

Notice if $|S| = 6k + 3$, then there is at least one attacker group and at least one defender group, and they must alternate, in which case *the number of attacker groups is equal to the number of defender groups.*

Recall from Corollary 3.3.1 that if S is not secure, then there exists a *witness* set $W \subseteq S$ such that $|N[W] \cap S| < |N[W] - S|$. Figure 8.2[1] enumerates the possible non-isomorphic attacker groups in $C_3 \times C_{4k+2}$. The enumeration process terminates when the configuration is either not secure (with a witness boxed), or not dominating (with an undominated vertex boxed). The enumeration is exhaustive, with rotations and reflections of the same configuration omitted. As seen from the enumeration, the only possible attacker groups of $C_3 \times C_{4k+2}$ are those shown in Figure 8.3, along with isomorphic (under vertical rotation) configurations. So, *any attacker group of $C_3 \times C_{4k+2}$ consists of either one or two columns.*

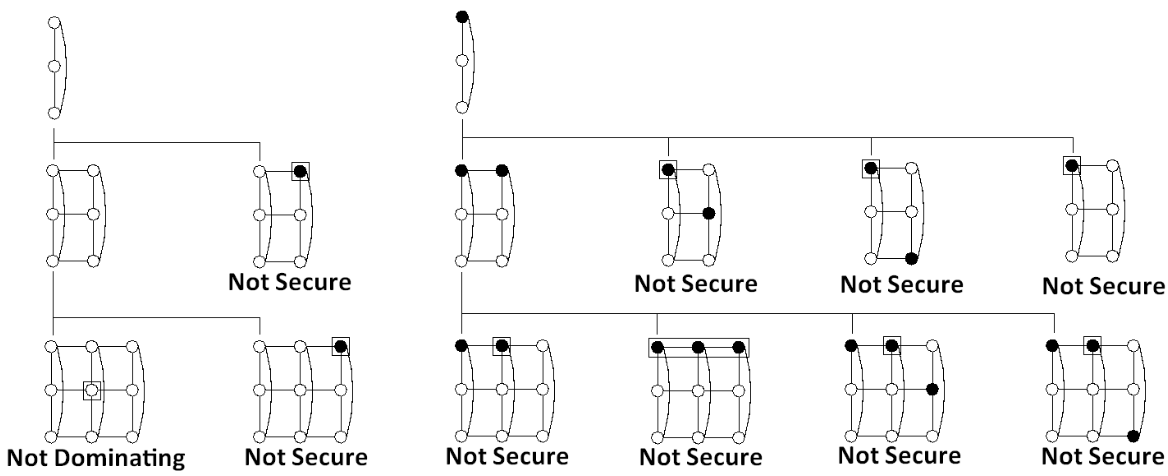

Figure 8.2: Enumeration of attacker groups

[1]In Figure 8.2 and subsequent figures (where applicable), each configuration corresponds to a partial projection of a global secure set of the entire graph, where the vertices in the set are marked in black.

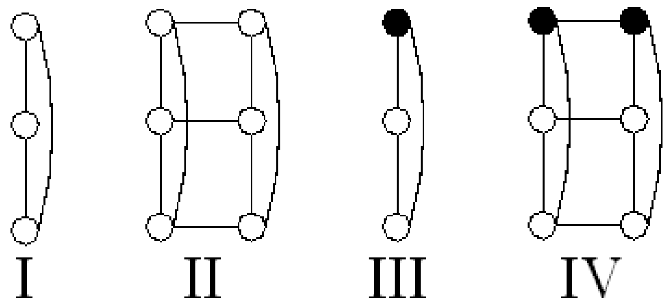

Figure 8.3: All non-isomorphic attacker groups in $C_3 \times C_{4k+2}$

We will refer to attacker groups in Figure 8.3 as attacker groups type I through IV, without explicit reference to Figure 8.3 repetitively. A column is *empty* if it contains no vertex in S. An attacker group may contain one or two consecutive empty columns, corresponding to attacker group types I and II respectively. We now proceed by case analysis based on the number of attacker groups of type II in $C_3 \times C_{4k+2}$. There are three cases. Lemma 8.2.5 considers global secure sets of $C_3 \times C_{4k+2}$ without any attacker groups of type II. Lemma 8.2.7 considers global secure sets of $C_3 \times C_{4k+2}$ with exactly one attacker group of type II. Finally, Theorem 8.2.8 considers the last case where a global secure set may contain at least two attackers groups of type II, and proving $\gamma_s(C_3 \times C_{4k+2}) = 6k + 4$.

First, Lemma 8.2.2 considers the possible configurations of an attacker group with an adjacent defender group, when $C_3 \times C_{4k+2}$ contains no attacker groups of type II. Then, Lemma 8.2.3 and Corollary 8.2.4 will develop a useful result to be used in the proof of Lemma 8.2.5, to show that $|S| > 6k + 3$ when S contains no attacker groups of type II.

Lemma 8.2.2. Suppose S is a global secure set of cardinality $6k+3$ for $C_3 \times C_{4k+2}$, such that $C_3 \times C_{4k+2}$ does not contain attacker groups of type II. If A is an attacker group and D is a defender group immediately adjacent to A (either to its left or right), then the only possible configurations of $A \cup D$, with isomorphic (under reflection and rotation) images omitted, are those shown in Figures 8.5 and 8.6.

Proof. Only three different types of attacker groups exist in $C_3 \times C_{4k+2}$ as seen from Figure 8.3. With reference to Figure 8.4, we start with each attacker group and enumerate, to its *right*, the possible configurations of S in $C_3 \times C_{4k+2}$. The column immediately following each group must be a defender column, because an attacker group is maximal. The column after this defender column may be of any kind. Let a be the number of attackers and d be the number of defenders in each configuration. The enumeration terminates either when the configuration is not secure (with witnesses boxed), or when $a \le d$. We have established that $a \le d$ in each $A \cup D$.

Let $A_1, D_1, \ldots, A_t, D_t$ be the attacker and defender groups of $C_3 \times C_{4k+2}$. Let a_i and d_i be the number of attackers and defenders in $A_i \cup D_i$, respectively. The above paragraph establishes $a_i \le d_i$. But $\sum_{i=1}^{t} a_i = \sum_{i=1}^{t} d_i = 6k+3$, so $a_i = d_i$ for $1 \le i \le t$.

Note that enumerations to the *left* of the three attacker groups are reflections of those shown in Figure 8.4. Consider $A_i \cup D_{i-1}$, with indexes wrapping around over $\{1, 2, \ldots, t\}$. By the enumerations, within each $A_i \cup D_{i-1}$ the number of attackers is at most the number of defenders. Since the total number of attackers in $C_3 \times C_{4k+2}$ equals the total number of defenders, the number of attackers within each $A_i \cup D_{i-1}$ must be equal to the number of defenders.

Since the number of attackers equals the number of defenders in each $A_i \cup D_i$ and $A_i \cup D_{i-1}$, the enumerations terminating with fewer attackers than defenders ($a < d$) in Figure 8.4 cannot appear under the given assumptions. As a result, the possible configurations of $A_i \cup D_i$ (or $A_i \cup D_{i-1}$), with isomorphic images omitted, are those in Figure 8.5 (or 8.6). Note that, although enumerations in Figure 8.4 might not have included all columns of a defender group (because it terminated whenever $a \leq d$), the configurations shown in Figures 8.5 and 8.6 do include all columns of each defender group, since any additional defender column will make the number of defenders in $A_i \cup D_i$ (or $A_i \cup D_{i-1}$) strictly greater than the number of attackers, and would be invalid. □

Lemma 8.2.3. Let G be a graph and let S be a global secure set of G such that $|V(G)| = 2|S|$. Then, for all $w \in S$, there exists $u \in V(G) - S$ such that $(N(u) \cap S) \subseteq (N[w] \cap S)$.

Proof. Suppose not. Let $w \in S$ be such that for all $u_i \in V(G) - S$, $(N(u_i) \cap S) \nsubseteq (N[w] \cap S)$, where $V(G) - S = \{u_1, u_2, \ldots, u_k\}$ is the set of attackers. So, every vertex $u_i \in V(G) - S$ has a neighbor $u_i' \in (N(u_i) \cap S) - (N[w] \cap S)$. In particular, $u_i' \in S$ is a neighbor of u_i, but not w (and $u_i' \neq w$). Consider $X = \{u_i' : 1 \leq i \leq k\}$. Since $w \notin (N[X] \cap S)$, $|N[X] \cap S| \leq |S| - 1 < |S| = |V(G) - S| = |N[X] - S|$, a contradiction to S being a secure set. □

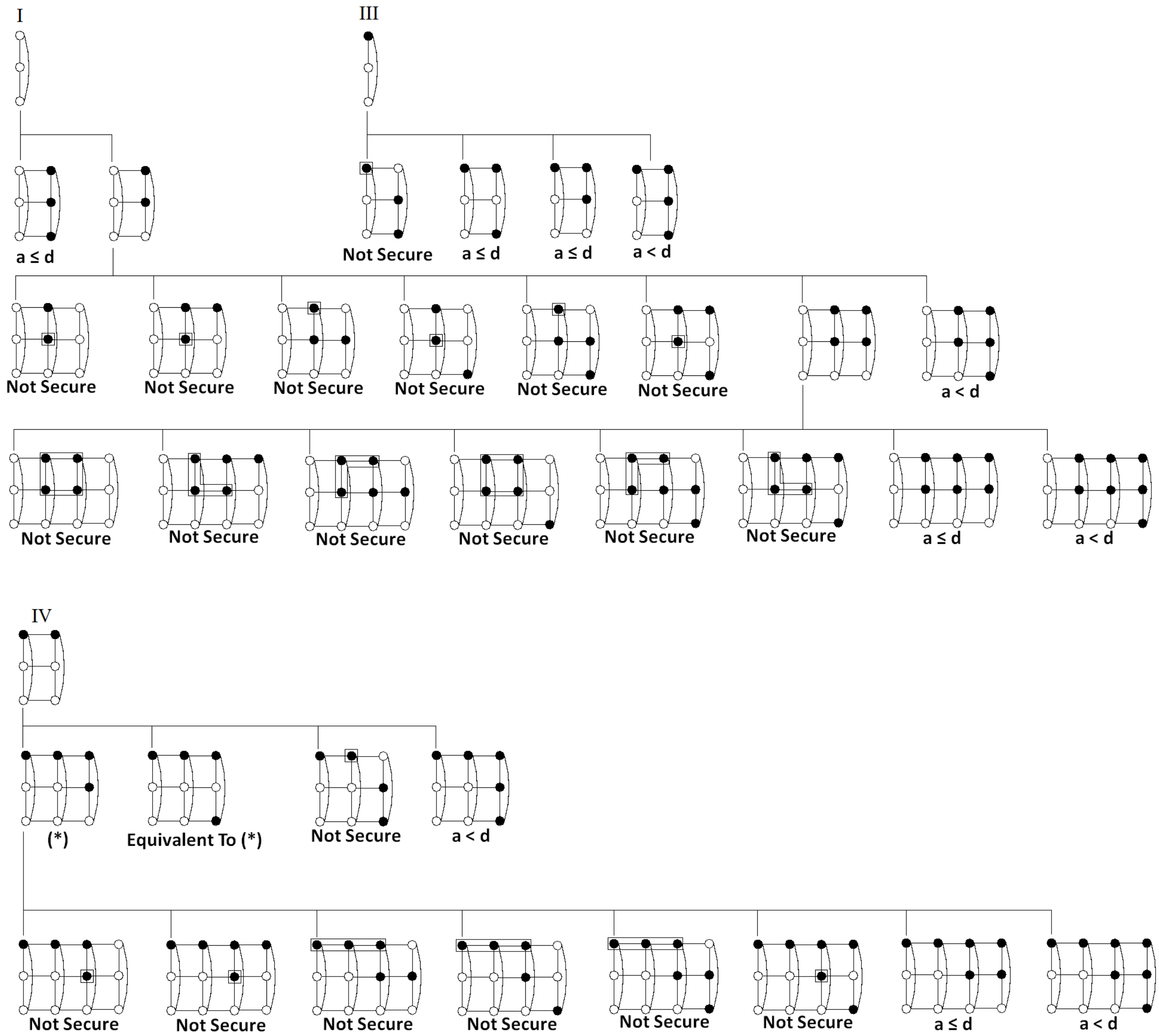

Figure 8.4: Enumerations of attacker group A followed by parts of defender group D. Symbols a and d denote the number of attackers and defenders in $A \cup D$, respectively. The attacker groups which start each enumeration are of types I, III and IV.

Corollary 8.2.4. Let G be a graph and let $S \subseteq V(G)$ be a dominating set of G such that $|V(G)| = 2|S|$. If there exists $w \in S$ such that for all $u \in V(G)-S$, $(N(u)\cap S)-(N[w]\cap S) \neq \emptyset$, then S is not a secure set.

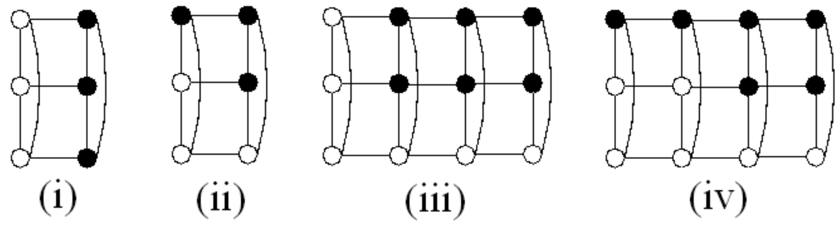

Figure 8.5: Possible attacker group with adjacent defender group to its right, assuming the global secure set does not contain attacker groups of type II. See the proof of Lemma 8.2.2 for details.

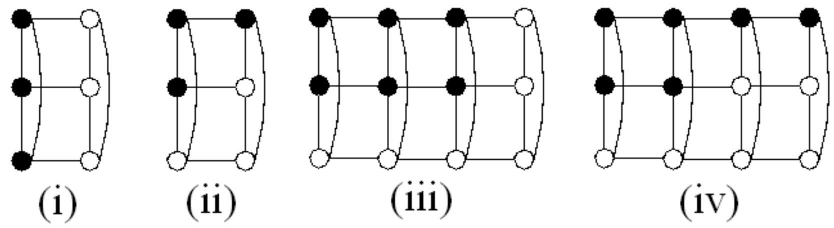

Figure 8.6: Possible attacker group with adjacent defender group to its left, assuming the global secure set does not contain attacker groups of type II. See the proof of Lemma 8.2.2 for details.

Lemma 8.2.5. Let S be a global secure set of $C_3 \times C_{4k+2}$ with no attacker groups of type II. Then, $|S| > 6k + 3$.

Proof. Assume $|S| = 6k + 3$. By Lemma 8.2.2, if A is an attacker group and D is a defender group adjacent to A, then Figures 8.5 and 8.6 present all non-isomorphic configurations of $A \cup D$. Recall that attacker and defender groups are maximal with respect to the number of consecutive attacker and defender columns they contain. Then, pattern (i) in Figures 8.5 and 8.6 cannot exist in S because the single column defender group would be next to an empty column and another attacker column, in which case the vertices in the defender group form a witness set.

The remaining three possible configurations have *different numbers of defenders* in their defender groups, and *different numbers of defenders* in their attacker groups. We claim that configuration S is composed of repeated patterns of exactly one type of $A \cup D$ groups. For example, if S contains adjacent attacker/defender groups that are instances of pattern (iii) of Figure 8.5 (or 8.6), then *all* adjacent attacker/defender groups of S are instances of pattern (iii) of Figure 8.5 (or 8.6), with possible vertical reflection and rotation applied to different instances. To justify this, let $A_1, D_1, \ldots, A_t, D_t$ be the attacker and defender groups of S. Without loss of generality, suppose $A_1 \cup D_1$ forms pattern (iii) in Figure 8.5. This implies D_1 *must contain exactly six defenders.* Then, in $D_1 \cup A_2$, the only valid choice from Figure 8.6 is pattern (iii) because it is the only pattern where the defender group contains exactly six defenders. In turn, this implies A_2 must consist of a single empty column. Now consider $A_2 \cup D_2$. Patterns (i) and (iii) in Figure 8.5 have attacker groups consisting of exactly one empty column, but we established in the previous paragraph that (i) cannot occur. Thus, the only valid pattern for $A_2 \cup D_2$ is (iii) of Figure 8.5. Similar arguments hold for the remaining groups.

Next, we claim that patterns (iii) and (iv) of Figure 8.5 (or 8.6) cannot occur in S. If there is a pattern of type (iii) (or (iv)), by the argument in the last paragraph the entire configuration must consist of only patterns of type (iii) (or (iv)). But, since pattern (iii) (or (iv)) has 4 columns, the total number of columns of $C_3 \times C_{4k+2}$ must be a multiple of 4, an impossibility.

Finally, we show that it is also impossible for S to be composed of only instances of pattern (ii) in Figure 8.5 (or 8.6). Figure 8.7 enumerates the possible configurations of S that consist of only instances of type (ii) patterns, with possible vertical reflection and rotation applied to different instances. In the enumeration, the columns must alternate between a column with exactly one vertex in S and a column with exactly two vertices in S. Note that, if a vertex $v \in S$ has two neighbors outside S, then the other two neighbors of v must be in S, for otherwise $\{v\}$ is a witness. Situations like this are noted by an arrow pointing from v to the neighbor of v that must be included in S. If a configuration is not secure, then either a witness is boxed, or a vertex $w \in S$ is labeled, where for all $u \in V(G) - S$, $(N(u) \cap S) - (N[w] \cap S) \neq \emptyset$ (Corollary 8.2.4). Figure 8.7 shows that there is no secure set configuration for $C_3 \times C_{4k+2}$, when the set consists of only instances of type (ii) pattern in Figures 8.5 and 8.6. $\square$

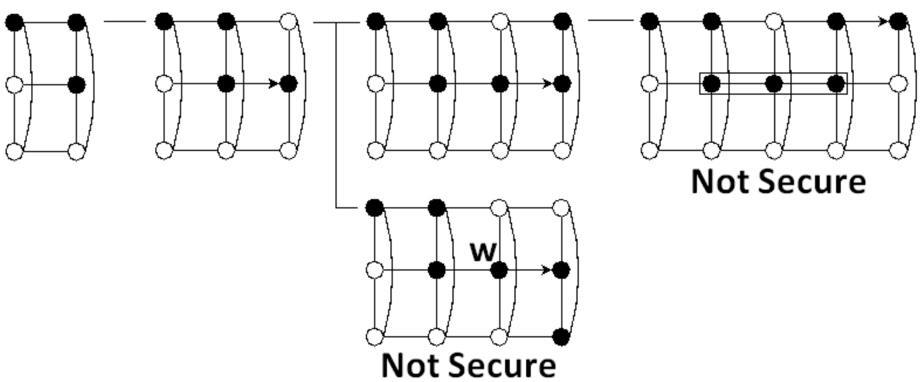

Figure 8.7: Enumeration of possible global secure set configurations in $C_3 \times C_{4k+2}$, assuming the set is composed of only instances of pattern (ii) in Figures 8.5 and 8.6. See the proof of Lemma 8.2.5 for details.

Next, Lemma 8.2.6 will be used in the proof of Lemma 8.2.7, to show that $|S| > 6k+3$ if S contains exactly one attacker group of type II.

Lemma 8.2.6. Assume S is a global secure set of cardinality $6k+3$ for $C_3 \times C_{4k+2}$. If exactly one attacker group in $C_3 \times C_{4k+2}$ is of type II, then there is no attacker group of type I.

Proof. Suppose not. Let $c_1, c_2, \ldots, c_{4k+2}$ be the columns of $C_3 \times C_{4k+2}$. Without loss of generality, label the two adjacent empty columns c_1 and c_{4k+2}. Then, c_1 and c_{4k+2} form an attacker group of type II. In order to dominate vertices of c_1 and c_{4k+2}, all vertices of c_2 and c_{4k+1} are in S. Let c_i be another empty column where $2 < i < 4k+1$. Column c_i forms an attacker group of type I. Since attacker groups are maximal, c_{i-1} and c_{i+1} are defender columns, so each contains at least two vertices in S.

Let a and d be the number of attackers and defenders in $C_3 \times C_{4k+2}$, respectively. Let $V_1 = c_2 \cup c_3 \cup \cdots \cup c_{i-1}$ and $V_2 = c_{i+1} \cup \cdots \cup c_{4k+1}$. Let a_j and d_j be the number of attackers and defenders in V_j, respectively, for $j \in \{1, 2\}$. With this notation, $a = d = 6k+3$, $a = a_1 + a_2 + 9$ and $d = d_1 + d_2$.

Let $X_1 = (V_1 \cap S)$ and $X_2 = (V_2 \cap S)$. Note that vertices in V_1 are not adjacent to vertices of V_2. So, $(N[X_1] \cap S) = X_1$ and X_1 dominates V_1. Among vertices in $N[X_1] - S$, there are a_1 attackers in V_1, three attackers in c_1 and at least two attackers in c_i (because c_{i-1} is a defender column). Thus, $|N[X_1] - S| \geq a_1 + 5$. Since S is a secure set, $|X_1| = |N[X_1] \cap S| \geq |N[X_1] -$

$S| \geq a_1 + 5$. Similarly, $|X_2| \geq a_2 + 5$. Then, $d = d_1 + d_2 = |X_1| + |X_2| \geq a_1 + a_2 + 10 = a + 1$, which is impossible since $d = a$. □

Lemma 8.2.7. Let S be a global secure set of $C_3 \times C_{4k+2}$. If exactly one attacker group of S is of type II, then $|S| > 6k + 3$.

Proof. Assume $|S| = 6k + 3$. By Lemma 8.2.6, there is no attacker group of type I in S. Let $c_1, c_2, \ldots, c_{4k+2}$ be the columns of $C_3 \times C_{4k+2}$. Without loss of generality, label the two adjacent empty columns c_1 and c_{4k+2}. Columns c_1 and c_{4k+2} form an attacker group of type II. In order to dominate vertices of c_1 and c_{4k+2}, all vertices in c_2 and c_{4k+1} must be in S.

Let $X = c_3 \cup c_4 \cup \cdots \cup c_{4k}$ and let a and d denote the number of attackers and defenders in X. Notice $a = d = 6k - 3$. Let d_i be the number of columns with i defenders, among the columns of X, for $0 \leq i \leq 3$. Since X contains no empty columns, $d_0 = 0$. With this notation, $d_0 + d_1 = d_1$ is the number of attacker columns among columns of X and $d_2 + d_3$ is the number of defender columns of X. Furthermore, $a = 3d_0 + 2d_1 + d_2 = 2d_1 + d_2$ and $d = d_1 + 2d_2 + 3d_3$. Since $a = d$,

$$d_1 = d_2 + 3 \cdot d_3 \tag{8.1}$$

Let $A_1, D_1, \ldots, A_t, D_t$ be attacker and defender groups of S. More specifically, let A_1 be the attacker group $c_1 \cup c_{4k+2}$, D_1 be the defender group which contains c_2, A_2 be the attacker group to the right (increase in column number) of D_1, D_2 the defender group to the right of

A_2, and so on. Note that all attacker groups, with the exception of A_1, must be of type III or IV. With this notation, D_t contains column c_{4k+1}.

We claim that $t \geq 2$. If $t = 1$, then A_1 is the only attacker group in S. But columns of A_1 are c_1 and c_{4k+2}, which are not included in X. Thus, X contains only defender columns, or $d_1 = 0$. But $d_1 = d_2 + 3d_3$, so $d_2 = d_3 = 0$, which is impossible since X contains at least two columns.

The columns in $A_2 \cup D_2 \cup A_3 \cup D_3 \cup \cdots \cup A_{t-1} \cup D_{t-1} \cup A_t$ form a subset of X. X does not contain any column of A_1, and may contain some, but not all, of the columns of D_1 and D_t. Let $|A_j|$ and $|D_j|$ denote respectively the number of columns of A_j and D_j, for $1 \leq j \leq t$. Let $p_{i,j}$ be the number of columns with exactly i defenders in $A_j \cup D_j$, for $0 \leq i \leq 3$ and $2 \leq j \leq t-1$. Note that $p_{0,j} = 0$, $|A_j| = p_{1,j}$ and

$$|D_j| = p_{2,j} + p_{3,j} \text{ for } 2 \leq j \leq t-1 \tag{8.2}$$

With reference to Figure 8.4, consider enumerations that start with attacker groups of type III or IV. The valid partial configurations that are also terminal configurations are shown in Figure 8.8. These are valid partial configurations of $A_j \cup D_j$ for $2 \leq j \leq t-1$. Each partial configuration shows the entire attacker group, but may show only part of the defender group. By examining each configuration, we may establish

$$p_{1,j} \leq p_{2,j} + 2p_{3,j} \text{ for } 2 \leq j \leq t-1 \tag{8.3}$$

Notice c_3 must be a defender column, for otherwise vertices in c_2 form a witness. Similarly, c_{4k} must be a defender column. Then, $c_3 \in D_1$ and $c_{4k} \in D_t$. There are $d_2 + d_3$ defender columns in X, specifically $\{c_3, c_{4k}\} \subseteq X$ and $D_2 \cup D_3 \cup \cdots \cup D_{t-1} \subseteq X$ are defender columns. Then,

$$d_2 + d_3 \geq 2 + \sum_{j=2}^{t-1} (p_{2,j} + p_{3,j}) \tag{8.4}$$

Note that strict inequality in (8.4) may be possible, since $d_2 + d_3$ may contain other defender columns of D_1 and D_t, which are not accounted for on the right hand side. Now, consider attacker columns of X.

$$\begin{aligned}
d_1 &= \sum_{j=2}^{t} |A_j| \\
&= |A_t| + \sum_{j=2}^{t-1} |A_j| && \text{(By } t \geq 2\text{)} \\
&\leq 2 + \sum_{j=2}^{t-1} |A_j| && \text{(By } |A_t| \leq 2\text{, Fig. 8.3)} \\
&= 2 + \sum_{j=2}^{t-1} p_{1,j} && \text{(By } d_0 = p_{0,j} = 0\text{)} \\
&\leq 2 + \sum_{j=2}^{t-1} (p_{2,j} + 2p_{3,j}) && \text{(By (8.3))} \\
&\leq 2 + d_3 + \sum_{j=2}^{t-1} (p_{2,j} + p_{3,j}) && \text{(By } d_3 \geq \textstyle\sum_{j=2}^{t-1} p_{3,j}\text{)} \\
&\leq d_2 + 2 \cdot d_3 && \text{(By (8.4))}
\end{aligned} \tag{8.5}$$

Then, by (8.1) and (8.5), $d_2 + 3d_3 = d_1 \leq d_2 + 2d_3$. So, $d_3 = 0$ and $d_1 = d_2$. Among the columns of X, there are exactly d_1 attacker columns, each containing exactly one defender, and exactly d_2 defender columns, each containing exactly two defenders. Since $d_3 = 0$, $d_1 = d_2 = d_2 + 2d_3$ and the first and last terms of (8.5) are equal. So, all intermediate expressions must be equal. Rewrite (8.5) as equalities and substitute $d_3 = p_{3,j} = 0$,

$$\begin{aligned} d_1 &= \sum_{j=2}^{t} |A_j| \\ &= |A_t| + \sum_{j=2}^{t-1} |A_j| \\ &= 2 + \sum_{j=2}^{t-1} |A_j| \\ &= 2 + \sum_{j=2}^{t-1} p_{1,j} \\ &= 2 + \sum_{j=2}^{t-1} p_{2,j} \\ &= d_2 \end{aligned} \tag{8.6}$$

Then, (8.7), (8.8) and (8.9) are consequences of (8.6).

$$|A_t| = 2 \tag{8.7}$$

$$\sum_{j=2}^{t-1} p_{1,j} = \sum_{j=2}^{t-1} p_{2,j} \tag{8.8}$$

$$
\begin{aligned}
d_2 &= 2 + \sum_{j=2}^{t-1} p_{2,j} \quad \text{(By (8.6))} \\
&= 2 + \sum_{j=2}^{t-1} |D_j| \quad \text{(By (8.2) and } p_{3,j} = 0\text{)}
\end{aligned}
\tag{8.9}
$$

The d_2 defender columns of X are composed of columns in $D_2 \cup D_3 \cup \cdots \cup D_{t-1}$, as well as some (but not all) columns from D_1 and D_t, such as c_3 and c_{4k}. Equation (8.9) indicates that exactly two defender columns, c_3 and c_{4k}, are in $X - (D_2 \cup D_3 \cup \cdots \cup D_{t-1})$, so

$$
|D_1| = |D_t| = 2 \tag{8.10}
$$

Since $d_3 = p_{3,j} = 0$, (8.2) becomes $|D_j| = p_{2,j}$ and (8.3) becomes $p_{1,j} \leq p_{2,j}$ for $2 \leq j \leq t-1$. Then, by (8.8), we know $p_{1,j} = p_{2,j}$, and

$$
|A_j| = |D_j| \text{ for } 2 \leq j \leq t-1 \tag{8.11}
$$

Recall that Figure 8.8 shows valid partial configurations of a given attacker group in X and the adjacent defender group to its right. Since $p_{3,j} = 0$, patterns (iii), (iv) and (vi) must not appear in X. In addition, since $|A_j| = |D_j|$, patterns (i), (ii) and (v) are *no longer partial* configurations, because any additional defender columns for the defender groups will make $|A_j| < |D_j|$. Then, possible patterns for $A_j \cup D_j$, $2 \leq j \leq t-1$ are those shown in Figure 8.9.

Next, consider $D_j \cup A_{j+1}$ for $2 \leq j \leq t-1$. Note that $D_j \cup A_{j+1}$ are columns in X, and A_{j+1} is an attacker group of type III or IV. Enumerations of possible defender groups to the left of A_{j+1} are exact reflections of those shown in Figure 8.4. Since $d_3 = 0$, the only possible partial patterns for $D_j \cup A_{j+1}$ are shown in Figure 8.10. These patterns are partial because they contain the entire attacker group, but may contain only part of the defender group. Nonetheless, $|D_j| \geq |A_{j+1}|$ for $2 \leq j \leq t-1$.

Notice from Figure 8.3 that $|A_2| \leq 2$. Since $|A_t| = 2$ (Eq. (8.7)), $|A_j| = |D_j|$ (Eq. (8.11)) and $|D_j| \geq |A_{j+1}|$ for $2 \leq j \leq t-1$, we have $2 \geq |A_2| = |D_2| \geq |A_3| = |D_3| \geq \cdots \geq |A_{t-1}| = |D_{t-1}| \geq |A_t| = 2$. Thus, $|A_j| = |D_j| = 2$ for $2 \leq j \leq t-1$. Along with $|A_1| = 2$, $|A_t| = 2$ and $|D_1| = |D_t| = 2$ (Eq. (8.10)), the result is $|A_j| = |D_j| = 2$ for $1 \leq j \leq t$. This implies $4k + 2 = \sum_{j=1}^{t} |A_j| + |D_j| = 4t$, which is not possible. $\square$

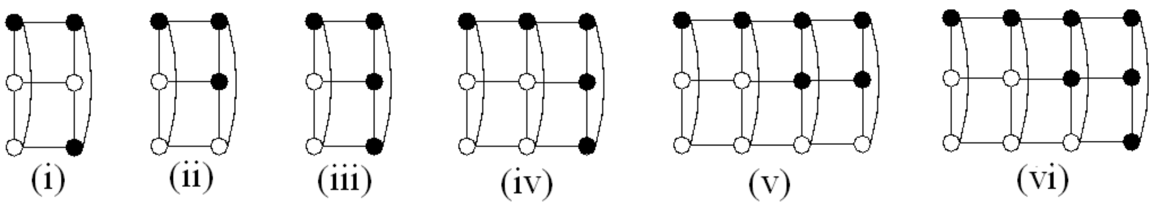

Figure 8.8: Valid partial configurations for $A \cup D$, if A is of type III or IV. See the proof of Lemma 8.2.7 for details.

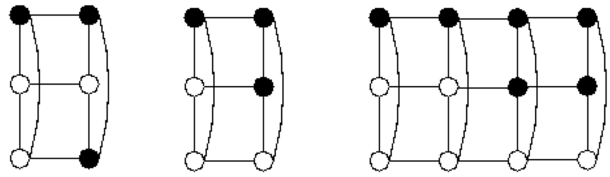

Figure 8.9: Possible configurations for $A_j \cup D_j$, $2 \leq j \leq t-1$. These configurations are *not* partial. See the proof of Lemma 8.2.7 for details.

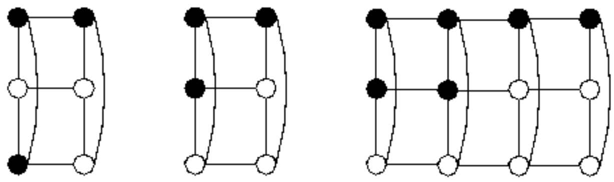

Figure 8.10: Possible partial configurations for $D_j \cup A_{j+1}$, $2 \leq j \leq t-1$. See the proof of Lemma 8.2.7 for details.

Theorem 8.2.8. $\gamma_s(C_3 \times C_{4k+2}) = 6k + 4$ for $k \geq 1$.

Proof. By Lemma 7.5.1, $\gamma_s(C_3 \times C_{4k+2}) \leq 6k + 4$. We prove $\gamma_s(C_3 \times C_{4k+2}) > 6k + 3$ by induction on the value of k. For $k = 1$, let S' be a minimum global secure set of $C_3 \times C_6$. If there is at most one attacker group of type II in $C_3 \times C_6$, by Lemmas 8.2.5 and 8.2.7, $|S'| > 9$. Otherwise, there are at least two attacker groups of type II in $C_3 \times C_6$, but then $|V(C_3 \times C_6) - S'| \geq 12$ and $|S'| \leq 6$, an impossibility.

By the way of induction, assume $\gamma_s(C_3 \times C_{4k'+2}) > 6k' + 3$ for $1 \leq k' < k$. Then, let S be a minimum global secure set of $C_3 \times C_{4k+2}$. By Lemmas 8.2.5 and 8.2.7, if there is at most one attacker group of type II, then $|S| > 6k + 3$.

Consider the remaining case where S contains at least two attacker groups of type II. Let $c_1, c_2, \ldots, c_{4k+2}$ be the columns of $C_3 \times C_{4k+2}$. More specifically, let $c_1, c_i, c_{i+1}, c_{4k+2}$ be empty columns. Columns $\{c_1, c_{4k+2}\}$ and $\{c_i, c_{i+1}\}$ form two attacker groups of type II. Since each attacker group is maximal, the two groups cannot be adjacent, so $2 < i < i+1 < 4k+1$. Let $X_1 = c_1 \cup c_2 \cup \cdots \cup c_i$ and $X_2 = c_{i+1} \cup \cdots \cup c_{4k+2}$. Let a_j and d_j denote respectively the number of attackers and defenders in X_j, for $j \in \{1, 2\}$.

Because c_1, c_i, c_{i+1} and c_{4k+2} are empty columns, every attacker in X_1 must be dominated by a defender in $X_1 \cap S$, and may attack a vertex in $X_1 \cap S$. Since S is secure, $X_1 \cap S$ is a global secure set of $C_3 \times C_i$ and $a_1 \leq d_1$. Similarly, $X_2 \cap S$ is a global secure set of $C_3 \times C_{4k+2-i}$ and $a_2 \leq d_2$.

Assume $|S| = 6k+3$. Then, $a_1 + a_2 = d_1 + d_2$. Since $a_1 \leq d_1$ and $a_2 \leq d_2$, it follows that $a_1 = d_1$ and $a_2 = d_2$. The number of vertices in X_1 is $3i = a_1 + d_1 = 2d_1$. So, i, and thus $4k+2-i$ are even. Then, either $i \equiv 2 \pmod 4$ or $(4k+2-i) \equiv 2 \pmod 4$. So, either $X_1 \cap S$ or $X_2 \cap S$ is a counter example to the inductive hypothesis. Thus, the assumption $|S| = 6k+3$ is false and $|S| > 6k+3$. $\square$

Theorem 8.2.8 concludes the global security number of $C_3 \times C_{4k+2}$ and completes Chapters 7 and 8. The list below collects results regarding the global security numbers of grid-like graphs, with the corresponding theorem (or lemma) treating each case given in parenthesis.

The global security numbers (γ_s) of grid-like graphs are as follows.

1. $\gamma_s(P_n) = \lceil n/2 \rceil$. (Theorem 7.2.1)

2. $\gamma_s(C_n) = \left\{ \begin{array}{lll} 2k+2 & \text{if } n \in \{4k+2 : k \geq 1\} & \text{(Theorem 8.1.2)} \\ \lceil n/2 \rceil & \text{otherwise} & \text{(Lemma 7.2.2)} \end{array} \right\}$

3. $\gamma_s(P_n \times P_m) = \lceil nm/2 \rceil$. (Theorem 7.3.34)

4. $\gamma_s(P_n \times C_m) = \lceil nm/2 \rceil$. (Theorem 7.4.12)

5. (a) $\gamma_s(C_3 \times C_{4k+2}) = 6k+4$. (Theorem 8.2.8)

   (b) $\gamma_s(C_n \times C_m) = \lceil nm/2 \rceil$ with the exception of $C_3 \times C_{4k+2}$. (Theorem 7.5.7, Part 2)

Note that the result shown above is equivalent to Theorem 7.1.1 stated at the beginning of Chapter 7.

CHAPTER 9

SUMMARY AND OPEN PROBLEMS

9.1 Summary

Let $G = (V, E)$ be a connected graph. A vertex subset S is a defensive alliance if $\forall x \in S, |N[x] \cap S| \geq |N[x] - S|$. In a graph model, the vertices represent countries and the edges represent country boundaries. In a defensive alliance S, if a single vertex $x \in S$ is *attacked* by its neighbors outside the alliance, the attack can be thwarted by x with the assistance of its neighbors inside the alliance. So, a defensive alliance can successfully defend against attacks on a single member.

In a more realistic setting, multiple members of an alliance may be attacked at the same time by their neighbors outside. As seen in Chapter 1, a defensive alliance may not be able to defend against simultaneous attacks on multiple members. The theory of secure sets in graphs are developed for modeling such situations. In the context of secure sets, every vertex $y \in N[S] - S$ can choose to attack one vertex in $N(y) \cap S$, and given these choices, every vertex in $x \in S$ can choose to defend one vertex in $N[x] \cap S$. The attack is defended if every vertex in S receives as many defenders as attackers. A secure set can defend against any attack under these assumptions. The formal definition is given in Definition 1.2.1 (originally appeared in [BDH07], Definition 1). A characterization of secure sets is given in Theorem 1.2.3 (originally appeared in [BDH07], Theorem 11). The characterization suggests that a set S is a secure set if and only if every subset of S has as many neighbors inside S compare to outside.

If a defensive alliance is also a dominating set, then it is a global defensive alliance. Similarly, if a secure set is also a dominating set, then it is a global secure set. Determining the existence of small defensive alliances and small global defensive alliances have shown to be NP-Complete in [JHM09, CBD06, Enc09] (Section 2.2). In Chapter 3, we explored the complexity of problems related to secure sets. The problem of finding a feasible defense for a given attack (or determine none exists) (`Feasible Defense`, Problem 3.1.1) can be solved in polynomial time using network flow (Section 3.2) or maximum matching (Section 4.2)[1]. On the other hand, the problem of verifying the validity of a given secure set (`Is Secure`, Problem 3.1.2), and the problem of finding a minimum secure set (`Secure Set`, Problem 3.1.3) are both in P if and only if P = NP (Corollary 3.3.8 and Theorem 3.4.4).

In Chapter 4, we presented a proof of the secure set characterization (Theorem 1.2.3) as an application of Hall's Matching Theorem.

Chapters 5 and 6 investigated problems relate to secure sets in trees. Chapter 5 presented an $O(n\Delta)$ algorithm (Algorithm 5.2.3) and an $O(n \lg(\Delta))$ algorithm (Algorithm 5.4.6) for finding a minimum rooted secure set (`Rooted Secure Set`, Problem 5.1.1) of a tree. A

[1] An earlier solution using integer programming appeared in [BDH07], Section 3.

rooted secure set is a secure set with the additional requirement that a specific vertex must be included in the set. In Chapter 6, Section 6.1 presented an $O(n\Delta)$ algorithm (Algorithm 6.1.4) for finding a minimum global secure set of a tree, and in Section 6.2 we showed that $\lceil n/2 \rceil \leq \gamma_s(T) \leq \lfloor 2n/3 \rfloor$ for any tree T of order $n \geq 2$, and both bounds are sharp (Theorem 6.2.9).

Chapters 7 and 8 investigated global secure sets of grid-like graphs. Grid-like graphs are paths, cycles and their Cartesian products (Definition 1.4.3). We showed that for any grid-like graph G of order n, $\gamma_s(G) = \lceil n/2 \rceil$, with two exceptions C_{4k+2} and $C_3 \times C_{4k+2}$, and in those cases $\gamma_s(G) = n/2 + 1$. Chapter 7 constructed minimum global secure sets for each of the grid-like graphs. Chapter 8 verified the two exceptions, showing that no global secure set of cardinality $n/2$ exists for C_{4k+2} or $C_3 \times C_{4k+2}$.

The remainder of this chapter discusses several open problems and possible future developments for secure sets in graphs.

9.2 Bounds

Let $G = (V, E)$ be a connected graph of order n.

Problem 9.2.1. Find a sharp upper bound on the security number of G in terms of n.

In [FLH03], it was shown that $\lceil n/2 \rceil$ is a sharp upper bound on the defensive alliance number of G. In [DLB08], it was shown that $\lceil n/2 \rceil$ is *not* an upper bound on the security number of G. In particular, the class of Kneser graphs $K(m, 2)$ with $m \geq 6$ have security number $\lceil (n+1)/2 \rceil$[2]. It is not known whether $\lceil (n+1)/2 \rceil$ is a sharp upper bound on the security number of G, and there is no known example for which the security number of a graph of order n is larger than $\lceil (n+1)/2 \rceil$.

Problem 9.2.2. Find a sharp upper bound on the global security number of G in terms of n.

In Section 6.2, it is shown that $\lfloor 2n/3 \rfloor$ is a sharp upper bound on the global security number of a tree of order $n \geq 2$. It is not known whether this is also an upper bound for general graphs. There is no known example for which the global security number of a connected graph of order $n \geq 2$ is larger than $2n/3$.

Problem 9.2.3. ([BDH07]) Is $s(G) + s(\overline{G}) \leq n + 1$ a correct Nordhaus-Gaddum bound for graphs of order n?

9.3 Algorithms and complexity

Problem 9.3.1. Design a polynomial algorithm for finding the security number of a series-parallel graph.

A polynomial algorithm for finding the defensive alliance number of a series-parallel graph is given in [Jam07].

[2]See Section 2.1 for details.

Problem 9.3.2. Design a polynomial algorithm for finding the global security number of a series-parallel graph.

Problem 9.3.3. Design a polynomial algorithm for finding the global security numbers of graphs with bounded treewidth.

In [Enc09], the author presents a polynomial algorithm for finding minimum global defensive alliances on graphs with fixed treewidth and maximum degree (i.e., fixed domino treewidth). The class of partial k-trees exhibits polynomial solutions for many NP-Complete problems ([BLW87, Wim87]). It is interesting to see if `Secure Set` (Problem 1.4.1) and `Global Secure Set` (Problem 1.4.2), two problems that may not belong to NP, can be solved in polynomial time on these graphs.

Problem 9.3.4. Prove or disprove: if `Global Secure Set` (Problem 1.4.2) is in P, then P = NP.

The converse of 9.3.4 is true (Corollary 3.4.7). The problem `Secure Set` (Problem 3.1.3) is in P if and only if P = NP (Theorem 3.4.4).

9.4 Problems related to secure sets

This section discusses two categories of problems related to secure sets in graphs. The first category introduces inclusion and exclusion requirements for particular vertices of the graph, similar to the problem `Rooted Secure Set` (Problem 5.1.1).

Problem 9.4.1.

Given: A graph $G = (V, E)$, sets $A, B \subseteq V$ with $A \cap B = \emptyset$, and integer $k < |V|$.

Question: Is there a secure set $S \subseteq V$ of cardinality k or less, such that $A \subseteq S$ and $B \cap S = \emptyset$?

Problem 9.4.1 asks for a secure set of cardinality k or less, such that every vertex in A is in the set, and every vertex in B is not in the set. This is a more general model (compared to `Rooted Secure Set`) for an application, in situations where some vertices are desired to be included in a secure set (e.g., friendly or neutral nations, points of interest, etc.), while some other vertices are not desired (e.g., hostiles, entities sharing opposed interest, etc.). The problem `Rooted Secure Set` (Problem 5.1.1) presented in Chapter 5 is a special case where $A = \{r\}$ and $B = \{\}$.

The second category introduces probability and random process into the problem of secure sets. Let $G = (V, E)$ be a connected graph and let S be a non-empty subset of k vertices. Let $\mathcal{P}(S)$ denote the power set of S.

Recall from Chapter 3 that verifying the validity of a secure set is in P if and only if P = NP (Corollary 3.3.8). So, there is no known polynomial algorithm (and none exists unless P = NP) to determine whether S is a secure set. In the case that S is not a secure set, S may still be able to defend a large number of attacks. Although there exists an attack for which S cannot be defended, an adversary who seeks such an attack will not be able to identify it efficiently.

There are two primary methods for identifying an attack where S cannot defend. The first method is to simply identify an attack $\mathscr{A}$ on S according to Definition 1.2.1, and verify that $\mathscr{A}$ is not defendable using polynomial algorithms presented in Sections 3.2 or 4.2. Note that each $y \in N[S] - S$ has $|N(y) \cap S|$ different choices on which vertex of S to attack, and there are a total of $K = \prod_{y \in (N[S]-S)} |N(y) \cap S|$ different attacks on S. The second method is to identify a *witness*, a subset $W \subseteq S$ such that $|N[W] \cap S| < |N[W] - S|$ (Theorem 1.2.3). Definition 9.4.2 defines when the set S is *probably* secure, according to the discussions so far.

Definition 9.4.2. With reference to the last three paragraphs, the set S is *probably secure* with probability p if an attack chosen uniformly at random from the set of K possible attacks can be defended with probability at least p, *and* a non-empty subset of S chosen uniformly at random from $\mathcal{P}(S) - \{\emptyset\}$ is not a witness with probability at least p.

Definition 9.4.2 states that if S is a probably secure set (with probability p), then at least pK of the K possible attacks on S must be defendable, *and* at least $p|\mathcal{P}(S) - \{\emptyset\}| = p(2^k - 1)$ non-empty subsets of S satisfy $|N[X] \cap S| \geq |N[X] - S|$. Thus, an adversary who attempts to identify an attack on S that is not defendable will succeed with probability at most $1 - p$ per trial (if S is in fact not secure). Here, we assume that when testing for the security of S, the adversary either selects an attack on S uniformly at random (where each attacker chooses to attack one of its $|N(y) \cap S|$ neighbors uniformly at random) and check whether it is defendable, or selects a non-empty subset of S uniformly at random and check whether it is a witness.

LIST OF REFERENCES

[BCH10] Mohamed Bouzefrane, Mustapha Chellali, and Teresa W. Haynes. "Global defensive alliances in trees." *Util. Math.*, **82**:241–252, 2010.

[BDH07] Robert C. Brigham, Ronald D. Dutton, and Stephen T. Hedetniemi. "Security in graphs." *Discrete Appl. Math.*, **155**(13):1708–1714, 2007.

[BDH09] Robert C. Brigham, Ronald D. Dutton, Teresa W. Haynes, and Stephen T. Hedetniemi. "Powerful alliances in graphs." *Discrete Mathematics*, **309**:2140–2147, 2009.

[BGH08] Jean R.S. Blair, Wayne Goddard, Stephen T. Hedetniemi, Steve Horton, Patrick Jones, and Grzegorz Kubicki. "On domination and reinforcement numbers in trees." *Discrete Mathematics*, **308**(7):1165–1175, April 2008.

[BLW87] M. W. Bern, E. L. Lawler, and A. L. Wong. "Linear-time computation of optimal subgraphs of decomposable graphs." *Journal of Algorithms*, **8**(2):216–235, June 1987.

[CBD06] Aurel Cami, Hemant Balakrishnan, Narsingh Deo, and Ronald D. Dutton. "On the Complexity of Finding Optimal Global Alliances." *Journal of Combinatorial Mathematics and Combinatorial Computing*, **58**:23–31, 2006.

[CFM03] E. J. Cockayne, O. Favaron, and C. M. Mynhardt. "Secure domination, weak Roman domination and forbidden subgraphs." *Bull. Inst. Combin. Appl.*, **39**:87–100, 2003.

[CJH04] Ernie J. Cockayne, Paul A. Dreyer Jr., Sandra M. Hedetniemi, and Stephen T. Hedetniemi. "Roman domination in graphs." *Discrete Mathematics*, **278**(1-3):11–22, March 2004.

[CLR01] Thomas H. Cormen, Charles Eric Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill Book Company, second edition, 2001.

[CMR07] Rodolfo Carvajal, Martin Matamala, Ivan Rapaport, and Nicolas Schabanel. "Small Alliances in Graphs." *Lecture Notes in Computer Science*, **4708**:218–227, 2007.

[DDH09] John Dabney, Brian C. Dean, and Stephen T. Hedetniemi. "A linear-time algorithm for broadcast domination in a tree." *Netw.*, **53**(2):160–169, 2009.

[DH10] Ronald D. Dutton and Yiu Yu Ho. "Global secure sets in maximum degree three graphs." *in preparation*, 2010.

[DLB08] Ronald D. Dutton, Robert Lee, and Robert C. Brigham. "Bounds on a graph's security number." *Discrete Applied Mathematics*, **156**:695–704, 2008.

[Dut06] Ronald D. Dutton. "Secure set algorithms and complexity." *Congr. Numer.*, **180**:115–121, 2006.

[Dut09] Ronald D. Dutton. "On a graph's security number." *Discrete Mathematics*, **309**(13):4443–4447, 2009.

[ED08] Rosa I. Enciso and Ronald D. Dutton. "Parameterized complexity of secure sets." *Congr. Numer.*, **189**:161–168, 2008.

[EG08] Linda Eroh and Ralucca Gera. "Global alliance partition in trees." *J. Combin. Math. Combin. Comput.*, **66**:161–169, 2008.

[Enc09] Rosa I. Enciso. *Alliances in Graphs: Parameterized Algorithms and on Partitioning Series-Parallel Graphs*. PhD thesis, University of Central Florida, 2009.

[FF56] Lester Randolph Ford Jr. and Delbert Ray Fulkerson. "Maximal flow through a network." *Canadian Journal of Mathematics*, **8**:399–404, 1956.

[FFG04] Odile Favaron, Gerd Fricke, Wayne Goddard, Sandra M. Hedetniemi, Stephen T. Hedetniemi, Petter Kristiansen, Renu C. Laskar, and R. Duane Skaggs. "Offensive alliances in graphs." *Discuss. Math. Graph Theory*, **24**(2):263–275, 2004.

[FGK08] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. "Solving Connected Dominating Set Faster than 2^n." *Algorithmica*, **52**(2):153–166, 2008.

[FGK09] Fedor V. Fomin, Fabrizio Grandoni, and Dieter Kratsch. "A measure & conquer approach for the analysis of exact algorithms." *J. ACM*, **56**(5):1–32, 2009.

[FGP08] Fedor V. Fomin, Fabrizio Grandoni, Artem V. Pyatkin, and Alexey A. Stepanov. "Combinatorial bounds via measure and conquer: Bounding minimal dominating sets and applications." *ACM Trans. Algorithms*, **5**(1):1–17, 2008.

[FLG00] G. W. Flake, S. Lawrence, and C. L. Giles. "Efficient identification of web communities." *ACM SIGKDD*, pp. 150–160, 2000.

[FLH03] Gerd H. Fricke, Linda M. Lawson, Teresa W. Haynes, Sandra M. Hedetniemi, and Stephen T. Hedetniemi. "A note on defensive alliances in graphs." *Bull. Inst. Combin. Appl.*, **38**:37–41, 2003.

[FR07] H. Fernau and D. Raible. "Alliances in graphs: A complexity-theoretic study." *Software Seminar, Student Research Forum*, **Proceedings Vol. II**:61–70, 2007.

[FRS09] Henning Fernau, Juan A. Rodriguez, and Jose M. Sigarreta. "Offensive r-alliances in graphs." *Discrete Appl. Math.*, **157**(1):177–182, 2009.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness.* W. H. Freeman, 1979.

[GM09] P.J.P. Grobler and C.M. Mynhardt. "Secure domination critical graphs." *Discrete Mathematics*, **309**(19):5820 – 5827, 2009. Structural Graph Theory - Dedicated to Professor Gary Chartrand in Honor of his 70th Birthday.

[Gol10] Oded Goldreich. *P, NP, and NP-Completeness: The Basics of Computational Complexity.* Cambridge University Press, first edition, August 2010.

[Gra06] Fabrizio Grandoni. "A note on the complexity of minimum dominating set." *Journal of Discrete Algorithms*, **4**(2):209–214, June 2006.

[GY03] Jonathan L. Gross and Jay Yellen, editors. *Handbook of Graph Theory (Discrete Mathematics and Its Applications).* CRC, 2003.

[Hal35] P. Hall. "On Representatives of Subsets." *Journal of the London Mathematical Society*, **s1-10**(1):26–30, 1935.

[HD10] Yiu Yu Ho and Ronald D. Dutton. "Constructions of Global Secure Sets of Grid Like Graphs." Technical Report 01, University of Central Florida, 2010. See http://www.eecs.ucf.edu/techreports/.

[HD11] Yiu Yu Ho and Ronald D. Dutton. "Global secure sets of grid-like graphs." *Discrete Applied Mathematics*, **159**:490–496, March 2011.

[HHH03] T. W. Haynes, S. T. Hedetniemi, and M. A. Henning. "Global defensive alliances in graphs." *The Electronic Journal of Combinatorics*, **10**:#R47, 2003.

[HHS98a] Teresa W. Haynes, S. T. Hedetniemi, and Peter J. Slater. *Fundamentals of domination in graphs.* CRC Press, 1998.

[HHS98b] Teresa W. Haynes, Stephen Hedetniemi, and Peter Slater. *Domination in Graphs: Advanced Topics.* CRC Press, 1998.

[HL07] T. W. Haynes and J. A. Lachniet. "The alliance partition number of grid graphs." *AKCE J. Graph. Combin.*, **4**(1):51–59, 2007.

[HWW09] Cheng-Ju Hsu, Fu-Hsing Wang, and Yue-Li Wang. "Global defensive alliances in star graphs." *Discrete Applied Mathematics*, **157**(8):1924 – 1931, 2009.

[Jam07] Lindsay Harris Jamieson. *Algorithms and complexity for alliances and weighted alliances of various types.* PhD thesis, Clemson University, Clemson, SC, USA, 2007. Adviser-Hedetniemi, Stephen.

[JD07] Lindsay H. Jamieson and Brian C. Dean. "Weighted alliances in graphs." *Congr. Numer.*, **187**:76–82, 2007.

[Jes10] Katarzyna Jesse-Jozefczyk. "The possible cardinalities of global secure sets in cographs." *manuscript*, 2010.

[JHM09] Lindsay H. Jamieson, Stephen T. Hedetniemi, and Alice A. McRae. "The algorithmic complexity of alliances in graphs." *J. Combin. Math. Combin. Comput.*, **68**:137–150, 2009.

[KHH04] Petter Kristiansen, Sandra M. Hedetniemi, and Stephen T. Hedetniemi. "Alliances in graphs." *J. Combin. Math. Combin. Comput.*, **48**:157–177, 2004.

[KM08] William F. Klostermeyer and Christina M. Mynhardt. "Secure domination and secure total domination in graphs." *Discuss. Math. Graph Theory*, **28**(2):267–284, 2008.

[Knu98] Donald E. Knuth. *Art of Computer Programming, Volume 3: Sorting and Searching.* Addison-Wesley, second edition, 1998.

[KOY09] Kyohei Kozawa, Yota Otachi, and Koichi Yamazaki. "Note: Security number of grid-like graphs." *Discrete Appl. Math.*, **157**(11):2555–2561, 2009.

[KT05] Jon Kleinberg and Eva Tardos. *Algorithm Design.* Addison Wesley, 2005.

[RA07] Juan Alberto Rodriguez-Velazquez and Jose Maria Sigarreta Almira. "Spectral study of alliances in graphs." *Discuss. Math. Graph Theory*, **27**:143–157, 2007.

[Rav93] James B. Orlin Ravindra K. Ahuja, Thomas L. Magnanti. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[RB08] Johan M. M. van Rooij and Hans L. Bodlaender. "Design by Measure and Conquer, A Faster Exact Algorithm for Dominating Set." In Susanne Albers and Pascal Weil, editors, *25th International Symposium on Theoretical Aspects of Computer Science (STACS 2008)*, volume 1 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 657–668, Dagstuhl, Germany, 2008. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Keywords: Exact algorithms, exponential time algorithms, branch and reduce, measure and conquer, dominating set, computer aided algorithm design.]

[RS06] J. A. Rodriguez-Velazquez and J. M. Sigarreta. "Global offensive alliances in graphs." *Electron. Notes Discrete Math.*, **25**, 2006.

[RS09] J.A. Rodrguez-Velzquez and J.M. Sigarreta. "Global defensive k-alliances in graphs." *Discrete Applied Mathematics*, **157**:211–218, 2009.

[RYS09] Juan A. Rodrguez-Velzquez, Ismael G. Yero, and Jos M. Sigarreta. "Defensive k-alliances in graphs." *Applied Mathematics Letters*, **22**:96–100, 2009.

[Sha04] Khurram H. Shafique. *Partitioning a Graph in Alliances and its Application to Data Clustering*. PhD thesis, University of Central Florida, 2004.

[SR09] J. M. Sigarreta and J. A. Rodríguez. "On the global offensive alliance number of a graph." *Discrete Appl. Math.*, **157**(2):219–226, 2009.

[Wes04] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, second edition, 2004.

[WHL85] T. V. Wimer, S. T. Hedetniemi, and R. Laskar. "A methodology for constructing linear graph algorithms." *Congr. Numer.*, **50**:43–60, 1985.

[Wim87] Thomas Victor Wimer. *Linear algorithms on k-terminal graphs*. PhD thesis, Clemson University, Clemson, SC, USA, 1987.