STUDENT MATHEMATICAL LIBRARY
Volume 61

The Game of Cops and Robbers on Graphs

Anthony Bonato
Richard J. Nowakowski

AMS
AMERICAN MATHEMATICAL SOCIETY

The Game of Cops and Robbers on Graphs

STUDENT MATHEMATICAL LIBRARY
Volume 61

The Game of Cops and Robbers on Graphs

Anthony Bonato
Richard J. Nowakowski

AMS
American Mathematical Society
Providence, Rhode Island

2010 *Mathematics Subject Classification*. Primary 05C57, 91A43, 05C75, 05C80, 05C63, 05C85.

For additional information and updates on this book, visit
www.ams.org/bookpages/stml-61

Library of Congress Cataloging-in-Publication Data

Bonato, Anthony, 1971–
The game of cops and robbers on graphs / Anthony Bonato, Richard J. Nowakowski.
p. cm. — (Student mathematical library ; v. 61)
Includes bibliographical references and index.
ISBN 978-0-8218-5347-4 (alk. paper)
1. Graph theory. 2. Random graphs. 3. Graph algorithms. I. Nowakowski, Richard J. II. Title.

QA166.B667 2011
511′.5—dc22

2011014177

∞ The paper used in this book is acid-free and falls within the guidelines
established to ensure permanence and durability.
Visit the AMS home page at http://www.ams.org/

10 9 8 7 6 5 4 3 2 1　　16 15 14 13 12 11

Dedicated to the memory of
Paolo Giovanni Bonato and Marian Jozef Nowakowski

Contents

List of Figures

Preface

You are reading a book about a game. More specifically, the game Cops and Robbers, which is played on a graph. Cops and Robbers, in the form we study it, was first introduced in the early 1980s, and a robust body of work on the topic has been growing steadily ever since. At its core, it is a game played with a set of cops (controlled by one player) trying to capture the robber (controlled by the opposing player). The cops and the robber are restricted to vertices, and they move each round to neighboring vertices. The smallest number of cops needed to capture the robber is the *cop number*. Such a simple-sounding game leads to quite a complex theory, as you will learn. A formal introduction to the game and the cop number is given in Chapter 1. Despite the fact that the game is nearly three decades old, the last five years however, have seen an explosive growth in research in the field. Some newer work settles some old problems, while novel approaches, both probabilistic, structural, and algorithmic, have emerged on this classic game on graphs.

We present a book which surveys all of the major developments (both historical and recent) on the topic of Cops and Robbers. As the moniker "Cops and Robbers" represents a class of games with varying rules, we emphasize that we primarily study the game where the cops and robber have perfect information, may only move to neighboring vertices, and move at unit speed (a player can only move at distance at most one at any step of the game). There is a large and growing literature on variants of the Cops and Robbers game, where there is some notion of “good guys” versus “bad guys”. For example, there are versions where there is imperfect information, players can occupy edges or only a subset of vertices, move at faster speeds, or the cops are trying to stop or contain a fire, disease, or contaminant spreading in a graph.

Although these games are not our main focus, we do discuss some variants in Chapters 8 and 9. There are a number of reasons why we wrote this book. One of our goals was to bring together all the most important results, problems, and conjectures in one place to serve as a reference. Hence, this book will be both invaluable to researchers in the field and their students, and a one-stop shop for the major results in the field. We also wanted the book to be self-contained and readable to an advanced undergraduate or beginner graduate student; on the other hand, there are enough advanced topics to either intrigue the seasoned mathematician or theoretical computer scientist. The book is designed to be used either in a course or for independent reading and study. The only prerequisites would be a first course in graph theory, though some mathematical maturity and some background on sets, probability, and algorithms would be helpful. One of our principal goals is to showcase the beauty of the topic, with the ultimate aim of preserving it for the next generations of graph theorists and computer scientists. We also showcase the most challenging open problems in the field. For example, Meyniel’s conjecture on upper bounds for the cop number (see Chapter 3) is a deep problem which deserves to be better known.

We now give a summary of the chapters. Chapter 1 supplies all the requisite motivation, notation, basic results, and examples for what comes later. We give a lower bound of Aigner and Fromme on the cop number in terms of girth and minimum degree, and we give the asymptotic upper bound on the cop number supplied by Frankl. Along the way, we discuss guarding isometric paths, and retracts and their critical connections to the game. In Chapter 2 we consider some new and old characterizations of k-cop-win graphs. We describe in detail the classic characterization of Nowakowski and Winkler and Quilliot of finite cop-win graphs. This beautiful characterization reduces the problem to the existence of a certain ordering of the vertices, a so-called cop-win ordering. It also leads to a strategy for catching the robber called the cop-win strategy. We survey the recent characterization of graphs with cop number $k > 1$ by Clarke and MacGillivray. This characterization uses, among other things, properties of graph products. Chapter 3 is all about Meyniel's conjecture, which concerns an upper bound on the cop number in connected graphs. We give some recent upper bounds and discuss the state-of-the-art on the conjecture. We present a recent proof of the conjecture in the special case of graphs of diameter at most 2. Chapter 4 focuses on the game in graph classes and graph products. We consider bounds for the cop number and related parameters for various products, such as the Cartesian, strong, categorical, and lexicographic products. A proof of the fact that the cop number of planar graphs is at most three is given, and graphs with higher genus are also discussed.

In Chapter 5, we consider algorithmic results on computing the cop number. After an introduction to the rudiments of complexity theory and graph algorithms, we prove that the problem of computing whether the cop number is at most k is in polynomial time, if k is fixed. If k is not fixed, we sketch the proof of the recent result that the problem is **NP**-hard. In Chapter 6 we investigate the cop number in random graphs. We present results for the cop number of the binomial random graph $G(n, p)$, when p is constant, and also consider recent work in case $p = p(n)$ is a function of n. We culminate with the beautiful Zig-Zag Theorem of Łuczak and Prałat, which reveals a surprising, literal twist to the behaviour of the cop number in random graphs. We finish with a study of the cop number in models for the web graph and other complex networks. In Chapter 7 we study the game of Cops and Robbers played in infinite graphs. Infinite graphs often exhibit unusual properties not seen in the finite case; the cop number in the infinite case is no exception to this. We introduce the cop density of a countable graph and show that the cop density of the infinite random graph can be any real number in $[0, 1]$. We survey the results of Hahn, Sauer, and Woodrow on infinite chordal cop-win graphs. We finish the chapter with a discussion of paradoxically large families of infinite vertex-transitive cop-win graphs. Chapters 8 and 9 consider variants of the game, and are more like surveys when compared with previous chapters. In Chapter 8, we consider the effect of changing the rules of Cops and Robbers. In particular, we consider imperfect information where the robber is partially invisible, and the inclusion of traps, alarms, and photo radar. We consider tandem-cops where cops must always be sufficiently close to each other during the course of the game. In addition, we consider a version of Cops and Robbers where the cops can capture the robber from some prescribed distance (akin to shooting the robber), and we investigate the length of time it takes for the cops to win assuming optimal play. At the heart of all the games we consider, there is the notion of a set of *good guys* trying to stop, contain, or capture a *bad guy*. Chapter 9 deals with several of these kinds of games, including firefighting, edge searching, Helicopter Cops and Robbers, graph cleaning, and robot vacuum. We conclude with a brief section on combinatorial games.

We think the book would make a solid second or topics course in graph theory. An ambitious (likely two-term) course would cover all nine chapters. For a one-term course, we suggest three options: generalist and specialist courses, along with an experiential option. A *generalist* course would cover each of the first four chapters and two of the remaining ones. Such a course would give a solid grounding in the field and supply some flexibility at the end, depending on the tastes of the instructor and audience. A *specialist* course would cover the first two chapters, one of Chapters 3 or 4, and three of the last five chapters. This option would appeal to those would like to learn one of the more advanced topics (such as algorithms, random or infinite graphs) in greater detail. Finally, an *experiential* course would cover Chapters 1, 2, 4, 8, and 9. The emphasis in such a course would be on projects, filling in omitted proofs (Chapters 4, 8, and 9 contain surveys with proofs omitted), coming up with new examples, and developing new variants of Cops and Robbers. This last option would be especially useful in the setting of a summer research project (such as one sponsored by an NSERC USRA or NSF REU).

To both aid and challenge the reader, there are over 200 exercises in the book, with many worked examples throughout. Open problems are cited in the exercises and elsewhere. We will maintain a website

`http://www.math.ryerson.ca/~abonato/copsandrobbers.html`

which will contain resources such as errata and lists of open problems. Hopefully, it will also contain their eventual solutions!

There are many people to thank. We thank Christine Aikenhead, Rebecca Keeping, Margaret-Ellen Messinger, Jennifer Wright Sharp, and Changping Wang for carefully proofreading drafts of the book. Graeme Kemkes, in particular, deserves heartfelt thanks for his very thorough proofreading of early drafts. Any errors or omissions, however, remain the sole responsibilities of the authors! We thank Ina Mette and the wonderful staff at the AMS for their support of this work. A warm thank you to our families, Douglas, Fran, Anna Maria, Paulo, Lisa, Mary, and Marian without whose support writing this book would have been impossible. The authors wish to especially thank the constant and loving support of their fathers, Paulo Giovanni Bonato and Marian Jozef Nowakowski, both of whom died just before the completion of this work.

Chapter 1

Introduction

1.1. The Game

We all grew up playing games. Some of us are lucky enough to play them while working. Such is the case with Cops and Robbers: it is at once a game you can play for fun on a piece of paper with some spare coins and a deep mathematical research topic containing hard conjectures and problems. The purpose of this chapter is a kind of *mezze*: readers will gain the requisite notation and background to tackle the harder topics in later chapters, and also gain some insight into the heart of the game.

To set the stage, do you remember the video game Pac-Man? If you are not a member of the video game generation, then let us recall how it is played. You, Pac-Man, are stuck in a maze. You can move up and down, and across, but not through walls. Unfortunately, there are some attackers in the form of ghosts who are trying to capture you. They do this by touching you, or by *occupying your position in the maze.* Your goal is to eat dots set throughout the maze while avoiding capture. We do not care as much about the dot eating. In some sense, the real goal is to move about the maze unfettered by the ghosts. This is fairly easy with one ghost, but the more ghosts, the greater chance you have of being captured sooner. You can see all

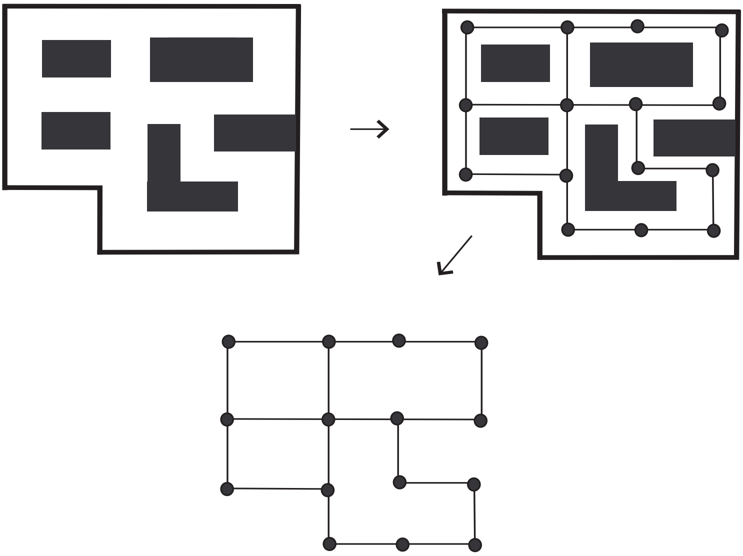


Figure 1.1. A maze and its corresponding graph.

the players loose in the maze and remember all the moves of ghosts (and they can see you).

We may think of the maze as a set of discrete cells, each joined to one above, below, or beside it, assuming there is no wall blocking your way. To help visualize this, see Figure 1.1. For more on this approach in artificial intelligence and so-called *moving target search*, see [**120, 121, 122, 159**]. (In moving target search, octile connected maps which allow diagonal moves are often studied. In this case, a cell becomes a clique of order 4.) Analyzing the movements of players in Pac-Man then becomes a problem about certain kinds of graphs. We focus on a particular view that deviates from the original game somewhat: how many ghosts are needed to ensure they can always capture you, by some strategy? Some mazes require more ghosts, some fewer. For example, think of a very simple maze consisting of a rectangle. One ghost would eternally chase you to no avail, but two can corner you. The game of Cops and Robbers is—in some sense—a discretized version of Pac-Man, and the cop number corresponds to the minimum number of ghosts needed to capture you. You are the intruder, or robber, and the cops are the ghosts.

To be more precise, *Cops and Robbers* (or, as it is sometimes called, *Cops and Robber*) is a game played on a reflexive graph; that is, the vertices each have at least one loop. Multiple edges are allowed but make no difference to the game play, so we always assume there is exactly one edge between adjacent vertices. There are two players consisting of a set of *cops* and a single *robber*. The game is played over a countable sequence of discrete time-steps or *rounds*, with the cops going first in round 0. The cops and robber occupy vertices; for simplicity, we often identify the player by the vertex they occupy. We refer to the set of cops as C and the robber as R. The rules of the game are straightforward: when a player is ready to move in a round, they must move to a neighboring vertex. Because of the loops, players can *pass* or remain on their own vertex. This may or may not be a wise strategy for the robber, depending on the graph. Note that if we play on irreflexive graphs, then we still allow passes. Also observe that any subset of C may move in a given round.

The cops win if after some finite number of rounds, one of them can occupy the same vertex as the robber (in a reflexive graph, this is equivalent to the cop landing on the robber). This is called a *capture.* The robber wins if he (usually the cops are considered female and the robber male) can evade capture indefinitely. A *winning strategy for the cops* is a set of rules that, if followed, result in a win for the cops. A *winning strategy for the robber* is defined analogously. Cops and Robbers is often called a *vertex-pursuit game* on graphs, for reasons that should now be apparent to the reader.

As an elementary but instructive example, consider the game played on a 5-cycle C_5. We label the vertices $1, 2$, 3, 4, and 5, as in Figure 1.2, and place a cop on vertex 1. If the robber chooses 1, then that would be suicide, and choosing vertex 2 or 5 would result in his losing in round 1. The robber chooses 3 and can evade capture in round 1. It is straightforward to see the robber has a winning strategy (just move to $i \pm 1 \pmod 5$ in order to maintain distance two from the cop).

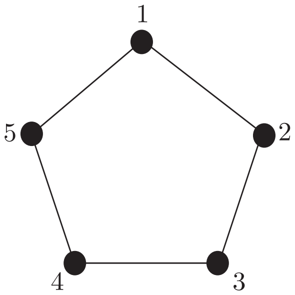


Figure 1.2. A labeled 5-cycle.

Two cops are enough, however, to win. If a second cop occupies 3, then the robber will be caught in round 0 or 1, depending on his initial move. Cycles of size 4 or larger are similar with respect to the game (note that cycles correspond to discretized versions of the simplified rectangular maze we discussed above), because two cops are necessary and sufficient to guarantee a win for the cops.

If we place a cop at each vertex, then the cops are guaranteed to win. Therefore, the minimum number of cops required to win in a graph G is a well-defined positive integer (or infinite cardinal) called the *cop number* (or *copnumber*) of the graph G. We write $c(G)$ for the cop number of a graph G. If $c(G) = k$, then we say G is *k-cop-win*. In the special case $k = 1$, we say G is *cop-win* (or *copwin*). A graph with $c(G) > 1$ is sometimes called *robber-win* (since one cannot capture the robber).

The game of Cops and Robbers was first considered by Quilliot [**169**] in his doctoral thesis, and was independently considered by Nowakowski and Winkler [**167**]. Although [**169**] predates [**167**], the latter reference is sometimes referred to as the starting point of the literature on the topic. The authors of [**167**] were told about the game by G. Gabor. Both [**169**] and [**167**] refer only to one cop. The introduction of the cop number came in 1984 with Aigner and Fromme [**2**]. Many papers have now been written on the cop number of graphs since these three early works; see the surveys [**7**] and [**103**]. For example, at least a dozen theses (at the master's and doctoral level) have been written on the topic; see [**14**], [**51**], [**52**], [**79**], [**97**], [**113**], [**124**], [**158**], [**163**], [**169**], [**170**], [**181**], [**186**], and [**190**].

As an introduction to the topic of Cops and Robbers, we begin this chapter by first covering some notation and definitions from graph theory in Section 1.2. The more advanced reader can skip this, although a casual perusal may eliminate any confusion with notation when reading later sections and chapters. We discuss some examples of cop number in Section 1.3, and include the elementary but helpful Theorem 1.3 which provides a lower bound on the cop number in terms of the minimum degree for graphs without small cycles. In Section 1.4 we prove Frankl's upper bound for the cop number; see Theorem 1.6. Along the way, we will show that one cop can guard an isometric path. We finish with a discussion of retracts in Section 1.5, which play a critical role in the structure of cop-win graphs.

1.2. Interlude on Notation

As we stated in the Preface, we assume (although it is not essential) that the reader has some background in graph theory, such as a first course on the topic. Two good references on the topic are [**68**] and [**197**]. However, as an aid to the reader, we summarize at least some of the notation used as well as some of the requisite background here. As such, the present section is short and may be safely skipped by more advanced readers.

We will use the following notation throughout. The set of natural numbers (which contains 0) is written $\mathbb{N}$, while the rationals and reals are denoted by $\mathbb{Q}$ and $\mathbb{R}$, respectively. The cardinality of $\mathbb{N}$ is $\aleph_0$, while the cardinality of $\mathbb{R}$ is $2^{\aleph_0}$. If $n > 0$ is a natural number, then define

$$[n] = \{1, \ldots n\}.$$

The Cartesian product of two sets A and B is written $A \times B$. The difference of two sets A and B is written $A \backslash B$.

As we will present a number of asymptotic results, we give some corresponding notation. Let f and g be functions whose domain is some fixed subset of $\mathbb{R}$. We write $f \in O(g)$ if

$$\limsup_{x \to \infty} \frac{f(x)}{g(x)}$$

exists and is finite. We will abuse notation and write $f = O(g)$. This is equivalent to saying that there is a constant $c > 0$ (not depending on x) and an integer N such that for $x > N$, $f(x) \leq cg(x)$.

We write $f = \Omega(g)$ if $g = O(f)$, and $f = \Theta(g)$ if $f = O(g)$ and $f = \Omega(g)$. If $\lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| = 0$, then $f = o(g)$ (or $g = \omega(f)$). So if $f = o(1)$, then f tends to 0. We write $f \sim g$ if

$$\lim_{x \to \infty} \left| \frac{f(x)}{g(x)} \right| = 1.$$

If x is a real number, then $1 + x \leq e^x$. We will sometimes write e^x as $\exp(x)$, especially if x is a complicated expression. We write $\log x$ for the logarithm in base e (other bases will be made explicit). If $4 \leq m \leq n$ are non-negative integers, then

$$\binom{n}{m} \leq \frac{n^m}{2^m} \leq n^m. \tag{1.1}$$

For a graph G, we often write $G = (V(G), E(G))$, or if G is clear from context, $G = (V, E)$. The set E may be empty. Elements of $V(G)$ are *vertices*, and elements of $E(G)$ are *edges*. Vertices are sometimes referred to as *nodes*. We write uv for an edge $\{u, v\}$, and say that u and v are *joined* or *adjacent* (we use both terms interchangeably); we say that u is *incident* with v, and that u and v are *endpoints* of uv. All the graphs we consider are reflexive unless otherwise stated.

The cardinality $|V(G)|$ is the *order* of G, while $|E(G)|$ is its *size*. Given a vertex u, define its *neighbor set* $N(u)$ to be the set of vertices joined and not equal to u (also called *neighbors* of u). The *closed neighbor set* of u, written $N[u]$, is the set $N(u) \cup \{u\}$. We write $G \restriction S$ (or as either $\langle S \rangle_G$ or $G[S]$) for the subgraph of G *induced by the set of vertices* S; that is, the graph with vertices in the set S, with two vertices joined if and only if they are joined in G. If S is a set of vertices, then $G - S$ is the subgraph induced by $V(G) \backslash S$; if $S = \{x\}$, then we write this as $G - x$. If H is an induced subgraph of G, then we sometimes write $G - H$ for $G - V(H)$.

The *degree* of a vertex is the cardinal $|N(u)|$, and is written $\deg_G(u)$ or simply $\deg(u)$. A graph is *k-regular* if each vertex has degree k. A *path* is a sequence of vertices such that each vertex is joined to the next vertex in the sequence; the length of a path is the number of its edges. A path of order n is written P_n. A graph is *connected* if there is a path between any two vertices. The relation of being connected by a path is an equivalence relation on V, and the equivalence classes are the *connected components* of G. A graph which is not connected is called *disconnected*; a connected component consisting of a single vertex is called an *isolated vertex*. A *cut vertex* is one whose deletion results in a disconnected graph. A vertex joined to all other vertices is called *universal*. A vertex of degree 1 will be called an *end-vertex*.

A *homomorphism* f from G to H is a function $f : V(G) \to V(H)$ which *preserves edges*; that is, if $xy \in E(G)$, then $f(x)f(y) \in E(H)$. We abuse notation and simply write $f : G \to H$. An *embedding* from G to H is an injective homomorphism $f : G \to H$ with the property that $xy \in E(G)$ if and only if $f(x)f(y) \in E(H)$. We will write $G \leq H$ if there is some embedding of G into H and say that G *embeds in H*. An *isomorphism* is a bijective embedding; if there is an isomorphism between two graphs, then we say they are *isomorphic*. We write $G \cong H$ if G and H are isomorphic. The relation $\cong$ is an equivalence relation on the class of all graphs, whose equivalence classes are *isomorphism types* or *isotypes*. We will always identify a graph with its isomorphism type. An *automorphism* of G is an isomorphism from G to itself. A graph is *vertex-transitive* G if for all pairs of vertices u and v of G, there is an automorphism f of G, so that $f(u) = v$. Note that every vertex-transitive graph is k-regular for some integer $k > 0$.

The *distance* between u and v, written $d_G(u, v)$ (or just $d(u, v)$), is either the length of a shortest path connecting u and v (and 0 if $u = v$) or ∞ otherwise. Note that $d(u, v)$ turns each graph into a metric space. The *diameter* of a connected graph G, written $\mathrm{diam}(G)$, is the supremum of all distances between distinct pairs of vertices. If the graph is disconnected, then $\mathrm{diam}(G)$ is ∞.

The *complement* of G, written $\overline{G}$, has vertices $V(G)$ with two distinct vertices joined if and only if they are not joined in G. A *complete graph of order* n or n*-clique* has all edges present and is written K_n. A set of vertices S is *independent* if $\langle S\rangle_G$ contains no edges. A *co-clique of order* n is $\overline{K_n}$. The graph of order n with no edges is $\overline{K_n}$.

A *wheel of order* n, written W_n, consists of a cycle C_n along with one universal vertex. A *hypercube* of dimension n, written Q_n, has vertices elements of $\{0,1\}^n$ with two vertices joined if they differ in exactly one coordinate.

The *chromatic number* of G, written $\chi(G)$, is the minimum cardinal n with the property that $V(G)$ may be partitioned into n many independent sets; that is, the minimum n so that G has *proper* n*-coloring*. If $\chi(G) = 2$, then G is *bipartite*. A *complete bipartite graph* has all possible edges present between the two colors, and is written $K_{m,n}$, where m and n are the orders of the vertex classes. A *star* is a graph $K_{1,n}$, for some positive integer n.

In a graph G, a set S of vertices is a *dominating set* if every vertex not in S has a neighbor in S. The *domination number* of G, written $\gamma(G)$, is the minimum cardinality of a dominating set. Since placing a cop on each element of a dominating set ensures a win for the cops in at most two rounds, we have that $c(G) \le \gamma(G)$.

Although our primary focus is on undirected graphs, we may sometimes assign orientations to edges. A *directed graph* or *digraph* is defined identically as a graph, except that $E(G)$ consists of ordered pairs of vertices. As with graphs, we assume our directed graphs are reflexive. The edges are then called *directed edges* or *arcs* (u,v), where u is the head and v is the tail. The vertex v is an *out-neighbor* of u, while u is an *in-neighbor* of v. The *in-degree* of u, written $\deg^-(u)$ is the number of vertices v such that (v,u) are directed edges; the *out-degree* $\deg^+(u)$ is defined dually. Subgraphs, induced subgraphs, and isomorphisms are defined analogously to graphs.

A digraph is *oriented* if it is antisymmetric: if (u,v) is a directed edge, then (v,u) is not a directed edge. An *orientation of a graph* is an assignment of directions to the edges resulting in an oriented graph. A *tournament* is an orientation of a clique.

An *order* (or *partially ordered set* or *poset*) is an oriented digraph that is *transitive*: whenever (u, v) and (v, w) are arcs, then so is (u, w). We write $u \leq v$ if (u, v) is an arc in an order. We say that v *covers* u if $u \leq v$, $u \neq v$, and there is no x such that $u \leq x \leq v$. A vertex u is *minimal* if $v \leq u$ implies that $v = u$; *maximal* elements are defined dually. Orders are often represented by *Hasse diagrams*, which are drawings in the plane (although edge crossings are allowed) so that u is below and adjacent to v if v covers u. Note that reflexive and transitive arcs are not shown in Hasse diagrams. See Figure 1.3.

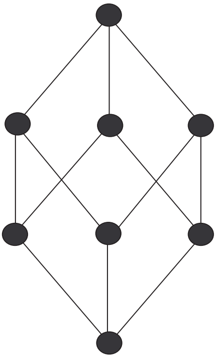


Figure 1.3. The Hasse diagram of an order.

A *directed path* is a path with all directed edges pointing in one direction (so all vertices internal to the path have in- and out-degree equaling one). A *directed cycle* is a cycle with all arcs directed in the same direction. A digraph is *strongly connected* if there is a directed path connecting every pair of vertices. A *weakly connected* digraph has its underlying undirected graph (with no orientations on edges) connected. A digraph is *acyclic* if it contains no directed cycle.

The cop number of directed graphs is defined in the analogous way to the undirected case. The only difference, of course, is that the players can only move following the orientation of a directed edge. A version of Cops and Robbers played on orders will be explored in Exercise 27.

1.3. Lower Bounds

When graph theorists see a new graph parameter, they usually first attempt to compute it for the most common graphs such as cycles, paths, and cliques. The following lemma—whose proof is left as an exercise—does just that.

Lemma 1.1. (1) *For $n > 0$ an integer we have that*

$$c(P_n) = c(W_n) = c(K_n) = 1,$$

and for $n \geq 4$,

$$c(C_n) = 2.$$

(2) *If G is the disjoint union of G_1 and G_2 written $G_1 + G_2$, then*

$$c(G_1 + G_2) = c(G_1) + c(G_2).$$

In particular,

$$c(\overline{K_n}) = n.$$

Owing to Lemma 1.1 (2), we usually restrict our attention to connected graphs. For example, one cop is needed for each isolated vertex, since the robber can occupy one and pass indefinitely. Trees, which are connected and contain no cycles, are a favourite graph class. An infinite one-way path (that is, the vertices of the path are just the non-negative integers, with i joined to $i + 1$ for all $i \in \mathbb{N}$) is called a *ray*, and a graph with no ray is called *rayless*.

Lemma 1.2. (1) *A finite tree is cop-win.*

(2) *The cop number of an infinite tree is either 1 or infinite. It is 1 exactly when the tree is rayless.*

Proof. For item (1), we use the fact that each finite tree contains an end-vertex (finite trees always contain at least two end-vertices; see Exercise 4a). Place the cop on an arbitrary vertex. The strategy of the cop is to move towards the robber on the unique path connecting the cop and robber. Note that with this strategy, $d(C, R)$ never increases. A simple induction establishes that this is possible in any connected graph (roughly put, the robber can never "move around" the cop). However, after some number of rounds (bounded above by diam(T)), the robber will move to an end-vertex. After that round, $d(C, R)$ decreases by one since there is a unique path connecting R and C. Repeating this argument after at most diam(T) $-$ 1 many rounds results in $d(R, C) = 0$, and the cop wins.

A tree of any order with no ray has an end-vertex (see Exercise 9). Now, in a rayless tree, apply the same winning strategy as the one used by the cop in a finite tree. If the tree has a ray and only a finite number of cops are at play, then the robber can always stay a distance of at least one away from any cop. Hence, no winning strategy exists for the cops, and the robber wins. □

End-vertices play a critical role in the proof of Lemma 1.2 (1). They are the simplest examples of *corners*: vertices x with the property that there is some vertex y such that $N[x] \subseteq N[y]$. Corners play a major role in characterizing finite cop-win graphs. See Section 1.5 and Chapter 2 for more discussion.

See Figure 1.4 for an example illustrating Lemma 1.2 (2). This tree is formed by attaching a path of each finite length to a root vertex. The rayless tree in Figure 1.4 has an unusual and vaguely morbid property: the robber in round 0, by choosing which branch to occupy, decides how long he wants to live! Infinite graphs demonstrate many pathological properties, as demonstrated by this example. They therefore deserve special attention and form the focus of Chapter 7. We therefore make the following assumption for the remainder of this

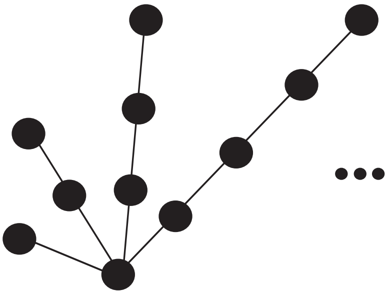


Figure 1.4. A rayless tree.

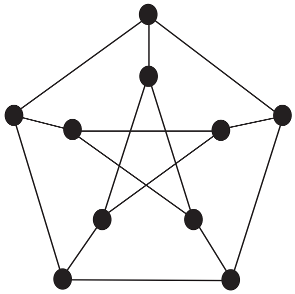


Figure 1.5. The Petersen graph.

chapter and all chapters except Chapter 7: *All graphs we consider are finite.*

One of the most famous graphs is the Petersen graph P; see Figure 1.5. The reader may verify that $c(P) \leq 3$, by placing cops at the bottom two vertices of the outer 5-cycle and the vertex at the top of the middle 5-cycle; see Figure 1.6.

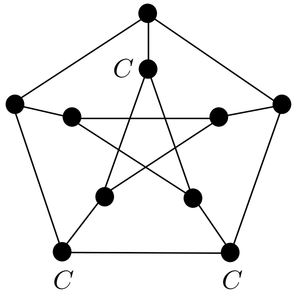


Figure 1.6. Three cops suffice on the Petersen graph.

Direct checking shows that $c(P) > 1$, but to establish $c(P) > 2$ directly requires some case analysis. Instead, we employ the following elementary but useful theorem of Aigner and Fromme [**2**]. The girth of a graph is the length of minimum order cycle. The *minimum degree* of G is written $\delta(G)$, while the *maximum degree* is denoted by $\Delta(G)$.

Theorem 1.3 ([**2**]). *If G has girth at least* 5, *then* $c(G) \geq \delta(G)$.

Proof. Suppose that $\delta(G) = d$ and that $d - 1$ many cops play the game. Most of the proof goes into showing that the robber survives round 1! For this, we must show that there is a vertex of G not joined to a vertex of the set of cops C. Otherwise, C is a dominating set, and we derive a contradiction.

Let u be a vertex outside of C. Suppose that u is joined to $x > 0$ many vertices in C, and y many vertices not in C. Let X be the vertices in C joined to u, and let Y be the set of vertices not in C joined to u. Note that $x + y \geq d$. As C is a dominating set by our assumption, each vertex of Y is joined to some vertex of C. As there are neither three nor four cycles, no vertex of Y is joined to a vertex of X, and no two distinct vertices in Y share a common neighbor in C. Hence, each vertex of Y is joined to a unique vertex of $C \setminus X$. Hence,

$$d - 1 = |C| \geq x + y \geq d,$$

which is a contradiction. Hence, some vertex is not joined to C, and the robber chooses that vertex in round 0.

Now suppose we are in round $t \geq 0$, and the robber has arranged things so he occupies a vertex u_t with the following property.

(C_t): The vertex u_t is not joined to any vertex in C.

In particular, the robber is safe in round t. If (C_t) holds for all $t \geq 0$ and u_t exists, then the robber's winning strategy is to keep moving to the vertices u_t. We prove that (C_t) holds for all t by induction. For the base step, condition (C_t) holds for $t = 0$ by the discussion in the previous paragraph. Suppose it holds for $t - 1$, and so the robber is on a vertex u_{t-1} of G at time t not joined to a cop. As the girth is at least 5, each cop is joined to at most one neighbor of u_{t-1}. As u_{t-1} has degree at least d, the robber simply moves to a vertex u_t not joined to any vertex in C. □

Theorem 1.3 implies that $c(P) \geq 3$, and hence, $c(P) = 3$. Frankl [**89**] proved the following theorem generalizing Theorem 1.3 (which is the case $t = 1$).

Theorem 1.4 ([**2**]). *For a fixed integer $t \geq 1$, if G has girth at least $8t - 3$ and $\delta(G) > d$, then $c(G) > \delta(G)^t$.*

We may use Theorem 1.3 to show that the cop number can be larger than any given integer, a fact not obvious a priori. A *graph class* $\mathcal{C}$ consists of a set of graphs closed under isomorphism. For example, the class of all graphs, bipartite graphs, or triangle-free graphs are graph classes. A graph class $\mathcal{C}$ is *cop-unbounded* if

$$\{c(G) : G \in \mathcal{C}\}$$

is unbounded; otherwise, it is *cop-bounded.*

Theorem 1.5. *The class of bipartite graphs is cop-unbounded.*

Proof. We consider a family of bipartite graphs derived from projective planes which will be important in Chapter 3. A *projective plane* consists of a set of points and lines satisfying the following axioms.

(1) There is exactly one line incident with every pair of distinct points.

(2) There is exactly one point incident with every pair of distinct lines.

(3) There are four points such that no line is incident with more than two of them.

See Figure 1.7 for the projective plane with seven points, called the *Fano plane.*

It can be shown (see [**39**], for example) that a projective plane has $q^2 + q + 1$ points, $q + 1$ many points on a line, and $q + 1$ lines

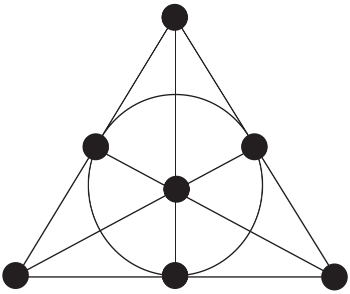


Figure 1.7. The Fano plane.

through a point. The *order* of the plane is q. Projective planes of order q are known to exist for q a prime power, and a deep conjecture is that these are the only possible orders.

We define a bipartite graph $G(P)$ for a given projective plane P. One vertex class consists of points of P, and the other vertex class consists of lines. Vertices of different vertex classes are joined if they are incident. Note that $G(P)$ is $q+1$ regular, and it is not hard to see that $G(P)$ has girth 6. By Theorem 1.3 we have that $c(G(P)) \geq q+1$. We leave it as an exercise that $c(G(P)) \leq q+1$ (this fact was first proved in [**175**]). Since there are infinitely many prime numbers, the proof of the theorem follows. □

Projective planes are one instance of an *incidence structure* (that is, a set of points P and a set of lines L, and a binary incidence relation, which is just a subset of $P \times L$). The construction in the proof of Theorem 1.5 will be further elaborated on in Chapter 3.

Aigner and Fromme [**2**] proved that the cop number of a planar graph is at most three and hence, the class of planar graphs is cop-bounded. Schroeder [**185**] generalized this in another direction by proving that graphs with genus g have cop number bounded by $\lfloor \frac{3g}{2} \rfloor +$ 3. For a fixed graph H, Andreae [**10**] generalized the result on planar graphs in another direction by proving that the cop number of a K_5-minor-free graph (or $K_{3,3}$-minor-free graph) is at most 3 (recall that planar graphs are exactly those that are K_5-minor-free and $K_{3,3}$-minor-free). Andreae [**11**] also proved that for any graph H the class of H-minor-free graphs is cop-bounded. Joret et al. [**125**] proved that a class of graphs defined by omitting a fixed graph H as an induced subgraph is cop-bounded if and only if each component of H is a path. If we consider the class of graphs omitting H as only a subgraph, then the class is cop-bounded if and only if every connected component of H is a tree with at most three leaves. An interesting and more open-ended research problem would be to classify the cop-bounded classes of graphs.

1.4. Upper Bounds

Not many good upper bounds on the cop number are known. For example, we have that $c(G) \leq \gamma(G)$, but these parameters can be arbitrarily far apart. For example, $c(P_n) = 1$, while $\gamma(P_n) = \lceil \frac{n}{3} \rceil$. We now consider the problem of how large the cop number is as a function of n the number of vertices. There is a small but growing set of results on this topic, which will be considered in more detail in Chapter 3. Recall that we only consider connected graphs (otherwise, the upper bound is n). For many years, the best known upper bound was the one proved by Frankl, which shows that $c(G) = o(n)$.

Theorem 1.6 ([**89**]). *If G is a graph of order n, then*

$$c(G) \leq O\left(n \frac{\log \log n}{\log n}\right).$$

One of the goals of this section is to prove Theorem 1.6; we will learn a few new things along the way. Currently, the best known upper bound is due to Lu, Peng [**141**], which is

$$c(G) = O\left(\frac{n}{2^{(1-o(1))\sqrt{\log_2 n}}}\right).$$

The conjectured ceiling of the cop number is $O(n^{1/2})$. This is *Meyniel's conjecture*, and is one of the deepest problems on the cop number. Owing to its importance, we devote all of Chapter 3 to Meyniel's conjecture.

For a fixed integer $k \geq 1$, an induced subgraph H of G is *k-guardable* if, after finitely many moves, k cops can move only in the vertices of H in such a way that if the robber moves into H at round t, then he will be captured at round $t + 1$. For example, a clique or a closed neighbor set in a graph are 1-guardable, and G is $\gamma(G)$-guardable.

A path P in G is *isometric* if for all vertices u and v of P,

$$d_P(u, v) = d_G(u, v).$$

See Figure 1.8 for an example.

The following theorem of Aigner and Fromme on guarding isometric paths has found a number of applications.

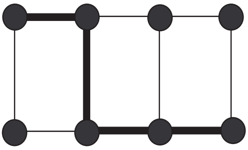


Figure 1.8. An isometric path, depicted in bold.

Theorem 1.7 ([**2**])**.** *An isometric path is* 1*-guardable.*

We may whimsically call an isometric path a *beat*: one cop can patrol the path effectively and ensure no robber can ever escape it without being captured.

Proof of Theorem 1.7. Let $P = \{v_0, v_1, \ldots, v_k\}$ be an isometric path in a graph G, and let

$$D_i = \{x \in V(G) : d(x, v_0) = i\}.$$

Since P is an isometric path, it follows that $v_i \in D_i$ for $i = 0, 1, \ldots, k$. See Figure 1.9.

The cop, restricted to P, plays as if the robber is on v_j when the robber is on some vertex of D_j, $j = 0, 1, \ldots, k-1$, and on v_k when $j \geq k$. We will call this the robber's *image.* If the robber is in D_j, then he can only move to vertices of D_{j-1}, D_j and D_{j+1}, so his image can only move from v_j to v_{j-1}, or v_{j+1}, or remain at v_j. Start the cop on v_0. As far as she and the image are concerned, they are

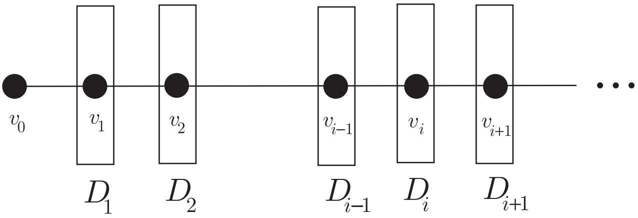


Figure 1.9. The sets D_i and their images on the path. Note that $D_0 = \{v_0\}$.

playing the game on a path, and in this game the cop wins. After the image has been caught, the actual robber can still move in G, but the robber's image moves to an adjacent vertex on P or is stationary. The cop now moves to recapture the image. Suppose that the robber tries to enter P after his image has been caught. Before his move onto P, he is in A_j for some j. If $j < k$, then his image is on v_j and so is the cop. The robber can only move to one of v_{j-1}, v_j or v_{j+1}. Whichever vertex he chooses, the cop will capture him on the next move. If $j \geq k$, then the image is on v_k and so is the cop. The robber's only possible moves are to v_k or v_{k-1} and the cop captures him on the next move. □

Our proof of Frankl's upper bound (inspired by the discussion of Lu and Peng [**141**]) makes use of the Moore bound, which is an important inequality involving the order n of graph, its maximum degree Δ, and its diameter. For simplicity, we will write $\mathrm{diam}(G) = D$. Note that if $\Delta = 2$, then it is an exercise that $n \leq 2D + 1$ (see Exercise 7).

Theorem 1.8. *Let G be a graph of order n, with maximum degree $\Delta > 2$ and diameter D. Then*

$$\begin{aligned} n &\leq 1 + \sum_{i=0}^{D-1} \Delta(\Delta - 1)^i \\ &= 1 + \Delta\left(\frac{(\Delta-1)^D - 1}{\Delta - 2}\right). \end{aligned} \tag{1.2}$$

Proof. For a fixed vertex u, and an integer $1 \leq i \leq D$, define $N_i(u)$ to be the set of vertices of distance i to u. (In particular, $N_1(u) = N(u)$.) It is evident that

$$\bigcup_{i=1}^{D} N_i(u) = V(G) \setminus \{u\}.$$

To finish the proof, we bound the cardinality of $N_i(u)$.

The vertex u contributes one to $|\bigcup_{i=1}^{D} N_i(u)|$, and $N_1(u)$ contributes at most Δ. A straightforward induction shows that for $2 \leq i \leq D$, $N_i(u)$ contributes at most $\Delta(\Delta-1)^{i-1}$ to $|N_i(u)|$. □

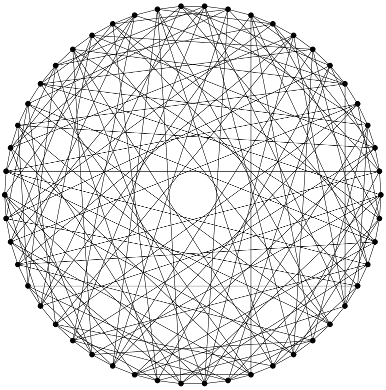


Figure 1.10. The Hoffman-Singleton graph.

The right-hand side of (1.2) is called the *Moore bound* and is named after Edward F. Moore. A graph whose order is equal to the Moore bound is called a *Moore graph*. The 5-cycle is a Moore graph for $D = 2$, while the Petersen graph is a Moore graph for $D = 3$. The *Hoffman-Singleton graph* is 7-regular, has $D = 2$, and girth 5; see Figure 1.10. Hoffman and Singleton [**115**] proved that Moore graphs exist for $D = 2$ and $\Delta = 2$, 3, 7 and possibly for 57, and they proved that C_7 is the unique Moore graph with $D = 3$. Darnell [**66**] proved there are no other Moore graphs for $D, \Delta \geq 3$, so $D = 2$, $\Delta = 57$ is the only open case! For more on the Moore bound and graphs, see the survey [**156**].

Proof of Theorem 1.6. Each closed neighbor set of a vertex u of maximum degree Δ is 1-guardable. By Theorem 1.7, an isometric path of length D is also 1-guardable. Asymptotically, the Moore bound becomes

$$n = O(\Delta^D).$$

By the Moore bound, both Δ and D cannot be less than

$$O\left(\frac{\log n}{\log \log n}\right),$$

(see Exercise 20). In particular, there is a subset X consisting of either a closed neighbor set or an isometric path of order at least

$$\frac{\log n}{\log \log n}$$

in G. Delete X to form the graph G''. Although graph G'' may be disconnected, the robber is confined to a connected component G' of this graph. The cops then move to G'. Then

$$c(G) \leq c(G') + 1, \tag{1.3}$$

since X is 1-guardable. Let $c(n)$ be the maximum of the cop numbers over connected graphs of order n. Now proceed by induction on n using (1.3) to derive that

$$\begin{aligned} c(n) &\leq c\left(\frac{n}{2}\right) + \frac{n/2}{\frac{\log n}{\log \log n}} \\ &= O\left(n \frac{\log \log n}{\log n}\right), \end{aligned}$$

where the equality follows by a straightforward induction. □

The *greedy approach* used above in the proof of Frankl's theorem will come up when we consider other upper bounds of $c(G)$ in Chapter 3.

1.5. Cops, Robbers, and Retracts

We close the chapter with a discussion of retracts, which play a vital role in understanding the game. Let H be an induced subgraph of G formed by deleting one vertex. We say that H is a *retract* of G if there is a homomorphism f from G onto H so that $f(x) = x$ for $x \in V(H)$; that is, f is the identity on H. The map f is called a *retraction* (or 1-*point retraction* or *fold*). Distances between pairs of vertices do not increase in the image (see Exercise 10). From the perspective of graph homomorphisms, retractions are idempotents (that is, satisfying $f^2 = f$) in the endomorphism monoid of G. For example, the subgraph formed by deleting an end-vertex is a retract.

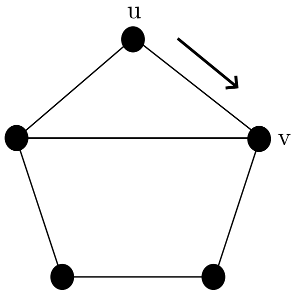


Figure 1.11. A retraction $u \to v$.

If u is a corner (that is, there is some vertex v such that $N[u] \subseteq N[v]$), then the mapping

$$f(x) = \begin{cases} v & \text{if } x = u, \\ x & \text{else,} \end{cases}$$

is a retraction (recall that our graphs are reflexive, so edges may map to a single vertex). In this case, we write $u \to v$. See Figure 1.11.

Retracts play an important role in characterizing cop-win graphs. The next theorem, due to Berarducci and Intrigila [**16**], shows that the cop number of a retract never increases.

Theorem 1.9 ([**16**]). *If H is a retract of G, then $c(H) \leq c(G)$.*

Theorem 1.9 fails if H is not a retract. For example, let G be a wheel W_n with $n \geq 4$ and universal vertex x, and let H be the subgraph isomorphic to C_n formed by deleting x.

Proof of Theorem 1.9. Suppose that k cops have a winning strategy in G, and let $f : G \to H$ be a retraction. We consider two parallel Cops and Robbers games: one played in G and one in H. The game in H may be considered as being played in G, since H is an induced subgraph of G. The strategy in G may not be sufficient alone to capture the robber in H (for example, the robber may need to leave H to be captured in G).

We therefore consider the following *shadow strategy*, which will come up several times in Chapter 2. Let the cops in G play as usual. In H, the cops play as the images of the cops in G. For simplicity, we label the images of the cops as $f(C)$. That is, if a cop C moves from vertex u to v, then a cop $f(C)$ moves from $f(u)$ to $f(v)$. These moves are possible as f is a homomorphism. We think of the $f(C)$ as shadowing the movements of the cops in H.

We claim the shadow strategy is a winning one for $f(C)$. Let the cops play in G with R restricted to H. Now suppose the cops are about to win in G. It must be that R and each of its neighbors v in H (as well as its neighbors in $G-H$) are joined to some cop. But then the edge RC becomes $Rf(C)$ under the retraction, and vC becomes $vf(C)$. Therefore, $N[R] \subseteq N[f(C)]$ in H, and the robber loses in the game played in H in the next round. Hence, $c(H) \leq k$, and the proof follows. □

We therefore have the following corollary.

Corollary 1.10. *If G is cop-win, then so is each retract of G.*

Corollary 1.10 hints at a recursive structure to cop-win graphs, as we will see in Chapter 2. If G is cop-win and has a retraction other than the identity map, then the retract is again cop-win. We now apply this idea repeatedly until we are left with an edge (or single vertex). Corollary 1.10 also gives us a simple sufficient condition for a graph to be robber-win. For example, if a graph retracts to a cycle, then it is robber-win.

In the following theorem and exercises, we now refer to more general retractions (with corresponding retracts which are their images) which are a a composition of a set of (1-point) retractions we have considered so far. (Indeed, retractions usually refer to these more general kind of mappings.) The following theorem gives us an upper bound for the cop number of a graph using retracts.

Theorem 1.11. [**16**] *If H is a retract of G, then*

$$c(G) \leq \max\{c(H), c(G-H)+1\}.$$

Proof. Define $m+1 = \max\{c(H), c(G-H)+1\}$. Let $f : G \to H$ be a retraction. We describe a winning strategy for $m+1$ cops in G. First, as $m+1 \geq c(H)$, the cops play by using the cops' strategy in H and capture the robber's image $f(R)$ in H. These moves are moves in G as f is a homomorphism. Now if the robber is in H, $R = f(R)$ and the cops win.

Otherwise, the robber is in $G - H$ and the cops do the following. One cop protects H by occupying in all rounds the image $f(R)$. Hence, H becomes 1-guardable, and so the robber must remain in $G - H$ to survive. As $c(G - H)$ cops can win in any connected component of $G - H$, the proof follows. □

If P is an isometric path with vertices $\{u_0, u_1, \ldots, u_n\}$, then it is straightforward to show that the following map $f : G \to P$ is a retraction:

$$f(v) = \begin{cases} u_i & \text{if } d(v, u_0) = i \text{ and } i \leq n, \\ u_n & \text{otherwise.} \end{cases} \tag{1.4}$$

We note that f describes the moves of one cop used to guard an isometric path as in proof of Theorem 1.7.

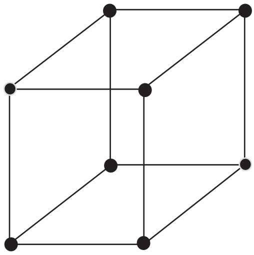
Exercises

1. Prove the following facts about binomial coefficients, where n is a positive integer and $0 \leq i \leq n$.
   (a) $\sum\limits_{i=0}^{n} \binom{n}{i} = 2^n$.
   (b) $\frac{4^n}{2n} \leq \binom{2n}{n}$.
   (c) The inequalities in (1.1).
2. Prove Lemma 1.1.
3. [**167**] Show that a cop-win regular graph is a clique.
4. (a) Prove that each finite tree with at least two vertices has at least two end-vertices.
   (b) A graph is *chordal* if each of its cycles of four or more vertices has a *chord*: an edge joining two vertices that are not adjacent in the cycle. A *simplicial vertex* has its neighbor set a clique. Prove that a chordal graph has at least two simplicial vertices.

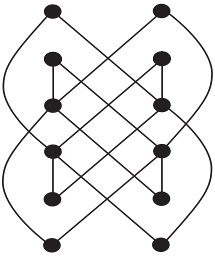
(c) Prove that the deletion of a simplicial vertex from a chordal graph results in another chordal graph.

5. Find the cop number of the following graphs.

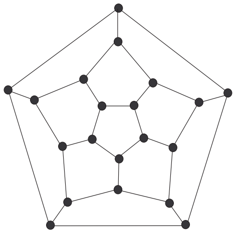
(a) The *cube* Q_3.



(b) The following graph.



(c) The *dodecahedron* depicted below.



6. (a) Prove that the Petersen graph is vertex- and edge-transitive.
   (b) Consider the graph G formed from the dodecahedron, whose vertex set consists of the pairs of vertices $\{x, y\}$, where we have that $d(x, y) = 5$, and pairs are adjacent if and only if there is a perfect matching between them. Show that G is isomorphic to the Petersen graph.
7. Derive the Moore bound in the case the maximum degree is 2:
$$n \leq 2D + 1.$$
8. Find a graph with girth 4 with $c(G) < \delta(G)$.
9. Prove that a tree (infinite or otherwise) either contains a ray or an end-vertex.
10. Prove that for all $x, y \in V(G)$, if $f : G \to H$ is a homomorphism, then $d_G(x, y) \geq d_H(f(x), f(y))$.
11. A vertex in a digraph is a *source* if it has in-degree 0. Show that every tournament with a source is cop-win.
12. For each positive integer k, give examples of weakly connected digraphs with cop number k.
13. [**164**] In the *active version* of Cops and Robbers, at least one cop and the robber must move on their respective turns. Our definition of Cops and Robbers and the active version coincide on reflexive graphs, but differ on irreflexive ones. Define $c'(G)$ to be the cop number in the active version of the game. Prove that for an irreflexive graph G,
$$c(G) - 1 \leq c'(G) \leq c(G).$$
14. [**34**] Make the following slight change to the rules of Cops and Robbers: To capture the robber, the cop must move along an edge to the vertex holding the robber. If there are loops at each vertex, then this is identical to the original game. If there are no loops, then this is the active version of the previous exercise. Let G be a graph with loops at some (but possibly not all) vertices. Show that G is cop-win if and only if G is dismantlable.
15. Show that the bipartite graph $G(P)$ using projective planes from the proof of Theorem 1.5 satisfies $c(G(P)) \leq q + 1$.

16. [**56**] Suppose G is a graph with an induced cycle of length at least 4, where at least one vertex of the cycle has degree 2. Then prove that G is not cop-win.

17. Assume that one cop and the robber play *optimally* on a tree T; that is, the cop is trying to catch the robber in the minimum number of rounds possible, and the robber is trying to maximize the number of rounds. Show that the cop captures the robber in at most $\left\lfloor \frac{|V(G)|}{2} \right\rfloor$ rounds.

18. [**56**] Show that if G is cop-win, then the subdivision of any edge in a cycle yields a graph with cop number two. (*Hint*: Use Exercise 16.)

19. Let G have $\Delta \leq 3$ and suppose any two adjacent edges are contained in a cycle of length at most 5. Then $c(G) \leq 3$.

20. Show that in a graph G, both Δ and D cannot be less than
$$(1+o(1))\frac{\log n}{\log\log n}.$$

21. Show that if H is a retract of G, then H is an *isometric subgraph* of G; that is, for all x, y in $V(H)$,
$$d_H(x,y) = d_G(x,y).$$

22. [**16**] Fix $k > 0$ an integer. Prove that if H is a retract of G and $c(G - H) \leq k$, then $c(H) \leq k$ if and only if $c(G) \leq k$.

23. For each infinite cardinal κ, give examples of 2^κ many non-isomorphic graphs with cop number κ.

24. Define the *isometric path number* or *precinct number* of G, written $p(G)$, as the minimum number of isometric paths (or *beats*) needed to cover G. Note that $c(G) \leq p(G)$.

    (a) Prove that
    $$p(G) \geq \left\lceil \frac{n}{\operatorname{diam}(G)+1} \right\rceil.$$

    (b) If G is a tree with L leaves, then prove that
    $$p(G) = \left\lceil \frac{L}{2} \right\rceil.$$

    (*Hint*: Use induction on the number of vertices of G.)

25. (a) [**16**] Show that subdividing all edges of a graph an even number of times does not decrease the cop number.
    (b) For a fixed integer $k > 2$, show that the class of all graphs with girth at least k is cop-unbounded.

26. The Hoffman-Singleton graph H is the unique Moore graph which is 7-regular. See Figure 1.10. Determine the cop number of H.

27. In this exercise, we consider Cops and Robbers played on finite orders. This version of the game was introduced in Hill's doctoral thesis [**113**]. For an order G, the cops are initially placed on minimal vertices, with the robber placed on a maximal vertex. A cop on u can move to v with $u \le v$ provided there is no x such that $u \le x \le v$. A move of the robber is defined dually. The cops win as in the usual game, and the robber wins if he reaches a minimal vertex or there is no cop below him. The *cop number* of G, written $c(G)$, is defined in the usual fashion.
    (a) Show that cop number of the order in Figure 1.3 is two.
    (b) [**113**] Vertices u and v are *incomparable* if neither $u \le v$ nor $v \le u$. Suppose that G is an order with the property that all maximal paths connecting a minimal and maximal vertex have the same length. Prove that $c(G)$ is at most the maximum cardinality of a set of incomparable vertices. (*Hint*: Apply Dilworth's theorem (see [**69**] or Theorem 8.4.33 of [**197**]).)
    (c) The *Boolean lattice* of order n, written $B(n)$, has vertices subsets of $[n]$, with $u \le v$ if u is a subset of v. The order in Figure 1.3 is $B(3)$. Calculate $c(B(n))$, where $2 \le n \le 4$.
    (d) [**113**] Show that if $n = 2m$ is even, then

$$2^m \le c(B(n)) \le \binom{n}{m}.$$

Chapter 2

Characterizations

2.1. Introduction

Some graph classes have beautiful characterizations. For example, bipartite graphs are those with no odd cycles as induced subgraphs, Eulerian graphs are those with each degree even, and trees are connected graphs with size one less than their order. Such characterizations, when available, are highly prized: they give us insight into the structure graphs in the class and can help enormously when probing their structural and algorithmic properties. Proving characterizations for some classes may take many years or be highly nontrivial (or both). A famous example of this is the recent characterization of perfect graphs; see [**48**].

Since their introduction, the structure of cop-win graphs has been well understood. Nowakowski and Winkler [**167**] and, independently, Quilliot [**169**] introduced a kind of ordering of the vertex set—now called a cop-win or elimination ordering—which completely characterizes such graphs. As we will demonstrate in Section 2.2, cop-win graphs in the finite case are exactly the dismantlable ones; that is, graphs resulting in a single vertex after a finite number of deletions of corners (see Chapter 1). As such, cop-win graphs have a certain linear or tree-like structure which makes them at once simple to recognize and simple to analyze.

For over 25 years, a characterization of graphs with cop number two or higher has remained elusive. Despite this fact, as proved by [**16**], [**100**], and [**105**], the k-cop-win graphs can be recognized by polynomial time algorithms; see Chapter 5. Although this gave an algorithmic characterization of such graphs, it did not immediately imply an explicit structural characterization. We therefore had a somewhat strange situation: for instance, there were polynomial time algorithms to recognize 2-cop-win graphs, but no explicit characterization of them. In 2009, Clarke and MacGillivray [**57**] found an explicit structural characterization of k-cop-win graphs for all $k > 1$. The characterization exploits a linear structure, not of the graph, but of a certain power of the graph (where powers are taken with respect to categorical products). The characterization is appealing as it generalizes that of cop-win graphs. We present a full discussion with proofs of this exciting new direction in the study of k-cop-win graphs in Section 2.3.

2.2. Characterizing Cop-win Graphs

The game of Cops and Robbers historically first considered only the case of one cop, and that is our focus in the present section. Recall from Chapter 1 that a graph G is *cop-win* (or *copwin*) if one cop has a winning strategy to capture the robber. That is, $c(G) = 1$. For example, a tree is cop-win (see Lemma 1.2), as is a clique. As we will see below, the cop-win case possesses a beautiful structural characterization, which remains one of the crown jewels of the field.

Consider the following graph, which is cop-win but less evidently so; see Figure 2.1. The reader should take a few minutes to consider a winning strategy for the cops played on this graph. Vertex u has a special property: if the robber moved here and if the cop moved to vertex v, then she can anticipate all the moves of R and win in the next round. Such a vertex is called a *corner* (or a *trap*, *pitfall*, or *irreducible*). More precisely, a vertex u is a *corner* if there is some vertex v such that $N[u] \subseteq N[v]$. The vertex v is said to cover u. We write $u \to v$, and say that v *dominates* (or *covers*) u.

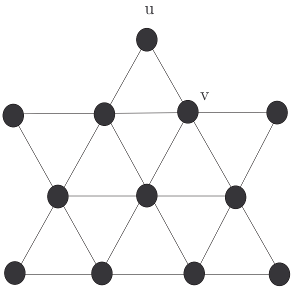


Figure 2.1. A cop-win graph with corner u.

As noted in Chapter 1, if $u \to v$, then the mapping $f : G \to G - u$ defined by

$$f(x) = \begin{cases} v & \text{if } x = u, \\ x & \text{else} \end{cases}$$

is a *retraction* (sometimes called a 1-*point retraction* or *fold*). By Theorem 1.9, the graph $G - u$ is once again cop-win.

Corners are clearly useless for the robber. But cop-win graphs always contain corners!

Lemma 2.1 ([**167**]). *If G is a cop-win graph, then G contains at least one corner.*

Proof. Consider the second to last move of the cop. The robber could pass, so C must be joined to R. Or the robber could move to a neighboring vertex, so C is joined to each neighbor of R. See Figure 2.2. Hence, $R \to C$. □

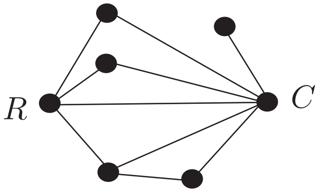


Figure 2.2. The second-to-last move of the cop.

A graph is *dismantlable* if some sequence of deleting corners results in the graph K_1. For example, each tree is dismantlable: delete end-vertices repeatedly (a tree always has at least two end-vertices) until a single vertex remains. The same approach holds with chordal graphs, which always contains at least two simplicial vertices (that is, vertices whose neighbor sets are cliques). No chordless cycle is dismantlable, as the reader can check directly.

With the help of retracts, we can now prove the following important theorem.

Theorem 2.2 ([**167**]). *If G is cop-win, then G is dismantlable.*

Proof. The proof follows by induction on the order of G. The base case is trivial as $G \cong K_1$. By Lemma 2.1, each cop-win graph contains a corner u dominated by some vertex v. Now form $G - u$, and note as above that $G - u$ is a retract of G. By Theorem 1.9, $G - u$ is cop-win. As $G - u$ is dismantlable by the induction hypothesis, the proof follows. □

The perhaps surprising fact is the converse of Theorem 2.2 is true. The main theorem of this section is the following, which serves to characterize the cop-win graphs. The theorem was proved by Nowakowski and Winkler [**167**] and, independently, by Quilliot [**169**] in his doctoral thesis.

Theorem 2.3 ([**167**]). *A graph is cop-win if and only if it is dismantlable.*

Proof. The forward direction is just Theorem 2.2. We next suppose that G is dismantlable. We use induction on the order of G to show G is cop-win. There is nothing to prove in the base case, as $G \cong K_1$. Now suppose that G is dismantlable of order $n + 1$, where $n \geq 1$ is fixed. Then G contains some corner u, with $u \to v$ for some vertex v, so that $G - u$ is dismantlable.

By the induction hypothesis, $G - u$ is cop-win as it has order n. We use this fact to show that G is cop-win. We use the shadow strategy from the proof of Theorem 1.9 in Chapter 1. In particular, the cop plays in $G - u$ using his winning strategy on there, but so that whenever R moves to u, then C moves as though R moves to v. This is possible as $u \to v$. We think of the vertices of $G - u$ as images under the retraction f which maps u to v and fixes all other vertices. Now C eventually captures the image of the robber $f(R)$ with his winning strategy on $G - u$. Now either $R = f(R)$ in which case the robber is captured, or R is on u with C on v. But then in the latter case, the cop wins in the next round. □

Cop-win (or sometimes called dismantlable) graphs have a recursive structure, which can be made explicit in the following sense. Observe that a graph is dismantlable if we can label the vertices by positive integers $[n]$ in such a way that for each $i < n$, the vertex i is a corner in the subgraph induced by $\{i, i+1, \ldots, n\}$. We call this ordering of $V(G)$ a *cop-win ordering*. See Figure 2.3 for a graph with vertices labeled by a cop-win ordering. Cop-win orderings are sometimes called *elimination orderings*, as we delete the vertices from lower to higher index until only vertex n remains. Cop-win orderings are usually not unique. The reader should produce at least one more cop-win ordering of the graph in Figure 2.3.

Cop-win orderings suggest a kind of linear structure to cop-win graphs; it roughly suggests that by "sweeping" from largest index vertex to smallest in the ordering, we may capture the robber. This intuition is made precise by the following winning strategy for the cops—first made explicit by Clarke and Nowakowski [**59**]—in a cop-win graph exploiting the cop-win ordering.

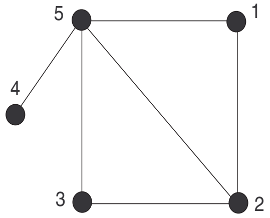


Figure 2.3. A cop-win ordering of a cop-win graph.

Cop-win (or No-backtrack) Strategy ([**59**]). Assume that $[n]$ is a cop-win ordering of G, and for $1 \le i \le n$ define

$$G_i = G \restriction \{n, n-1, \ldots, i\}.$$

Note that $G_1 = G$ and G_n is just the vertex n. For each $1 \le i \le n-1$, let $f_i : G_i \to G_{i+1}$ be the retraction map from G_i to G_{i+1} mapping i onto a vertex that covers i in G_i. Define F_1 to be the identity mapping on $G : F_1(x) = x$ for all $x \in V(G)$. For $2 \le i \le n$ define

$$F_i = f_{i-1} \circ \cdots \circ f_2 \circ f_1.$$

In other words, the F_i is the mapping formed by iteratively retracting corners $1, 2, \ldots, i-1$. As the f_i are homomorphisms, so are the F_i (recall that all our graphs are reflexive). Further, for all i, as the f_i are retractions, $F_i(x)$ and $F_{i+1}(x)$ are either equal or joined. If the robber is on vertex x in G, then we think of $F_i(x)$ as the robber's *shadow* on G_i; see Figure 2.4.

With this terminology in hand, we now describe the Cop-win Strategy. The cop begins on G_n (the vertex n), which is the shadow of the robber's position under F_n (note that everything in G maps to n under F_n). Suppose that the robber is on u and the cop is occupying the shadow of the robber in G_i equaling $F_i(u)$. If the robber moves to v, then the cop moves onto the image $F_{i-1}(v)$ of R in the larger graph G_{i-1}.

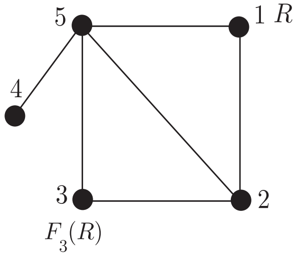


Figure 2.4. The robber and his shadow $F_3(R) = f_2 \circ f_1(R)$.

Theorem 2.4 ([**59**]). *The Cop-win Strategy results in a capture for the cop in at most n moves.*

Proof. The proof follows by induction on n, with the base case being trivial. Suppose now that for some $i \leq n$, the cop has captured $F_i(u)$, where u is the present position of the robber, and it is the robber's turn to move. Suppose the robber moves to v. We must show that $F_i(u)$ is joined or equal to $F_{i-1}(v)$, showing that the cop can move to capture the image of R in G_{i-1}.

If $F_i(u) = F_{i-1}(u)$, then $F_{i-1}(v)$ is joined or equal to $F_i(u)$. In the other case, we must have that $F_i(u)$ is joined and not equal to $F_{i-1}(u)$. But then $F_{i-1}(u)$ is the corner removed from G_{i-1} to obtain G_i, from which it follows that

$$N[F_{i-1}(u)] \subseteq N[F_i(u)],$$

and so $F_{i-1}(v)$ is joined or equal to $F_i(u)$. In each step of the induction, the robber's image may be caught with one move. Hence, the robber's actual position will coincide with his image in at most n moves of the cop. □

Theorem 2.4 reveals two important features of cop-win graphs. First, under the assumption that the cop is trying to minimize the number of rounds needed to win, the cop never needs more than n moves to win. This is not obvious a priori. Further, when playing the Cop-win Strategy, the robber can never get "behind" the cop, in the following sense. If the cop is occupying $F_i(R)$, and the robber moves to G_{i-1}, then he is immediately captured by the cop. This follows since F_i is the identity on G_i. For this reason, the Cop-win Strategy is also called the *No-backtrack Strategy.*

The Cop-win Strategy may not be the fastest strategy for the cop, in general. For example, consider a path P_n with n vertices labeled from left to right by $1, 2, \ldots, n$. Consider the cop-win ordering simply $1, 2, \ldots, n$. Using the Cop-win Strategy, the cop requires $n - 1$ moves to catch the robber (whose best move is to start at 1 and simply pass in each round). However, it is evident that the cop can win on P_n in at most $\lfloor \frac{n}{2} \rfloor$ moves by occupying a vertex in the centre of the path. (A similar strategy optimizes play for a robber on a tree.) See

Chapter 8 for more on estimates on the number of rounds needed for cop-win graphs when both players play *optimal strategies* (that is, the cop is trying to make the game as short as possible, while the robber is avoiding capture as long as possible).

Theorem 2.5. *In a cop-win graph G of order $n \geq 5$, the cop can capture the robber in at most $n-3$ rounds.*

Proof. The proof is by induction on n, with the case for $n=5$ following by checking all the cop-win graphs with five vertices. Suppose that the conclusion holds for all graphs of order $n \geq 5$, and let G have $n+1$ vertices. Hence, G contains a corner u dominated by v and, since $G-u$ is a retract of G, it is cop-win.

By the induction hypothesis, the cop can capture the robber in $G-u$ in at most $n-3$ rounds. The cop plays her winning strategy on $G-u$, and captures the shadow of R. So if R is on u, then C plays as if he were on v. After $n-3$ rounds, the robber is caught, or R is on u and C is on v. Hence, the robber is caught in at most $(n+1)-3$ moves in G. □

In [**97**] and [**98**], the bound of $n-3$ in Theorem 2.5 was improved to $n-4$ for graphs with order at least 7, and shown to be best possible. Further, all graphs which realize the "capture time" $n-4$ are characterized in [**98**]. See Section 8.6 of Chapter 8 for more on the capture time parameter.

It is important to note that the dismantling characterization of Theorem 2.3 fails badly for infinite graphs. For example, a ray is dismantlable (if we allow infinitely many vertex deletions), but fails to be cop-win. There is a characterization of cop-win graphs of any order (finite or infinite) that we include here, owing to [**167**]. It will be important in Section 2.3 when we discuss the characterization of k-cop-win graphs, where $k>1$. For that reason, we urge the reader to study it carefully before reading that section. Although it appears more opaque on first viewing than the characterization using cop-win orderings, after some reflection it is quite natural.

We define a relation $\preceq$ on vertices.The relation is defined recursively on ordinals, with $x \leq_0 x$ for all vertices u (in other words, $\leq_0$

is just the *diagonal* or *equality relation* on $V(G)$). We will make our definitions so that $u \leq_\alpha v$ will mean that when a robber is on vertex u, a cop is on vertex v, and it is the robber's turn to move, the robber will lose in at most α rounds. For an ordinal α, define $u \leq_\alpha v$ if and only if for each $a \in N[u]$ there exists a $b \in N[v]$ such that $a \leq_\beta b$ for some $\beta < \alpha$. Let ρ be the least ordinal such that $\leq_\rho = \leq_{\rho+1}$ and define $\preceq = \leq_\rho$.

Note that if $\rho < \alpha$, then the relation $\leq_\rho$ is a subset of $\leq_\alpha$. As such relations are bounded above in cardinality, the ordinal ρ exists. More precisely,

$$\rho \leq |V(G)|(|V(G)| - 1) \tag{2.1}$$

if G is finite, and

$$\rho \leq |V(G)| \tag{2.2}$$

in the infinite case by elementary set theory. In particular, ρ is finite if $|V(G)|$ is finite. As we will see in the proof of Theorem 2.6, an intuitive view of ρ is that it is equal to the maximum number of rounds needed for the cop to capture the robber (assuming she is playing optimally; that is, minimizing the number of rounds in the game). As an exercise (see Exercise 18), the reader should verify that $\preceq_\alpha$ is a *quasi-order* (that is, a reflexive and transitive relation) for each ordinal α. A binary relation on a set X is *trivial* if it equals the Cartesian product of X with itself, $X \times X$.

Theorem 2.6 ([**167**]). *A graph G is cop-win if and only if the relation $\preceq$ on $V(G)$ is trivial.*

Proof. For the forward direction, we prove the contrapositive and assume that for some x_0 and y_0 in $V(G)$, $y_0 \not\preceq x_0$. As G is connected, assume without loss of generality that the cop begins at x_0 and the robber begins at y_0. If a graph is cop-win, then the cop may choose any vertex to begin the game, at the cost of a finite number of additional moves. By the definition of $\preceq$, the robber can move to $y_1 \in N[y_0]$ such that for all $x \in N[x_0]$, $y_1 \not\preceq x$. If no such y_1 exists, then $y_0 \leq_{\rho+1} x_0$. By induction, the robber guarantees that for all i there is a vertex $y_{i+1} \in N[y_i]$ such that for all $x \in N[x_i]$, $y_{i+1} \not\preceq x$.

In particular, there is always a vertex joined to R and not to the cop, ensuring the robber can escape capture for one more round.

For the reverse direction, suppose that the relation $\preceq$ is trivial. We show that G is cop-win. In the first round, the cop chooses an arbitrary vertex x_0. Say that R chooses y_1. As $y_1 \leq x_0$, there is some $x_1 \in N[x_0]$ and some $\rho_1 < \rho$ satisfying $y_1 \leq_{\rho_1} x_1$. Suppose that after the ith move the cop is at x_i and the robber is at y_i so that $y_i \leq_{\rho_i} x_i$. When the robber moves to some y_{i+1} in $N[y_i]$, the cop moves to x_{i+1} in $N[x_i]$ with the property that for some $\rho_{i+1} < \rho_i$,

$$y_{i+1} \leq_{\rho_{i+1}} x_{i+1}.$$

As the ordinals are well-ordered, they do not contain infinite descending chains. Therefore, we must have that $\rho_i = 0$ for some finite index i. But then $x_i = y_i$ (as $\leq_0$ is just the equality relation) and the cop has captured the robber. □

As pointed out in [**167**], a winning strategy for the cop is implicit in the proof of the reverse direction of Theorem 2.6. In round 0, the cop moves to an arbitrary vertex x_0. In all later rounds, the cop moves so that his position and that of the robber belong to a sequence of relations indexed by a strictly decreasing sequence of ordinals: when the robber moves to u, the cop moves to a vertex $v \in N[y]$ such that $u \leq_\beta v$ for some $\beta < \rho$.

While the characterization in Theorem 2.6 is useful for an algorithmic perspective on finite cop-win graphs, it is less revealing of the structure of infinite cop-win graphs. Infinite cop-win graphs have a much more complex structure, as illustrated in Chapter 7. A graph G is *vertex-transitive* if for each pair of vertices x and y there is an automorphism of G mapping x to y. The Petersen graph, cliques, and cycles are all vertex-transitive. As an indicator of the complexity of infinite cop-win graphs, we show in Theorem 7.15 that for each infinite cardinal κ, there are families of 2^κ-many non-isomorphic cop-win graphs that are vertex-transitive. In contrast, in the finite case, a vertex-transitive cop-win graph must be a clique!

We mention in passing that there is a small literature [**88, 106**] on Cops and Robbers on Cayley graphs, which are a special class of vertex-transitive graphs. We do not survey those results here.

2.3. Characterizing Graphs with Higher Cop Number

Although k-cop-win graphs for $k > 1$ were defined over 25 years ago by Aigner and Fromme [**2**], a structural characterization of such graphs has remained elusive. In 2009, Clarke and MacGillivray [**57**] announced that such a characterization does indeed exist. In fact, they gave both a relational and elimination-ordering characterization akin to the relational characterization of Theorem 2.6 and cop-win orderings, respectively. We present both characterizations here with full proofs. As their approach relies heavily on graph products, we begin with a brief interlude on categorical and strong products of graphs.

2.3.1. The Categorical and Strong Product of Graphs. Graph products are a powerful tool in graph theory and are concerned with taking two (or more) graphs and generating new ones. Products often lead to more complex graphs, and yet they are good instruments for probing the structure of graphs. While several graph products can be defined, in this subsection we focus on two particular products: the categorical and strong products. See the book [**119**] for a thorough discussion of these and other graph products. The definitions of both products coincide in the case of reflexive graphs, which is the case for our discussion in this book. More details on Cops and Robbers played on graph products will be discussed in Chapter 4. We discuss both products separately, since Cops and Robbers is sometimes played on irreflexive graphs.

Let G and H be graphs. The *categorical* (or *direct*) *product* of graphs G and H, written $G \times H$, has vertex set $V(G) \times V(H)$. Vertices (a, b) and (c, d) are joined if $ac \in E(G)$ and $bd \in E(H)$. Note that if we consider the product of irreflexive graphs, then $K_2 \times K_2 \cong K_2 + K_2$, which justifies the notation of the product; see Figure 2.5 for an example. Note that the categorical product is commutative, up to isomorphism:

$$G \times H \cong H \times G.$$

The graphs G and H are called *factors* of the product. The categorical product may be naturally generalized to the product of a finite

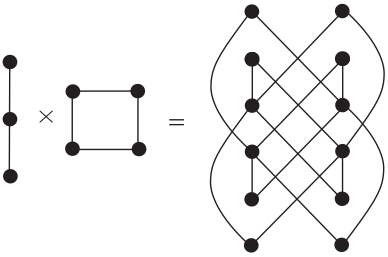


Figure 2.5. The categorical product $P_3 \times C_4$.

number of graphs $(G_i : 1 \leq i \leq n)$ by induction, written

$$G_1 \times G_2 \times \cdots \times G_n$$

or $\prod_{i=1}^n G_i$, with vertex set $V(G_1) \times V(G_2) \times \cdots \times V(G_n)$, and n-tuples are joined if they are joined in each factor. In particular, the *projection maps*

$$\pi_j : \prod_{j=1}^n G_j \to G_j$$

defined by $\pi_j(u_1, u_2, \ldots, u_n) = u_j$ are homomorphisms. We leave it as an exercise (see Exercise 11) to show that the categorical product is associative:

$$(G \times H) \times J \cong G \times (H \times J).$$

The categorical product of two connected irreflexive graphs need not be connected, as illustrated by $K_2 \times K_2$. In fact, if G and H are irreflexive, then $G \times H$ is connected if and only if both G and H are connected and at least one of them is non-bipartite (see Exercise 12).

The *strong product* of G and H, written $G \boxtimes H$, has vertex set $V(G) \times V(H)$. Vertices (a, b) and (c, d) are joined if $a = c$ and $bd \in E(H)$, or $ac \in E(G)$ and $b = d$, or $ac \in E(G)$ and $bd \in E(H)$. In other words, each coordinate is either joined or equal in each factor. Note that for reflexive graphs, $G \boxtimes H = G \times H$. Observe that $K_2 \boxtimes K_2 \cong K_4$ which justifies the notation for the product. Like the categorical product, the strong product is commutative. However, K_1 acts as a *unit*:

$$K_1 \boxtimes G \cong G \boxtimes K_1 \cong G.$$

Projections are defined analogously as in the categorical product, and are also homomorphisms if all the factors are reflexive.

The strong product may be generalized to the product of a finite number of graphs $(G_i : 1 \leq i \leq n)$ in the natural way, written

$$G_1 \boxtimes G_2 \boxtimes \cdots \boxtimes G_n$$

or $\boxtimes_{i=1}^{n} G_i$, with vertex set $V(G_1) \times V(G_2) \times \cdots \times V(G_n)$, and n-tuples joined if they are joined or equal in each factor. The strong product is associative (Exercise 11).

To further illustrate these two products, we give bounds on their cop numbers. Both Theorem 2.7 and 2.8 were first proved in [**164**]. We give some notation which will help the discussion. For a fixed $x \in V(H)$, define $G.\{x\}$ to be the subgraph of either product induced by $\{(y, x) : y \in V(G)\}$. The graph $\{y\}.H$ is defined analogously. Note that in the case where G and H are irreflexive, then $G.\{x\} \cong G$ and $\{y\}.H \cong H$ in the strong product, while both subgraphs are co-cliques in the categorical product. For simplicity, in both cases, we refer to $G.\{x\}$ and $\{y\}.H$ as *copies* of G and H, respectively. Sometimes $G.\{x\}$ and $\{y\}.H$ are called G- or H-*layers*, respectively.

A player *moves on* G if the projection onto H is constant; moving on H is defined analogously. The *cops capture a projection of* (u, v) *onto* G (respectively, *onto* H) if they capture (u, x) for some $x \in V(H)$ (respectively, (x, v) for some $x \in V(G)$). A cop *shadows the robber on* G (respectively, *on* H) if after each round the cop's projection onto G (respectively H) equals the robber's projection.

Theorem 2.7 ([**164**]). *Let G and H be connected, non-bipartite irreflexive graphs (so $G \times H$ is connected), with $c(H) \geq c(G)$.*

(1) *If G and H are both cop-win, then*

$$c(G \times H) \leq 3.$$

(2) *If $c(H) > 1$, then*

$$c(G \times H) \leq 2c(G) + c(H) - 1.$$

Proof. For item (1), let two cops, say C_1 and C_2, each capture the projection of the robber onto G with one cop capturing $G.\{x\}$ and the other on $G.\{y\}$ with x joined to y in H. To accomplish this, we start by placing the cops on (v, x) and (v, y) for some vertex v in G. As we are working in the categorical product, a cop's move must either change both coordinates or neither coordinate. The cops each use the winning strategy for G in the first coordinate. When the strategy calls for a pass, they also pass in the second coordinate. When the strategy calls for a move, they also move in the second coordinate, sliding between x and y. Next, these two cops will use the winning strategy for H, all the while shadowing the robber in G. To accomplish this, if the robber passes, then they must also pass in order to maintain their shadowing. Thus, the third cop's role is to keep the robber moving (that is, the robber cannot pass indefinitely). It is clear that one cop can accomplish this task. When the robber moves, the cops shadow the robber in the first coordinate and use C_1's winning strategy for H in the second coordinate. If the strategy calls for a move, then C_1 makes this move with C_2 trailing behind at distance one. If the strategy calls for a pass, then they must still make some move, so in this case C_1 and C_2 simply swap places and exchange roles. Using this strategy, eventually the robber is captured.

For item (2), let

$$n = 2c(G) + c(H) - 1.$$

As in item (1), let the n-many cops play in two joined copies of H, say $\{x\}.H$ and $\{y\}.H$, with $xy \in E(G)$. Suppose that the robber occupies (a, b). Let a set $P = \{C_1, C_2, \ldots, C_{2c(G)}\}$ of cops each capture b. In more detail, we build the set P by adding one cop at a time. First, $c(H)$ cops use their winning strategy to capture the robber. Any one cop that captures the robber is added to P. The process is repeated with any remaining set of $c(H)$ cops, which gives us a second cop for P. We repeat this process until $|P| = 2c(G)$. Each C_i in P shadows the robber on H. Let one of the remaining $c(H)-1$ cops, say C_k, force the robber to move. Since G has an odd cycle and owing to C_k, the cops in P move so that $c(G)$-many of them occupy $\{x\}.H$ and $c(G)$-many of them occupy $\{y\}.H$. (The role of the joined vertices x and y is analogous to the situation in the proof of item Theorem 2.7 (1).)

For each cop C_i on $\{x\}.H$, match them with a cop C_i' on $\{y\}.H$. In other words, if C_i moves from (x, u) to (y, v), then C_i' moves to (x, v). Define U_x to be the set of cops in P occupying $\{x\}.H$, and U_y

to be the set of cops in P occupying $\{y\}.H$. The cops in U_x and U_y shadow the robber on H which prevents the robber from entering the same copy of H as the cops in P.

Now the cops in U_x play their winning strategy in G if the robber moves (recall that C_k forces the robber to eventually move!). When the strategy calls for a cop C_i to move, her partner C_i' follows, trailing at distance one. When the strategy calls for C_i to pass, C_i and C_i' exchange places and swap names. Hence, the robber must either eventually move to a copy of H joined to one containing a cop C_i in U_x (and so is captured by C_i) or to a copy of H containing a cop $C_i \in U_x$ (so the robber is then captured by $C_i' \in U_y$). □

We now discuss the cop number of the strong product.

Theorem 2.8. (1) [**167**] *A strong product of cop-win graphs is cop-win.*

(2) [**164**] *Let G and H be graphs with $c(G) \geq 2$ or $c(H) \geq 2$. Then*

$$c(G \boxtimes H) \leq c(G) + c(H) - 1.$$

(3) [**27**] *If k is a positive integer, $c(G) = k > 1$, and H is cop-win, then*

$$c(G \boxtimes H) = k.$$

Proof. For item (1), the projections of the cops capture the robber in each projection, and then shadow the robber in each factor. After a finite number of rounds, each projection of the robber is captured, and the proof of item (1) follows.

For item (2), without loss of generality, suppose that $c(H) \geq 2$. Let a set

$$X = \{C_1, C_2, \ldots, C_{c(H)}\}$$

of cops capture the projection of the robber onto G. The reader should verify that $c(G) + c(H) - 1$ cops are sufficient for this. The cops in X then shadow the robber on G and in parallel, play their winning strategy on H.

For item (3), note that the upper bound follows by item (2). For the lower bound, suppose that $j(< k)$ cops can win on $G \boxtimes H$. If C is a cop on vertex (x, z), then write C' for its projection onto the vertex (x, y) in $G.\{y\}$. The strategy of R is to remain in $G.\{y\}$ and to use his strategy in G to avoid capture by C': whenever a cop moves to (x, z), the robber moves as if the cop were on (x, y). As $c(G) = k$, the robber can always avoid capture by the cops' projection in $G.\{y\}$. If the cops can win in $G \boxtimes H$ with the proposed robber's strategy, then consider the second-to-last move of the cops. Then $N[R]$ is contained in the union of the sets $N[C]$ for the cops C. Hence, $N[R] \cap V(G.\{y\})$ is contained in the union of the sets $N[C']$, and so we derive the contradiction that the cops win in $G.\{y\}$ in the next round. □

A class of graphs $\mathcal{C}$ is called a *variety* if it is closed under strong products and taking retracts. That is, if $G, H \in \mathcal{C}$, then $G \boxtimes H \in \mathcal{C}$, and if $H \leq G$ is a retract of $G \in \mathcal{C}$, then $H \in \mathcal{C}$. Theorem 2.8 (1) and Theorem 1.9 establish that the class of cop-win graphs is a variety, a fact first noted in [**167**].

2.3.2. A Relational Characterization of k-cop-win Graphs. We now give a characterization of k-cop-win graphs from [**57**], which is closely related to the relational characterization of cop-win graphs from [**167**] presented above as Theorem 2.6. Throughout the remainder of the chapter, we fix a positive integer k and a connected, reflexive, finite graph G. The characterization makes essential use of the categorical product of graphs. We can view the movements of many cops as the movement of a sole cop in a large enough power of the graph. To be more precise, if all the factors in a categorical product of k graphs are equal to a graph G, then we refer to the product as the *kth power* of G, and write G^k. Let $P = P(G) = G^k$. We identify vertices of P with positions of k cops in G; the definition of the product simulates the moves of k cops in G. In particular, if k cops C_i, $1 \leq i \leq k$, move from positions u_i to v_i, where $v_i \in N[u_i]$, then $(v_1, \ldots, v_k) \in N[(u_1, \ldots, u_k)]$ in G^k. We suppress the subscripts in the neighbor set notation in this and the next section, as they are clear from context and would just serve to complicate matters (hence, opting for $N[u]$ over say $N_G[u]$).

For $i \in \mathbb{N}$, the relation $\leq_i$ on $V(G) \times V(P)$ is defined as follows by induction on i. For $x \in V(G)$ and $p \in V(P)$, $x \leq_0 p$ if in position p, at least one of the k cops is occupying x. For $i > 0$, $x \leq_i p$ if and only if for each $u \in N[x]$ there exists a $v \in N[p]$ such that $u \leq_j v$ for some $j < i$. Just as in the cop-win case, the relations $\leq_i$ are non-decreasing sets in i, and hence, as G and P are finite, there is an integer M such that $\leq_M = \leq_{M+1}$ and set $\preceq = \leq_M$. Although the notation $\preceq$ in this case clashes with the one for cop-win graphs, we use it here to avoid introducing more notational baggage.

Theorem 2.9 ([**57**]). *The graph G is k-cop-win if and only if there exists $p \in V(P)$ such that $x \preceq p$ for every $x \in V(G)$.*

The proof of Theorem 2.9 closely follows that of Theorem 2.6. Before we prove Theorem 2.9, we define a bipartite graph $B = B(G)$ with red vertices P and blue vertices G. A blue x is joined to a red y if $x \preceq y$. Theorem 2.9 therefore says that some red vertex is joined to all the blue vertices. By Theorem 2.9 there exists red vertex p and a least integer $i \geq 0$ such that $x \leq_i p$ for all blue x. Each neighbor q of p satisfies $x \leq_{i+1} q$. As G is connected and finite, we have that each red vertex is joined to each blue vertex. Hence, we have the following.

Corollary 2.10 ([**57**]). *A graph G is k-cop-win if and only if the graph $B(G)$ is complete bipartite.*

Proof of Theorem 2.9. For the forward direction, we prove the contrapositive. Suppose that for all $p \in P$, $x \not\preceq p$ for some $x \in V(G)$. The vertex x must have a neighbor y such that $y \not\preceq q$ for all $q \in N[p]$. In particular, at neither p nor q do the cops capture the robber.

If the cops start in position p, then the robber chooses x such that $x \not\preceq p$. If the cops move to q, then as in the previous paragraph, the robber moves to a vertex y such that $y \not\preceq q$. In this way, by induction, the robber can indefinitely avoid capture.

For the reverse direction, we prove by induction on i that if $x \leq_i p$ for $i \leq M$, then the cops occupying p can capture the robber located on x in at most i moves. As the result holds for all x in $V(G)$, the proof of the direction will follow.

The base case is trivial, because one of the cops occupies x. Now suppose it holds for a fixed $i < M$, and consider the case $x \leq_{i+1} p$. If the robber moves from x to y, then the cops move to some q such that

$y \leq_i q$. Using the induction hypothesis, the robber can be captured from y in at most i moves, and so he can be captured from x in at most $i+1$ moves. □

2.3.3. An Elimination-ordering Characterization of k-cop-win Graphs. As we discussed in Section 2.2, cop-win graphs are characterized by a cop-win ordering. In particular, in a cop-win ordering, corners may be successively deleted from lower to higher index terminating with a single vertex. We consider an analogue of this elimination-ordering for k-cop-win graphs, once again due to Clarke and MacGillivray [**57**]. In the case $k = 1$, the vertex ordering reduces to a cop-win ordering.

The elimination-ordering now takes place in $P \times G$. We "process" (or, in the terminology of [**57**], "paint") vertices of $P \times G$ as follows, using the relations $\leq_i$. At step $i = 0$, all vertices (p, x), where x is in one of the k components of p, are labeled as *processed.* For $i > 0$, label any unprocessed (p, x) as *processed* if it is joined to a processed vertex in the P-layer $P.\{y\}$, for every $y \in N[x]$.

The vertices (p, x) processed in step i or earlier are exactly those with $x \leq_i p$. By Theorem 2.9, the graph G is k-cop-win if and only if some G-layer $\{p\}.G$ has each vertex processed. By Corollary 2.10, this is in turn equivalent to each vertex of $P \times G$ being processed.

We now make the sequence by which vertices are processed more explicit. A vertex (p, x) is called *removable* with respect to

$$S \subseteq V(P \times G)$$

if one of the following properties is satisfied.

(1) In position p, at least one of the cops is located at x.

(2) $N[(p, x)] \cap S \cap P.\{y\}$ is non-empty for every $y \in N[x]$.

A *k-cop-win ordering* is a sequence

$$S = ((p_i, x_i) : 1 \leq i \leq t \leq |V(P \times G)|)$$

such that for all $1 \leq i \leq t$, the vertex (p_i, x_i) is removable with respect to $((p_j, x_j) : j < i)$, and $(p_t, x) \in S$ for all $x \in V(G)$.

Theorem 2.11 ([**57**]). *A graph G is k-cop-win if and only if it has a k-cop-win ordering.*

Before we prove Theorem 2.11, we show that a 1-cop-win ordering S of $G \times G$ gives rise to a cop-win ordering (the converse is left as Exercise 23). (Note that if $k = 1$, then $P = G$.) Let (x_1, v_1) be the first pair in S with x_1 distinct from v_1. By definition, $N[v_1] \subseteq N[x_1]$. Define S_1 to be the sequence from S formed by first deleting all pairs (u, v_1), where $u \in V(G)$, then replacing each pair (v_1, y) where $y \in V(G)$ by (x_1, y), and then finally deleting the second instance of all repeated pairs. As $N[v_1] \subseteq N[x_1]$, if (v_1, y) is removable, then so is (x_1, y). It follows that S_1 is a 1-cop-win ordering of $G - v_1$. If we repeat this argument $n - 1$ many times, then we obtain an ordering $v_1, v_2 \ldots, v_{n-1}$, with v_n chosen to be a vertex of G not belonging to this sequence. By construction, this is a cop-win ordering of G.

Proof of Theorem 2.11. If G is k-cop-win, then by Theorem 2.9 and the definition of $\preceq$, listing all vertices in $\leq_0$, then those vertices in $\leq_1$, and so on, stopping once (p_t, x) appears, is a k-cop-win ordering.

Now suppose we are given a k-cop-win ordering. We prove by induction on u that the cops in p_i can capture the robber at x_i in at most i moves. This holds for $i = 1$ since (p_1, x_1) is removable with respect to the empty set. In particular, one of the cops in position p_1 is located at x_1.

Now suppose the condition holds for a fixed $i < t$, and consider (p_{i+1}, x_{i+1}). Then in position p_{i+1} either one of the cops is at x_{i+1} (in which case the robber is captured) or

$$N[(p_{i+1}, x_{i+1})] \cap S \cap P.\{y\}$$

is non-empty for every $y \in N[x_{i+1}]$. In the latter case, as (p_{i+1}, x_{i+1}) is removable with respect to $((p_j, x_j) : j < i + 1)$, for all $y \in N[x_{i+1}]$, there is some $q \in N[p_{i+1}]$ such that

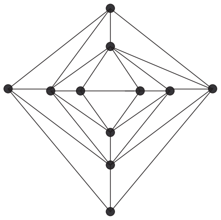
$$(q, y) \in \{(p_j, x_j) : j < i + 1\}.$$

The proof now follows by induction. □

We view Theorem 2.11 as a higher-dimensional analogue of the dismantlability characterization of cop-win graphs. In particular, the dismantling occurs in a suitable power of the graph G. Note that both the relation $\preceq$ and the k-cop-win ordering can be computed in polynomial time, with complexity $O(n^{f(k)})$, where $f(k)$ is a polynomial in k. See Chapter 5 for more on the complexity of Cops and Robbers. Despite this fact, it is much more challenging to apply Theorem 2.11 directly (that is, without computer aid), even to determine if a graph is 2-cop-win. For example, to determine a 2-cop-win ordering of C_4, one builds a power graph $(C_4)^3$ and processes the vertices there. It is much easier to determine that $c(C_4) = 2$ by hand than finding an elimination-ordering of a graph with 64 vertices! (See Exercise 24). Nevertheless, Theorem 2.11 represents a breakthrough in the theory of k-cop-win graphs where $k > 1$, and will likely have an impact for many years to come.

Exercises

1. Find a cop-win ordering of the following graph:



2. Find infinitely many examples of cop-win graphs with exactly one corner.
3. Give an example of an infinite cop-win graph G where for each positive integer k, the robber can arrange to play on G and survive for at least k moves.
4. (a) Prove that a chordal graph (see Exercise 4 in Chapter 1) is cop-win.
   (b) Show that a chordal graph has a *perfect elimination ordering*; that is, an ordering of the vertices such that, for each vertex u, u and the neighbors of u that occur later than u in the ordering form a clique.
   (c) Show that a perfect elimination ordering of a chordal graph is a cop-win ordering.

5. [**98**] A *bridge* is an edge whose deletion increases the number of connected components. Prove that if G is a cop-win graph, then each edge uv of G is either a bridge or there is a vertex $w \in V(G)$ joined to both u and v. In particular, in a cop-win graph without bridges (a so-called *bridgeless graph*), each edge of G belongs to some K_3.

6. A graph is *bridged* if it contains no isometric cycles of length greater than 3.
   (a) Show that each chordal graph is bridged, but not all bridged graphs are chordal.
   (b) [**12, 46**] Prove that a bridged graph is cop-win.

7. Suppose that G has an induced cycle with length at least 4, and at least one vertex of the cycle has degree at least 2. Show that G is not cop-win. (*Hint*: For a contradiction, assume that G has a cop-win ordering.)

8. Verify inequalities (2.1) and (2.2).

9. For each of $i = 1$, 2, and 3, give infinitely many examples of planar graphs G with $c(G) = i$.

10. [**52**] A graph G with $c(G) \leq 2$ is *tandem-win* if, under the restriction that the cops stay a distance of at most one apart during play, the cops have a winning strategy.
    (a) Prove that tandem-win graphs are cop-win.
    (b) Show that tandem-win graphs are closed under taking retracts.
    (c) A vertex v is *nearly irreducible* (or *o-dominated*; see Chapter 8) if there if there exists a vertex $y \neq v$ such that $N(v) \subseteq N[y]$. Prove that if v is nearly irreducible, then G is tandem-win if and only if $G - v$ is tandem-win.
    (d) Find an example of a tandem-win graph with no nearly irreducible vertices.

11. Prove that both the categorical and strong products are associative.

12. Show that $G \times H$ is connected if and only if both G and H are connected and at least one of them is non-bipartite.

13. For graphs G and H, let $\mathrm{Hom}(G, H)$ be the set of homomorphisms between G and H. We turn $\mathrm{Hom}(H, G)$ into a graph by setting vertices to be homomorphisms, and fg is an edge if $f(u) = g(u)$ for all u in G, except for possibly one vertex of G. Note that this is the empty graph if there is no homomorphism from G to H. For simplicity, we assume $\mathrm{Hom}(H, G) \neq \varnothing$.
   (a) Determine the graph $\mathrm{Hom}(K_{1,3}, K_3)$.
   (b) Show that $|\mathrm{Hom}(G, K_n)|$ is the number of proper n-colorings of G.
   (c) Prove that $|\mathrm{Hom}(K_n, G)|$ is $n!$ times the number of n-cliques in G.
   (d) [**34**] Show that G is dismantlable if and only if the graph $\mathrm{Hom}(H, G)$ is connected, for each graph G.
   (e) [**140**] Let G and H be finite graphs. Prove that if
$$|\mathrm{Hom}(J, G)| = |\mathrm{Hom}(J, H)|$$
   for all finite graphs J, then $G \cong H$. (*Hint*: Show that the number of injective homomorphisms from J to G equals the number of injective homomorphisms from J to H.)

14. Graphs G and H are *homeomorphic* if they can be obtained from a graph J by subdividing edges. For example, any two cycles are homeomorphic.
   (a) Find a 2-cop-win graph homeomorphic with the Petersen graph.
   (b) [**16**] Prove that a connected graph is homeomorphic to a graph with cop number at most two.

15. Find examples of irreflexive cop-win graphs G and H, where $c(G \times H)$ is one, two, or three.

16. The *Cartesian product* of G and H, written $G \Box H$, has vertex set $V(G) \times V(H)$. Vertices (a, b) and (c, d) are joined if $a = c$ and $bd \in E(H)$, or $ac \in E(G)$ and $b = d$.
   (a) Justify the notation $G \Box H$ by considering $K_2 \Box K_2$.
   (b) [**192**] Prove that
$$c(G \Box H) \leq c(G) + c(H).$$
   (c) [**144**] Prove that the cop number of the Cartesian product of n trees is at most $\left\lceil \frac{n+1}{2} \right\rceil$.

17. [**164**] Let G be the strong product of n cycles, each of length at least 5. Prove that $c(G) \leq n+1$.

18. Prove that $\leq_\alpha$ is a *quasi-order* (that is, a reflexive and transitive relation) for each ordinal α.

19. In a cop-win graph, the *cop plays optimally* if she catches the robber in the least number of rounds. The *robber plays optimally* if he is caught in the maximum number of rounds.
    (a) Describe an optimal play of both the cop and robber on a tree.
    (b) Assuming both players play optimally, show that for a positive integer k, $y \leq_k x$ but not $y \leq_{k-1} x$ if and only if when the cop is at x and the robber is at y with the robber to move, the cop captures the robber in exactly k moves.

20. Let G be a graph with a corner v. Show that G is k-cop-win if and only if $G - v$ is k-cop-win.

21. [**34**] A graph is *stiff* if it does not contain any corners.
    (a) Show that a regular graph with at least three vertices is stiff.
    (b) Show that each graph G contains a unique isomorphism type of stiff subgraph H for which G retracts to H by a sequence of 1-point retractions. The subgraph H is called the *stiff-core*.
    (c) Find the stiff core of the Petersen graph, and of the dodecahedron.

22. Let G be a cop-win graph with vertices $[n]$, and consider a sequence S of retractions $i \to j$, for $i \in [n]$ and some $j < i$. Define the *cop-win spanning tree* T (relative to S) so that $V(T) = V(G)$ and $ij \in E(T)$ if and only if $i \to j$ in S. Clarke [**52**] first introduced cop-win spanning trees.
    (a) Justify that in fact T is a tree.
    (b) Find all the cop-win spanning trees for the cop-win graph shown in Exercise 1.

23. Show that if G has a cop-win ordering, then $G \times G$ has a 1-cop-win ordering. (*Hint*: Use a cop-win spanning tree.)

24. Using a computer or directly, find a 2-cop-win ordering of $(C_4)^3$.

Chapter 3

Meyniel's Conjecture

3.1. Introduction

Every mathematician loves a good conjecture. All robust fields of graph theory or mathematics need at least one easy to state but tough to solve problem. Such problems should spur sufficient interest and debate, leading to new ideas and techniques. We think of the impact of Hilbert's 23 problems—posed in 1900—on modern mathematics; see [**201**]. One hundred years later, the Clay Foundation posted seven Millennium prize problems [**62**]; the solution of any one problem comes with the prize of one million dollars!

Meyniel's conjecture is fulfilling the role of an elegant but challenging problem on the game of Cops and Robbers. Recall that we consider connected graphs only. Meyniel's conjecture states that if G is a graph of order n, then

$$c(G) = O(\sqrt{n}). \tag{3.1}$$

In other words, for n sufficiently large there is a constant $d > 0$ such that

$$c(G) \leq d\sqrt{n}.$$

We will refer to (3.1) as the *Meyniel bound.* The conjecture was mentioned in Frankl's paper [**89**] as a personal communication to him by Henri Meyniel in 1985 (see page 301 of [**89**] and reference [8] in that paper). Despite this somewhat cryptic reference, Meyniel's conjecture stands out as one of the deepest (if not *the* deepest) problems on the cop number. See Figure 3.1 for the only photograph we could find of Meyniel.

Figure 3.1. Henri Meyniel in Aussois, France, in the 1980s. Photo courtesy of Geňa Hahn.

For n a positive integer, let $c(n)$ be the maximum value of $c(G)$, where G is of order n. For example, $c(1) = c(2) = c(3) = 1$, while $c(4) = c(5) = 2$. Note that $c(n)$ is a non-decreasing function. (See Exercise 1b). We can rephrase Meyniel's conjecture more compactly as

$$c(n) = O(\sqrt{n}).$$

At the heart of Meyniel's conjecture, of course, is finding good upper bounds for the cop number. As demonstrated in the proof of Theorem 1.5, for a projective plane with $q^2 + q + 1$ many points, the bipartite graph $G(P)$ has cop number equaling $q + 1$. Hence, if the conjecture is true, then the bound is asymptotically tight. As a first step towards proving Meyniel's conjecture, Frankl [**89**] proved that $c(n) = o(n)$. More precisely, he proved (see Theorem 1.6 of Chapter 1) that

$$c(n) \leq (1 + o(1))n \frac{\log \log n}{\log n}. \tag{3.2}$$

There is a large gap in the bound of the conjecture and (3.2). Over 20 years passed, and the conjecture received relatively little attention. It is we think, deserving of more attention, and so it is the focus of this chapter.

Chinifooroshan [**47**] in 2008 improved (3.2) by showing that

$$c(n) = O\left(\frac{n}{\log n}\right). \tag{3.3}$$

The best known upper bound at the time of writing this book is due to Lu and Peng [**141**], who proved using the probabilistic method that

$$c(n) = O\left(\frac{n}{2^{(1-o(1))\sqrt{\log_2 n}}}\right). \tag{3.4}$$

The same bound was achieved a little time later and independently by Scott and Sudakov [**187**], and Frieze et al. [**90**]. The bound (3.4) is still far from the Meyniel bound, especially for large n. Ignoring constants, if n is a billion, then the Meyniel bound states that fewer than $32,000$ cops are needed to capture the robber, while (3.4) requires over 42 million.

To further highlight how far we are from proving the conjecture, even the so-called *soft Meyniel's conjecture* is open, which states that for a fixed constant $c > 0$,

$$c(n) = O(n^{1-c}).$$

At a glance, the soft Meyniel's conjecture is a much weaker one than the original one. Nevertheless, solving the soft conjecture would represent a significant breakthrough. Even establishing that, say,

$$c(n) = O(n^{0.99999999})$$

remains wide open!

In Section 3.2 we prove in Theorem 3.1 the bound (3.3), whose proof uses the notion of guarding subgraphs. The reader will recall from Chapter 1 that (3.2) was proved using isometric paths and the Moore bound. The main idea of the proof of Theorem 3.1 is to generalize guarding paths with one cop to guarding certain trees with at most five cops. We discuss families of graphs realizing the tightness of the Meyniel bound (3.1) in Section 3.3. In Section 3.4, we prove the conjecture in the special case of diameter 2 graphs; see Theorem 3.10. We close with some discussion and problems surrounding the conjecture.

3.2. An Improved Upper Bound for the Cop Number

For many years, Frankl's bound (3.2) was the best known upper bound on the cop number. The bound given in the following theorem was proved by Chinifooroshan in 2008.

Theorem 3.1 ([**47**]).

$$c(n) = O\left(\frac{n}{\log n}\right). \tag{3.5}$$

The bound (3.5), therefore, represents the first important step forward in proving Meyniel's conjecture in over 25 years. The key to proving (3.3) comes from the notion of guarding an induced subgraph introduced in Chapter 1 and which we recall here. For a fixed integer $k \geq 1$, an induced subgraph H of G is *k-guardable* if, after finitely many moves, k cops can move only in the vertices of H in such a way that if the robber moves into H at round t, then he will be captured in round $t+1$.

The idea is that with some work, the cop can arrange things so that as soon as the robber enters his turf, the robber is captured. An easy example of a 1-guardable subgraph is a clique; an isometric path is 1-guardable (see Theorem 1.7). We now consider a special class of trees which require a few more cops to effectively guard. A *minimum distance caterpillar* (or *mdc*) is an induced subgraph H of G with the following properties.

(1) The graph H is a tree.

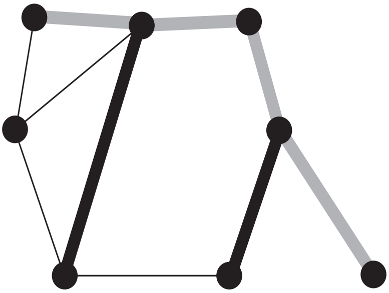


Figure 3.2. An example of an mdc, represented by the thicker lines. The grey lines form the path P.

(2) There is a path P in H that is *dominating*: that is, for each vertex u of H not in P, there is a vertex v of P joined to u.

Figure 3.2 gives an example of a minimum distance caterpillar. Frankl's proof of (3.2) relies heavily on the fact that isometric paths are 1-guardable. The strategy of the proof of Theorem 3.1 rests on first proving that mdc's are 5-guardable. We then show that small order mdc's always exist in graphs (where "small order" means order $\log n$), and then we use the greedy approach to bound the cop number.

Mdc's are "sticky" analogues of isometric paths, and require just a few more cops to guard.

Theorem 3.2 ([**47**])**.** *An mdc is* 5*-guardable.*

Proof. Let H be an mdc, and with an isometric path P, which has vertices $\{1, 2, \ldots, k\}$. Since isometric paths are 1-guardable (see Theorem 1.7), the path P is 1-guardable. We station a cop C_0 there on a permanent beat: her role is just to capture the robber in round $t+1$ if he enters P in round t.

We now see how the four additional cops can guard the rest of H. The cop C_0 divides P into vertices of behind her (that is, of lower index) and in front of her (higher index). We now place two cops directly in front of C_0 and two cops directly behind her. That is, if C_0 occupies i, then C_{-2} moves to $i-2$, C_{-1} moves to $i-1$, C_1 moves to $i+1$, and C_2 moves to vertex $i+2$. If any of these values are below 1 or above n, then the cops just double-up on vertices (for example,

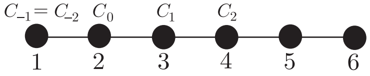


Figure 3.3. Cops doubling up at the end of the isometric path.

if C_0 is on 2, then C_{-2} and C_{-1} both occupy 1). See Figure 3.3. It is evident that after finitely many rounds, the cops can arrange to move in this fashion.

To show that H is 5-guardable, we consider cases. If the robber moves onto P in round t, then he will be caught in round $t+1$. Now suppose that the robber moves onto v in round t, where v is not in P. By hypothesis, there is a vertex u of P joined to v. We must have that C_0 is on $u-2$, $u-1$, u, $u+1$, or $u+2$ in round t. Otherwise, the robber can move onto u in round $t+1$, and not be caught in round $t+2$. Owing to the location of the cops C_i, where $-2, -1 \leq i \leq 1, 2$, one of the cops must be on u at round t, and so can capture the robber in round $t+1$. □

We now show that mdc's of sufficiently large order always exist in G.

Theorem 3.3 ([**47**]). *If G has order n, then there is an mdc in G of order at least* $\log n$.

The proof of Theorem 3.1 follows with the aid of Lemma 3.4, which finds isometric paths with large neighborhoods.

We use Theorems 3.2 and 3.3 together with Lemma 3.4 to obtain 5-guardable subgraphs with large neighborhoods. We actually prove the following more general fact, which is used in the proof of Theorem 3.1 in the case $d = 1$. For a set S of vertices in a subgraph H and d a positive integer, let $N_H^d[S]$ be the set of vertices of distance d or less in H from a vertex of S; see Figure 3.4 for an example. A *rooted tree* is one with a specified vertex called the *root*. A *root-to-leaf path* in a rooted tree consists of a path with endpoints at the root and an end-vertex (or leaf).

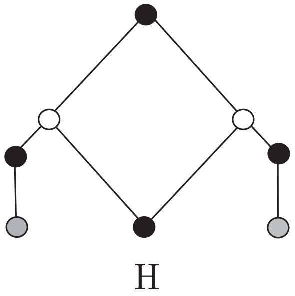


Figure 3.4. The set S equals the white vertices, while $N_H^1[S]$ equals the white and black vertices, and the set $N_H^2[S]$ equals the white, black, and grey vertices.

Lemma 3.4 ([**47**])**.** *For integers $n, d \geq 1$ with $d \leq n$ and any rooted n-vertex tree T, T has a root-to-leaf path P such that*

$$\left|N_T^d[P]\right| \geq \frac{d\log(1+\frac{n}{d})}{1+\log d}.$$

Proof. Let $\tau(n,d)$ be the largest number such that any rooted n-vertex tree T has a root-to-leaf path P such that

$$|N_T^d[P]| \geq \tau(n,d).$$

We use induction on n to prove that

$$\tau(n,d) \geq \frac{d}{1+\log d}\log\left(1+\frac{n}{d}\right).$$

As for the base case, it is straightforward to see that for all $1 \leq n \leq 2d$,

$$\tau(n,d) = n \geq \frac{d}{1+\log d}\log\left(1+\frac{n}{d}\right).$$

We assume that the hypothesis is true for all integers up to $n \geq 2d$, and we prove that

$$\tau(n+1,d) \geq \frac{d}{1+\log d}\log\left(1+\frac{n+1}{d}\right).$$

Now, let T be an $(n+1)$-vertex tree in which all root-to-leaf paths P satisfy $|N_T^d[P]| \leq \tau(n+1,d)$, let r be the root of T, let B_i be the set of vertices of distance at most i from r, and define $b_i = |B_i|$. We can assume that $b_d - b_{d-1} > 0$, otherwise, if $b_d = b_{d-1}$, all the vertices of T are at distance at most $d-1$ of r, and thus,

$$\tau(n+1,d) \geq |N_T^d[r]| = n+1 \geq \frac{d}{1+\log d} \log\left(1+\frac{n+1}{d}\right).$$

Since any path of length $d-1$ has d vertices, $b_{d-1} \geq d$. Let $v \in B_d \setminus B_{d-1}$ be the vertex that maximized the number of vertices in T_v, the subtree of T rooted at v. It is immediate that

$$|V(T_v)| \geq \frac{n+1-b_{d-1}}{b_d - b_{d-1}}.$$

Therefore, there is a path P_v in T_v from v to an end-vertex such that

$$|N_{T_v}^d[P_v]| \geq \tau\left(\frac{n+1-b_{d-1}}{b_d-b_{d-1}}, d\right).$$

Let $P_{r,v}$ denote the path from r to v in T from which v is removed. By joining $P_{r,v}$ and P_v we obtain a root-to-leaf path P in T, and have that

$$\begin{array}{rcl}
\tau(n+1,d) & \geq & \left|N_T^d[P]\right| \\
& \geq & \tau\left(\frac{n+1-b_{d-1}}{b_d-b_{d-1}}, d\right) + b_d - 1 \\
& \geq & \tau\left(\frac{n+1-d}{b_d-d}, d\right) + b_d - 1 \\
& \geq & \frac{d\log\left(1-\frac{1}{b_d-d}+\frac{n+1}{d(b_d-d)}\right)}{1+\log d} + b_d - 1 \\
& = & \frac{d\log\left(\left(1-\frac{1}{b_d-d}\right)(2d)^{\frac{b_d-1}{d}} + \frac{(2d)^{\frac{b_d-1}{d}}}{b_d-d}\frac{n+1}{d}\right)}{1+\log d} \\
& \geq & \frac{d\log\left(1+\frac{n+1}{d}\right)}{1+\log d}. \quad \square
\end{array}$$

The lower bound of $\frac{d\log(1+\frac{n}{d})}{1+\log d}$ is not necessarily tight; however, it cannot be larger than $2d\log(1+\frac{n}{d})$ by considering the complete binary tree in which all the edges are subdivided $d-1$ times; see Figure 3.5.

We now prove the main result of this section.

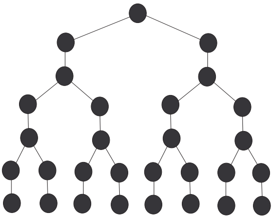


Figure 3.5. A complete binary tree in which all edges are subdivided once.

Proof of Theorem 3.1. Fix a graph G of order n. By Theorem 3.3 we may find an mdc H in G of order at least $\log n$. By Theorem 3.2 H is 5-guardable. We now use the greedy approach exploited in Theorem 1.6. Station five cops on H to guard it. Delete H, and consider the connected component of $G - H$ with the largest cop number, which we call G'. Then

$$c(G) \leq c(G') + 5. \tag{3.6}$$

Now proceed by induction using (3.6) to derive that

$$c(n) \leq c\left(\frac{n}{2}\right) + 5\frac{n/2}{\log n}. \tag{3.7}$$

Hence, by induction and (3.7), we derive the desired bound

$$c(G) = O\left(\frac{n}{\log n}\right). \qquad \square$$

The use of mdc's in the proof of Theorem 3.1 and isometric paths in Frankl's proof of (3.2) points to the use of finding larger order k-guardable subgraphs, where k is a constant. As noted in Chapter 1, isometric paths are retracts in reflexive graphs: the cops stay on the image of the robber under the retraction. If the robber moves to the subgraph, then the cop captures the robber on his image or shadow there. One could imagine exploiting larger retracts in graphs as an approach to proving Meyniel's conjecture. Unfortunately, this will not substantially improve upper bounds on the cop number for general graphs. A recent result from [**19**] puts a poly-logarithmic upper bound on the order of retracts in some graphs. The proof relies on the probabilistic method.

Theorem 3.5. *For all integers $n > 0$, there is a graph of order n whose largest retract is of order $O(\log^8 n)$.*

An improvement exists to the bound (3.5) in Theorem 3.1. The following theorem was proved by Lu and Peng [**141**] in 2009. For a vertex u, the set $N_i(u)$ is the set of vertices of distance i to u.

Theorem 3.6 ([**141**])**.** *Fix an integer $n > 0$, and let G have order n.*

(1) *For a fixed $k < n$, define*

$$M_k = \min_{v \in V(G)} |N_{2^{k-1}}(v)|.$$

(In particular, $M_1 = \delta$.) Then

$$c(G) \leq 8k \left(\frac{\log n}{M_k} \right)^{1/k} n.$$

(2) *If* $\operatorname{diam}(G) \leq 2^{k-1}$, *then*

$$c(G) \leq 8kn^{1-1/k} \log^{1/k} n.$$

(3) *The function $c(n)$ satisfies*

$$c(n) = O\left(\frac{n}{2^{(1-o(1))\sqrt{\log_2 n}}} \right). \tag{3.8}$$

The bound $O\left(\frac{n}{2^{(1-o(1))\sqrt{\log_2 n}}} \right)$ in (3.8) is currently the best upper bound for general graphs that is known, but it is still far from proving Meyniel's conjecture or even the soft version of the conjecture. The proof of Theorem 3.6 (which is omitted) uses the greedy approach as in the proofs of Theorems 1.6 and 3.1, as well as the probabilistic method, which represents a new and interesting approach to proving the conjecture.

3.3. How Close to $\sqrt{n}$?

Meyniel's conjecture states that the cop number is at most approximately $\sqrt{n}$. Examples are known (and will be discussed immediately below) which have cop number very close to $\sqrt{n}$. However, the question remains how close the cop number can approach $\sqrt{n}$ *from below.*

For graphs with large cop number, we turn to incidence graphs. An *incidence structure* consists of a set P of points and a set L of lines along with an incidence relation consisting of ordered pairs of points and lines. Given an incidence structure S, we define its incidence graph $G(S)$ to be the bipartite graph whose vertices consist of the points (one color) and lines (the second color) with a point joined to a line if it is incident with it in S. Incidence structures (and graphs) are quite general, but we restrict our attention to *partial linear spaces*, where any pair of points (lines) is incident with at most one line (point). It is an exercise that the incidence graph of a partial linear space is diameter at least 3 with girth at least 6.

Projective planes are some of the most well-studied examples of incidence structures. Recall from Chapter 1 that a *projective plane* consists of a set of points and lines satisfying the following axioms:

(1) There is exactly one line incident with every pair of distinct points.

(2) There is exactly one point incident with every pair of distinct lines.

(3) There are four points such that no line is incident with more than two of them.

Hence, projective planes are particular partial linear spaces; condition three rules out certain degenerate cases where all points are on a single line or all lines are on a single point. We are interested in finite projective planes, which always have $q^2 + q + 1$ points for some integer $q > 0$ (called the *order* of the plane). Figure 3.6 depicts the projective plane of order 3 with 13 points.

We recall the projective plane graphs from the proof of Theorem 1.5 in Chapter 1. For a given projective plane P, define $G(P)$ to be the bipartite graph with red vertices the points of P, and the blue vertices represent the lines. Vertices of different colors are joined if they are incident. We call this the *incidence graph of* P; see Figure 3.7 for $G(P)$ where P is the Fano plane (that is, the projective plane of order 2). It is an exercise that the incidence graph of the Fano plane is isomorphic to the *Heawood graph* (see Exercise 6a in this chapter).

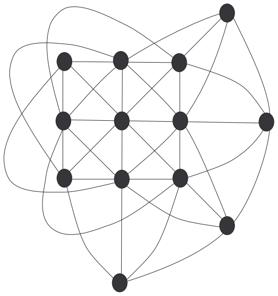


Figure 3.6. The projective plane of order 3.

We showed in Theorem 1.5 and Exercise 15 of Chapter 1 that if P has order q, then $c(G(P)) = q + 1$. However, the orders of $G(P)$ depend on the orders of projective planes. The only orders where projective planes are known to exist are prime powers; indeed, this is a deep conjecture in finite geometry (see [**39**]). A large computer search ruled out the existence of such a plane of order $q = 10$; see [**133**]. For example, it is not even known if there is a projective plane of order $q = 12$.

What about integers which are not prime powers? A family of graphs $(G_n : n \geq 1)$ is *Meyniel extremal* if G_n has order n and for sufficiently large n, there is a constant d such that $c(G_n) \geq d\sqrt{n}$.

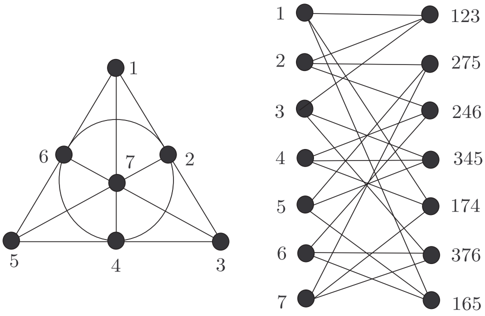


Figure 3.7. The Fano plane and its incidence graph. Lines are represented by triples.

We now give examples of Meyniel extremal classes. We use the following famous so-called *Bertrand's postulate*, first proved by Cheybshev in 1850, on the existence of primes in intervals. (An elementary proof was given by Erdős at the age of 19; see [**72**].)

Theorem 3.7 ([**45**]). *For all integers $x > 1$, there is a prime in the interval $(x, 2x)$.*

The number of primes between x and $2x$ is about $x/\log x$, which follows by the famous *prime number theorem* (for more background on this and other properties of primes, see, for example [**65**]).

The following theorem uses projective planes to give a Meyniel extremal family of graphs.

Theorem 3.8 ([**175**]). *There is a family of graphs H_n of order n which is Meyniel extremal.*

Proof. Consider $n \geq 72$. Let q be a prime power such that

$$2(q^2 + q + 1) \leq n.$$

If $2(q^2+q+1) = n$, then we are done by considering the graphs $G(P)$. Otherwise, assume that $2(q^2 + q + 1) < n$. Let G_q be a graph $G(P)$ of order $2(q^2 + q + 1)$.

Form H_n by adding a path of length $n - 2(q^2 + q + 1)$ to G_q. Hence, H_n has order n. It is straightforward to see that

$$c(H_n) = c(G_q) = q + 1.$$

If there are fewer than $q + 1$ cops, then the robber stays in G_q and never ventures on the newly attached path. In this way, he can avoid capture. Therefore, $c(H_n) = q + 1$.

Now by Theorem 3.7 choose a prime q in the interval

$$\left(\left\lfloor\sqrt{\frac{n}{8}} - 1\right\rfloor, 2\left\lfloor\sqrt{\frac{n}{8}} - 1\right\rfloor\right),$$

and consider a graph G_q of order $2(q^2 + q + 1)$. By the choice of n, the left-hand side of the interval is at least one. Then

$$\sqrt{\frac{n}{8}} \leq c(H_n) \leq \sqrt{\frac{n}{2}},$$

and the result follows. □

In particular, we have that

$$c(n) \geq \sqrt{\frac{n}{8}} \tag{3.9}$$

for $n \geq 72$. Using more number theory, we can make the bound (3.9) even tighter for large values of n.

Theorem 3.9 ([**175**]). *For sufficiently large integers x, there is a prime in $(x - x^{0.525}, x)$.*

Using this theorem and the technique in the proof of Theorem 3.8, we have that for sufficiently large n,

$$c(n) \geq \sqrt{\frac{n}{2}} - n^{0.2625}. \tag{3.10}$$

We do not know if (3.10) is the best possible lower bound for $c(n)$, and it would be interesting to find out.

3.4. Meyniel's Conjecture in Graph Classes

There are a myriad of graph classes, some of which we have already discussed. Formally, a *graph class* is a set of graphs closed under taking isomorphism. This is quite a broad definition. Planar graphs, graphs with bounded diameter, graphs with a given chromatic number, perfect graphs, and asteroidal triple-free graphs are all examples of graph classes. For a good reference on graph classes, see [**33**].

While Meyniel's conjecture is unresolved for general graphs, we may attempt to solve it in certain graph classes. In some cases, the extra structure available in a class of graphs can bound the cop number from above more easily. For example, Aigner and Fromme [**2**] proved that $c(G) \leq 3$ if G is planar. For a fixed graph H, Andreae [**10**] generalized this result by proving that the cop number of a K_5-minor-free graph (or $K_{3,3}$-minor-free graph) is at most 3 (recall that planar graphs are exactly those which are K_5-minor-free and $K_{3,3}$-minor-free). Andreae [**11**] also proved that for any graph H the cop number of the class of H-minor-free graphs is bounded by a constant.

We consider in this section a recent proof by Lu and Peng [**141**] that the Meyniel bound holds in the class of graphs with diameter 2. The proof uses the notion of guarding a subgraph described in

Chapter 1, but it also uses a randomized argument. An advantage of randomized methods is their ability to prove some object exists without actually explicitly constructing it. The probabilistic method—championed by Erdős and Rényi—is a central tool in combinatorics and a number of other disciplines. See the text of Alon and Spencer [**6**] for a thorough survey of the method. Also see Chapter 6, which is devoted to Cops and Robbers played on random graphs.

Our main theorem of the section is the following, which establishes that Meyniel's conjecture is true for the class of graphs of diameter 2.

Theorem 3.10 ([**141**]). *If G is a graph on n vertices with diameter* 2, *then*

$$c(G) \leq 2\sqrt{n} - 1. \tag{3.11}$$

The same bound (3.11) was also shown in [**141**] in the case when G is bipartite and of diameter at most 3. Our main tool in proving Theorem 3.10 is the following lemma from [**141**]. For a positive integer k, a graph H is called *k-degenerate* if every subgraph of H has a vertex with degree at most k. For example, a clique of order at most $k+1$ is k-degenerate, as well as a star $K_{1,k}$.

Lemma 3.11 ([**141**]). *Suppose that G has diameter* 2. *Consider a k-degenerate subgraph H of G, and assume that the movements of the robber are restricted to the edges of H. Then k cops can win in this modified game.*

Proof. The proof proceeds by induction on the order of H. If we have that $|V(H)| \leq k$, then we place at least one cop on each vertex of H and the proof follows. Now suppose the conclusion holds for a k-degenerate graph of order m, where $m \geq k$ is fixed. We consider H of order $m+1$.

Let v be a vertex of degree at most k in H. Each connected component of $H - v$ is k-degenerate. If the robber moves only in a single connected component of $H - v$, then the inductive hypothesis applies, and the robber may be captured by k cops. Hence, after some number of rounds, the robber is forced onto the vertex v.

Suppose that the cops are on the vertices

$$x_1, x_2, \ldots, x_k$$

(not necessarily all distinct), and list the neighbors of v in H as

$$y_1, y_2, \ldots, y_j,$$

where $j \leq k$. If one of the cops is on the vertex y_i, then the robber immediately loses. Otherwise, each cop is distance at most 2 from the neighbors of v; see Figure 3.8. In the next round, each cop moves along such a path so the cops are adjacent or equal to each vertex y_i. No matter what the robber does (that is, either passes, or moves to some y_i), he will be caught in at most two rounds. □

Lemma 3.11 also holds for other classes of graphs, such as bipartite graphs of diameter 3; see [**141**] and Exercise 12.

The idea of the proof of Theorem 3.10 is to divide the graph into a $(k-1)$-degenerate subgraph (which is guarded by $(k-1)$-many cops) and a subgraph with a random number of cops. We use the probabilistic method to bound the random number of cops, thereby proving the upper bound (3.11).

A (*discrete*) *random variable* X on a probability space S is a function $X : S \to \mathbb{R}$. The expectation (also called the *mean, average,* or *first moment*) of a random variable X, written $\mathbb{E}(X)$, is defined by

$$\mathbb{E}(X) = \sum_{s \in S} X(s)\mathbb{P}(\{s\}).$$

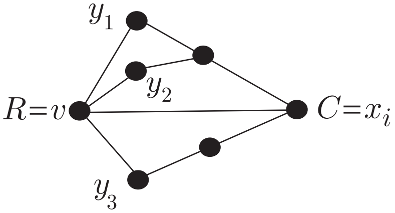


Figure 3.8. A cop is distance at most 2 from the neighbors of R.

A basic but useful fact in the probabilistic method (especially when applied to graphs), is that there is an element in the probability space for which

$$X \leq \mathbb{E}(X). \tag{3.12}$$

The elementary bound (3.12) plays a crucial role in the proof of Theorem 3.10. Another useful tool in the proof of Theorem 3.10 is the notion of a k-core. For a given positive integer k, the *k-core* is the (unique) maximum subgraph with minimum degree k; see Figure 3.9 for the 2- and 3-cores of a graph. Note that the k-core is empty if $\Delta \leq k-1$. To find the k-core, we can just successively delete vertices with degree less than k; the remaining subgraph is the k-core (see Exercise 17b).

Proof of Theorem 3.10. Fix $\epsilon > 0$ a constant, and a positive integer k that will be explicitly specified later. Define

$$p = \frac{\epsilon}{k+1},$$

and note that $p \in (0,1)$ for suitable choice of ϵ.

Define $G_0 = G$, and let H_0 be the k-core of G_0. We place a cop on a vertex of H_0 independently with probability p. Note that the number of cops on H_0 is a random variable, whose expected value is

$$p|V(H_0)| \leq pn.$$

Delete the vertices occupied by these cops and their neighbors, resulting in the graph G_1. Assuming G_i was defined, let H_i be the k-core

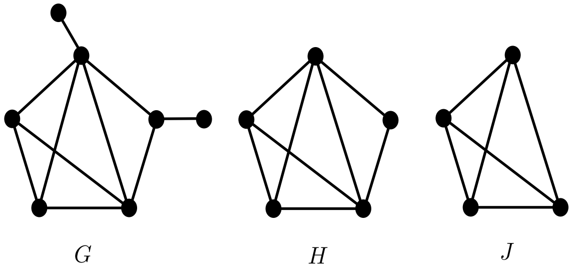


Figure 3.9. The graph G, its 2-core H and 3-core J.

of G_i. Add cops to the vertices of H_i independently with probability p, and delete the vertices with cops and their neighbors to form G_{i+1}. Continue with this process until either H_i or G_i are empty.

Let H be the induced subgraph of G on

$$V(H) = \bigcup_{i \geq 0} V(G_i) \setminus V(H_i).$$

By construction, each vertex of $G - H$ either contains a cop or is adjacent to one. Hence, for the robber to survive, he must move along edges of H. Further, it is straightforward to see that H is $(k-1)$-degenerate.

By Lemma 3.11, $(k-1)$-many cops can guard H, while the remaining cops control $G - H$. This represents a winning strategy for the cops. Therefore, to bound the cop number of G from above, we need to estimate the number of cops in $G - H$. However, the number of cops there is a random variable!

For each $i \geq 0$, the expected number of cops in H_i is

$$p\mathbb{E}\left(|V(H_i)|\right).$$

For a vertex v in H_i, the probability that v is not deleted when G_{i+1} is formed is equal to

$$\begin{aligned}(1-p)^{1+\deg_{H_i}(v)} &\leq (1-p)^{k+1}\\ &\leq \exp(-p(k+1)),\end{aligned}$$

using the inequality that $1 - x \leq e^{-x}$ for real x.

Therefore,

$$\begin{aligned}\mathbb{E}\left(|V(H_i)|\right) &\leq \mathbb{E}\left(|V(G_i)|\right)\\ &\leq \exp(-p(k+1))\mathbb{E}\left(|V(H_i)|\right)\\ &\leq \exp(-p(i+1)(k+1))\mathbb{E}\left(|V(H_0)|\right)\\ &\leq \exp(-p(i+1)(k+1))n,\end{aligned}$$

where the third inequality follows by induction.

Let X be the number of cops needed in the above winning strategy. We have that

$$\mathbb{E}\left(X\right) \leq k - 1 + \sum_{i \geq 0} \exp(-pi(k+1))np. \tag{3.13}$$

By (3.12), there is some random instance of a placement of cops such that $X \leq \mathbb{E}(X)$. Because $c(G) \leq X$ and (3.13) we have that

$$\begin{aligned} c(G) &\leq k - 1 + \sum_{i \geq 0} \exp(-p(i+1)(k+1))np \\ &= k - 1 + \frac{np}{1 - \exp(-p(k+1))} \\ &= k - 1 + \frac{\epsilon}{1 - e^{-\epsilon}} \frac{n}{k+1}, \qquad (3.14) \end{aligned}$$

by properties of geometric series and the choice of p.

Now define

$$f(x) = \frac{x}{1 - e^{-x}}.$$

In particular, by (3.14) we have that

$$c(G) \leq k - 1 + f(\epsilon)\frac{n}{k+1}. \qquad (3.15)$$

Now choose $k = \left\lceil \sqrt{f(\epsilon)n} \right\rceil - 1$ to minimize the value of (3.15). Then

$$\begin{aligned} c(G) &\leq \left\lceil \sqrt{f(\epsilon)n} \right\rceil - 2 + f(\epsilon)\frac{n}{\left\lceil \sqrt{f(\epsilon)n} \right\rceil} \\ &\leq 2\sqrt{f(\epsilon)n} - 1. \qquad (3.16) \end{aligned}$$

It is not hard to see that as $\epsilon \to 0$, $f(\epsilon) \to 1$. As ϵ was arbitrary, we have from (3.16) that

$$c(G) \leq 2\sqrt{n} - 1,$$

as desired. □

The incidence graphs of projective planes are bipartite of diameter 3, and so show that the bound (3.11) is asymptotically tight in that class. However, we do not know of an infinite family of graphs of diameter 2 whose cop number is $c\sqrt{n}$, where c is a constant.

Meyniel's conjecture remains open for most other graph classes. It would be interesting to verify it in classes where the chromatic number is bounded by some constant k.

The only Meyniel extremal class of graphs we are aware of are based on the projective plane graphs $G(P)$ or closely related incidence structures, such as affine planes. Are there others? See Exercise 16. Perhaps if we replace the plane P by more exotic incidence structure or combinatorial design, we will find another Meyniel extremal class. It is also open to improve the lower bound (3.10) on the function $c(n)$ given by

$$c(n) \geq \sqrt{\frac{n}{2}} - n^{0.2625}.$$

Another fascinating topic is the analogue of Meyniel's conjecture in digraphs. Directed graphs are unusual, in that we have no structural characterization for cop-win digraphs akin to cop-win orderings (although an algorithmic characterization is given in [**105**]). There is a notion of corner for digraph, and every cop-win digraph has a corner, but the corresponding retract may fail. For the conjecture to be sensible, we should restrict our attention to strongly connected graphs (otherwise, a digraph can have cop number $n-1$ even if the underlying graph is connected; see Exercise 3). Recent work by Frieze et al. [**90**] using expansion properties shows that the cop number of a connected digraph of order n is $O(n(\log\log n)^2/\log n)$. Can we do better? In other words, does the Meyniel bound hold for strongly connected digraphs? What are the Meyniel extremal digraphs (if any)?

For tournaments, Meyniel's bound fails. A set D is dominating in a tournament, if for each vertex x not in D, there is a vertex y in D with (y, x) a directed edge. The *domination number* of a tournament G, written $\gamma(G)$, is the minimum cardinality of a dominating set. We note the following theorem, attributed by Moon to Erdős (see p. 28 of [**160**]). As $c(G) \leq \gamma(G)$, we have a logarithmic upper bound on the cop number of tournaments.

Theorem 3.12 ([**160**]). *If G is a tournament on n vertices, then $\gamma(G) \leq \lceil \log_2 n \rceil$.*

Proof. As the average out-degree of G is $(n-1)/2$, some vertex has out-degree at least $(n-1)/2$. Deleting this vertex and its out-neighbors and applying a recursion completes the proof. □

Chapter 6 is concerned with the cop number of random graphs. Meyniel's conjecture has been proven for random graphs $G(n, p)$. Let

$p = p(n)$ be a function of n with range in $[0, 1]$. The probability space $\mathcal{G}(n, p) = (\Omega, \mathcal{F}, \mathbb{P})$ of random graphs is defined so that Ω is the set of all graphs with vertex set $[n]$, $\mathcal{F}$ is the family of all subsets of Ω, and for every $G \in \Omega$,

$$\mathbb{P}(G) = p^{|E(G)|}(1-p)^{\binom{n}{2}-|E(G)|} .$$

The space $G(n, p)$ can be viewed as a result of $\binom{n}{2}$ independent coin flips, one for each pair of vertices $\{x,y\}$, with the probability that x and y are joined equaling p. We will abuse notation and consider $G(n, p)$ as a graph, and so write $c(G(n, p))$ (note that the cop number is a random variable on the probability space $G(n, p)$). We say that an event holds *asymptotically almost surely (a.a.s.)* if it holds with probability tending to 1 as $n \to \infty$.

In 2009, Bollobás, Kun, and Leader proved the following result in [**19**], which essentially proves Meyniel's bound in random graphs $G(n, p)$ (up to a multiplicative logarithmic factor) for a wide range of $p = p(n)$; see Theorem 6.11. In particular, it is proven in [**19**] that if

$$p = p(n) \geq 2.1 \log n/n,$$

then a.a.s.

$$c(G(n, p)) = O(\sqrt{n} \log n). \tag{3.17}$$

Recent work by Prałat and Wormald [**179**] removes the $\log n$ factor in (3.17) and hence, proves the Meyniel bound for random graphs (and also for random regular graphs). Based on this result, it would be natural to assume that the cop number of $G(n, p)$ is close to $\sqrt{n}$ for $np = n^{\alpha+o(1)}$, where $0 < \alpha < 1/2$. We will see that this is far from the case in Chapter 6.

Exercises

1. (a) Show that $\lim_{n\to\infty} c(n) = \infty$.
   (b) Prove that the function $c(n)$ is non-decreasing; that is, for all $n \geq 1$,
   $$c(n) \leq c(n + 1).$$

2. The function $c(n)$ can be extended to infinite cardinals. For a cardinal κ, define

$$c(n) = \sup\{c(G) : G \text{ has order } \kappa\}.$$

   Show that for all infinite cardinals κ, $c(\kappa) = \kappa$.

3. Give examples of digraphs of order n whose underlying graph is connected, with the property that their cop number is $n-1$.

4. (a) Prove that for a positive integer k, subdividing each edge of a graph increases the cop number by at most one.
   (b) Give an example of a graph where subdividing each edge results in the cop number remaining unchanged, and another where the cop number increases by exactly one.

5. [**125**] Show that if the cop number of a graph obtained by subdividing the edges of K_n is at most $O(\sqrt{n})$, then Meyniel's conjecture holds.

6. (a) Show that the incidence graph of the Fano plane is isomorphic to the *Heawood graph* in Figure 3.10.
   (b) Prove that the Heawood graph is a 6-*cage*: a cubic graph with girth 6 with the smallest possible order.

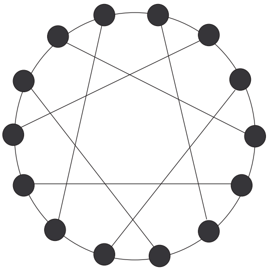


Figure 3.10. The Heawood graph.

7. Prove that for $n > 0$,

$$c(Q_n) = \left\lceil \frac{n+1}{2} \right\rceil.$$

8. [**175**] Let P be a projective plane of order q. Show that
$$c(G(P)) \leq q+1.$$
Hence, the cop number of this graph equals $q+1$.

9. Show that the incidence graph of a partial linear space has diameter at least 3 and girth at least 6.

10. A 2-$(v,k,1)$ *design* is a partial linear space with v points, where all lines (called *blocks*) have order k, and each pair of points is in a unique block.
    (a) Show that there are $\frac{v(v-1)}{v(k-1)}$ many blocks in a 2-$(v,k,1)$ design and each point is on $\frac{v-1}{k-1}$ blocks.
    (b) A *Steiner triple system* is a 2-$(v,3,1)$ design. For example, the Fano plane is a Steiner triple system. Show that a Steiner triple system must have order $v \equiv 1,3 \pmod 6$.
    (c) Prove that the cop number of the incidence graph of a 2-$(v,k,1)$ design is k.

11. Prove Bertrand's postulate for all $x \leq 4000$ without using a computer.

12. [**141**] Show that the conclusion of Lemma 3.11 holds for bipartite graphs of diameter 3.

13. [**113**] Let G and H be graphs both containing given k-cliques
$$u_1, u_2, \ldots, u_k$$
and
$$v_1, v_2, \ldots, v_k,$$
respectively. Define a *clique sum* of G and H, written $G \oplus H$, to be the graph formed by identifying u_i with v_i, where $1 \leq i \leq k$. Note that clique sums are not unique, but depend on the given cliques in G and H.
    (a) Prove that
$$\max\{c(G), c(H)\} \leq c(G \oplus H).$$
    (b) Show that
$$c(G \oplus H) \leq \max\{c(G), c(H)\} + 1.$$

14. The *strong isometric dimension* of a graph G, which is denoted by $\mathrm{idim}(G)$, is defined to be the least number k such that there is a set of k paths $Q_1, Q_2, \ldots, Q_k$ with G an isometric subgraph of the strong product
$$\boxtimes_{i=1}^{k} Q_i.$$
For example, the graph in Figure 3.11 has strong isometric dimension 2.
    (a) Find $\mathrm{idim}(C_4)$ and $\mathrm{idim}(K_5)$.
    (b) [**80**] Prove that if $\mathrm{idim}(G) \le 2$, then $c(G) \le 2$. Much less is known in the case $\mathrm{idim}(G) = 3$. In [**80**], it was shown that if $\mathrm{idim}(G) = 3$, then $c(G) \le \mathrm{diam}(G) + 3$.

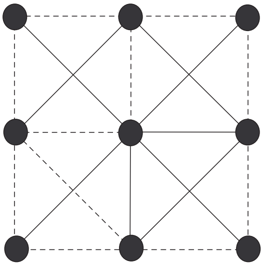


Figure 3.11. A graph with strong isometric dimension 2.

15. [**15**] Define m_k to be the minimum order of a connected graph G satisfying $c(G) \ge k$. It was shown in [**15**] that $m_1 = 1$, $m_2 = 4$, and $m_3 = 10$.
    (a) Prove that $m_k = O(k^2)$. (*Hint*: Use incidence graphs of projective planes and Bertrand's postulate.)
    (b) Show that Meyniel's conjecture is equivalent to the property that
$$m_k = \Omega(k^2).$$

16. Let $\mathcal{A}$ and $\mathcal{B}$ be two Meyniel extremal classes of graphs. Define $\mathcal{A} \times \mathcal{B}$ to be the class of graphs $G \times H$, where $G \in \mathcal{A}$ and $H \in \mathcal{B}$ and the orders of G and H are the same. Prove that $\mathcal{A} \times \mathcal{B}$ is Meyniel extremal.

17. Suppose that $\delta \leq k$, where k is a non-negative integer.
    (a) Show that k-core of a graph is unique up to isomorphism.
    (b) Show that if we can just successively delete vertices with degree less than k, then the remaining subgraph is the k-core.

18. Use the probabilistic method (or otherwise) to show that for all positive integers k, there is a graph whose cop number is at least k.

19. Repeat the previous exercise, but working with tournaments rather than graphs. (*Hint*: First consider what is meant by a random tournament. Then show that a random tournament satisfies a certain adjacency property.)

20. [**111**] A graph G is an *absolute retract* if it is a retract of any graph H containing G as an isometric subgraph.
    (a) Show that every absolute retract is dismantlable.
    (b) Prove that a (reflexive) path is an absolute retract.
    (c) Show that the class of absolute retracts is the smallest variety containing all reflexive paths.

21. [**111**] Let $\mathcal{C}$ be a class of graphs. The *variety generated by* $\mathcal{C}$, written $V(\mathcal{C})$, is the smallest variety of graphs containing $\mathcal{C}$. Show that the class of all absolute retracts equals the variety generated by paths.

22. (a) [**70**] Fix $p \in (0,1)$ a constant. Prove that for every $\varepsilon > 0$, a.a.s. every set of cardinality at least

    $$(1+\varepsilon)\log_{\frac{1}{1-p}} n$$

    is a dominating set in $G(n,p)$.
    (b) [**28**] Show that a.a.s.

    $$c(G(n,p)) = \Theta(\log n).$$

23. Compute $c(n)$ for small values of n. For example, it would be interesting to know the exact value of $c(n)$, for $n \leq 20$.

Chapter 4

Graph Products and Classes

4.1. Introduction

Our approach so far to the cop number has been fairly general. We now focus on the cop number for specific constructions and in certain specified classes of graphs. One of the most popular graph classes in the 20th century were planar graphs: graphs that can be drawn without edge crossings. A beautiful early result was that of Aigner and Fromme in [**2**] proving that the cop number of a planar graph is at most 3; see Theorem 4.25. One of our main goals is to prove this theorem. Along the way, we will consider outerplanar graphs (Theorem 4.23) and graphs of higher genus. We first begin with a discussion of graph products and their cop numbers, which we began to discuss in Chapter 2. As before, all the graphs we consider are connected unless otherwise stated.

Graph products give us interesting ways of forming new graphs from old. The book [**119**] is a good reference on the subject. We introduced three graph products in Chapter 2: the Cartesian, categorical, and strong products. These are not the only products that have been considered. We digress a little to introduce the other products in a systematic way. We use $\otimes$ as the symbol for an arbitrary product, where the vertices of the product graph $G \otimes H$ will always be

$$\{(a, x) : a \in V(G), x \in V(H)\},$$

and whether two vertices in the product are adjacent depends solely on the adjacency relations in the factors. Hence, we can represent a product graph by a 3 × 3 matrix, called the *edge matrix.* The rows (columns) are labeled by E which denotes adjacency of distinct vertices of the first (second) factor, N for non-adjacency, and Δ for the case where there is a loop from the vertex to itself. An E in the matrix indicates there is an edge between the vertices of the product, an N non-adjacency, and in the case where the relationship in both factors is Δ then the two vertices are the same and so the entry is Δ. Here is an incomplete edge matrix:

$$\begin{array}{c} \\ E \\ \Delta \\ N \end{array} \begin{array}{c} \begin{array}{ccc} E & \Delta & N \end{array} \\ \begin{pmatrix} - & - & - \\ - & \Delta & - \\ - & - & - \end{pmatrix} \end{array}.$$

Since the rows and columns will always be labeled in this fashion, we drop the labels in the sequel.

Edge matrices were introduced by Imrich and Izbicki [**118**]. They showed that out of the 256 possible products there are 20 associative products, but only 10 of these depend on the edge structure of both factors (that is, these products do not have all E's or all N's in the first and third rows or in the first and third columns). Further, eight of these are commutative (see Harary and Wilcox [**107**]).

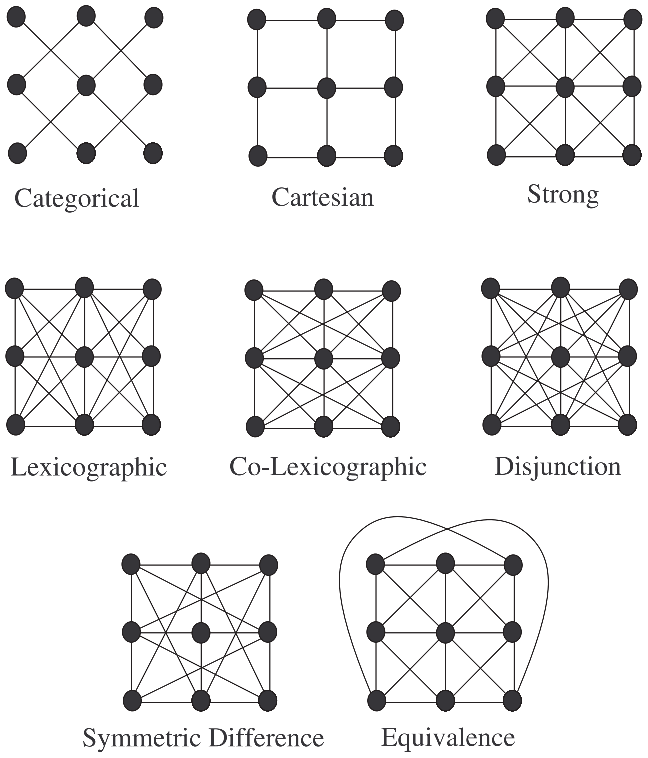
Since a graph can be defined in terms of non-edges, there is the notion of a complementary product. Specifically, the *complementary* product $\otimes^c$ to a product $\otimes$ is given by

$$G \otimes^c H = \overline{(\overline{G} \otimes \overline{H})}.$$

The symbols used to denote products are based mainly on those found in [**162**]. Some of these products are known by other names (for more details, see [**164**]). The table below contains the notation and edge matrices of these 10 associative products, and examples can be found in Figure 4.1.

Product	Notation	Edge Matrix
Categorical	$G \times H$	$\begin{pmatrix} E & N & N \\ N & \Delta & N \\ N & N & N \end{pmatrix}$
Co-Categorical	$G \times^c H$	$\begin{pmatrix} E & E & E \\ E & \Delta & E \\ E & E & N \end{pmatrix}$
Cartesian	$G \Box H$	$\begin{pmatrix} N & E & N \\ E & \Delta & N \\ N & N & N \end{pmatrix}$
Co-Cartesian	$G \Box^c H$	$\begin{pmatrix} E & E & E \\ E & \Delta & N \\ E & N & E \end{pmatrix}$
Strong	$G \boxtimes H$	$\begin{pmatrix} E & E & N \\ E & \Delta & N \\ N & N & N \end{pmatrix}$
Disjunction	$G \vee H$	$\begin{pmatrix} E & E & E \\ E & \Delta & N \\ E & N & N \end{pmatrix}$
Equivalence	$G \equiv H$	$\begin{pmatrix} E & E & N \\ E & \Delta & N \\ N & N & E \end{pmatrix}$
Symmetric Difference	$G \bigtriangledown H$	$\begin{pmatrix} N & E & E \\ E & \Delta & N \\ E & N & N \end{pmatrix}$
Lexicographic	$G \bullet H$	$\begin{pmatrix} E & E & E \\ E & \Delta & N \\ N & N & N \end{pmatrix}$

The only two products which are not commutative are self-complementary. They are the lexicographic product and the co-lexicographic product, whose edge matrix is the transpose of that of the lexicographic product. We do not consider this latter product explicitly, since all details can be derived from the results of the lexicographic product. See Figure 4.1 for examples of the product $P_3 \otimes P_3$.



Categorical

Cartesian

Strong

Lexicographic

Co-Lexicographic

Disjunction

Symmetric Difference

Equivalence

Figure 4.1. Graph products with both factors equaling P_3.

All 256 products can be ordered by inclusion; that is, $\oplus \leq \otimes$ if for each pair of graphs G and H, $E(G \oplus H) \subseteq E(G \otimes H)$. The suborder for the products of interest for us is shown in Figure 4.2.

We emphasize that in this chapter, the graphs G we consider are finite and have at least two vertices. The latter assumption is made so as to avoid listing many exceptions in our results.

We first recall some earlier definitions and present some elementary results that can be found in [**166**]. We write $a \simeq b$ if a is either equal or adjacent to b, $a \sim b$ if a is adjacent to but not equal to b, and $a \not\simeq b$ if a is neither adjacent nor equal to b. For ease of notation, we

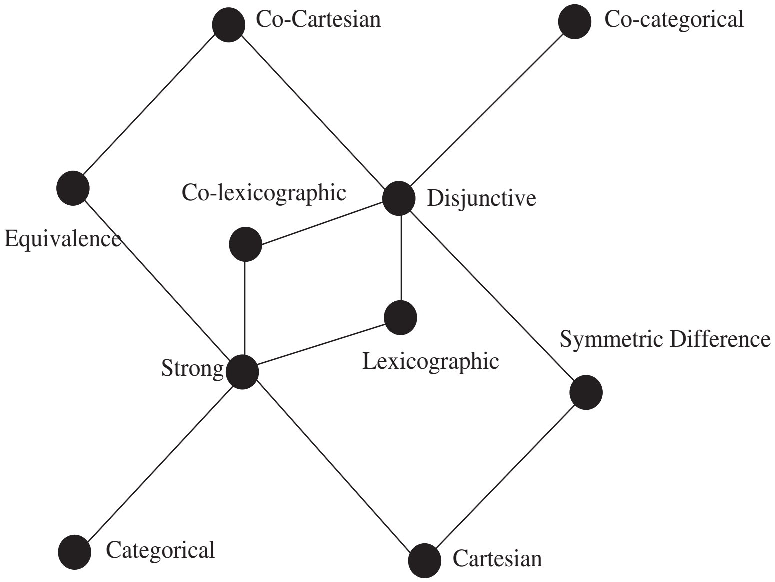


Figure 4.2. An ordering of graph products.

use (ax) to denote a vertex in the product $G \otimes H$, where $a \in V(G)$ and $x \in V(H)$.

4.2. Cop Numbers and Corners in Products

What about the cop numbers of products of graphs? This was first considered in the following theorem of Tošić [**192**] for Cartesian products.

Theorem 4.1 ([**192**]). *For graphs G and H,*

$$c(G \square H) \leq c(G) + c(H).$$

More generally, for graphs $G_1, G_2, \ldots, G_k$, we have that

$$c(\square_{i=1}^{k} G_i) \leq \sum_{i=1}^{k} c(G_i).$$

Maamoun and Meyniel [**144**] found the cop number of the Cartesian product of trees.

Theorem 4.2 ([**144**]). *If $T_1, T_2, \ldots, T_k$ are trees, then*

$$c(\Box_{i=1}^{k} T_i) = \lceil (k+1)/2 \rceil.$$

The examination of Cartesian products continued with Neufeld in [**163**], who considered products of cycles and trees in the following theorem.

Theorem 4.3 ([**163**]). *If $C_1, C_2, \ldots, C_k$ are cycles each with length of at least 4, then*

$$c(\Box_{i=1}^{k} C_i) = k+1.$$

He also proved the following.

Theorem 4.4 ([**163**]). *If $G = \Box_{i=1}^{k} C_i$ and $H = \Box_{i=1}^{j} T_i$, where*

$$C_1, C_2, \ldots, C_k$$

are cycles each with length of at least 4, and $T_1, T_2, \ldots, T_j$ are trees, then

$$c(G \Box H) = c(G) + c(H) - 1 = k + \lceil (j+1)/2 \rceil.$$

Another product which has received attention in this area is the strong product. We saw that the strong product of two cop-win graphs is cop-win. Neufeld and Nowakowski found the following generalized result for the strong product of graphs with arbitrary cop numbers (see Theorem 2.8 parts (1) and (2)).

Theorem 4.5 ([**164**]). *For graphs G and H,*

$$c(G \boxtimes H) \le c(G) + c(H) - 1.$$

Now, what about the seven other products that were introduced earlier? At present, these have received relatively little attention! Even before attempting to estimate their cop numbers, do these products preserve corners? More explicitly, let G and H be graphs and $a \in V(G)$, $x \in V(H)$ be corners. For what graph products $\otimes$ is it true that (ax) is a corner? Let us break that down even further. The corner a has a vertex $v_a \neq a$ such that $N[a] \subseteq N[v_a]$. We will call v_a a *corner-dominating* vertex for a or a *cord.* (Intuitively, v_a cordons or ropes off the corner. We also say that v_a *covers* a.) Similarly, the corner x has a cord $v_x \neq x$ such that $N[x] \subseteq N[v_x]$. When is $(v_a v_x)$ a cord for (ax)? First of all, $(v_a v_x)$ must be adjacent to (ax), therefore, the 3×3 matrix, which defines the product, must have an E in the (E, E) cell. This eliminates the Cartesian and symmetric difference products.

(1) Suppose $(E, E) = E$; that is, if $b \sim a$ and $y \sim x$, then $(by) \sim (ax)$. If $c \sim a$, then $(cv_x) \sim (ax)$, therefore we must have $(v_a v_x) \sim (cv_x)$ and hence $(E, \Delta) = E$. Similarly, we also have $(\Delta, E) = E$.

(2) Suppose $(E, \Delta) = E$; that is, if $b \sim a$, then $(bx) \sim (ax)$. This forces $(bx) \sim (v_a v_x)$; that is, $(E, E) = E$.

(3) Suppose $(N, \Delta) = E$; that is, $b \not\sim a$ implies $(bx) \sim (ax)$. But then since b could be either adjacent or non-adjacent to v_a, we must have $(v_a v_x) \sim (bx)$ which forces $(N, E) = E$ and $(E, E) = E$. Now by (1), this forces $(E, \Delta) = E = (\Delta, E)$. Similarly, $(N, \Delta) = E$ forces $(E, N) = (E, E) = (E, \Delta) = (\Delta, E) = E$.

(4) Suppose $(E, N) = E$; that is, if $b \sim a$ and $y \not\sim x$, then $(by) \sim (ax)$. This forces $(v_a v_x) \sim (by)$ and therefore, that $(E, E) = E$. Similarly, $(N, E) = E$ forces $(E, E) = E$ and consequently $(E, \Delta) = E = (\Delta, E)$.

(5) Suppose $(N, N) = E$; that is, if $b \not\sim a$ and $y \not\sim x$, then $(by) \sim (ax)$. Since b and y might or might not be adjacent to v_a and v_x, respectively, we must have $(E, N) = (N, E) = E$. Also, we must have $(E, E) = E$ and consequently $(E, \Delta) = E = (\Delta, E)$.

Summing up, by (1) to (5) above, the product of corners remains a corner just if $\otimes$ is the strong, lexicographic, co-Cartesian, co-categorical, or disjunctive product. Of course, corners remaining corners does not guarantee that the $G \otimes H$ will be cop-win since after the removal of a corner in $G \otimes H$ the remaining graph is no longer a product of graphs. The more general question in this direction is the following. *Characterize the graphs G and H and products such that $G \otimes H$ is cop-win.* This problem is wide open and we leave the reader to generate conjectures and results.

4.3. Covering by Cop-win Graphs

One approach to bounding the cop number of a graph G is to find induced subgraphs G_i, $i = 1, 2, \ldots, k$ such that $V(G) = \bigcup_{i=1}^{k} V(G_i)$ such that each G_i is both a cop-win graph and a retract of G. We will call our special cover a *retract-cover.* Given a retract-cover, we need only put one cop on each subgraph. Eventually, the cop in each subgraph captures the shadow of the robber, and since the robber must be in one of the subgraphs he is caught. This is the idea of the precinct or beat number (see Theorem 1.7 and Exercise 22 in Section 1.6). It is taking us too far away from the focus of the book to include the proofs of these so we present them in an abbreviated fashion. Throughout this section, $\otimes$ will represent a product and

$$G_{\otimes}^{n} = \bigotimes_{i=1}^{n} G;$$

that is, the product of G with itself n times.

4.3.1. Isometric Paths. Covering with paths seems relatively easy but generally gives a very high bound. Clarke [**52**] obtained many results. Recall from Chapter 1, Exercise 22, that the *isometric path number* (or *precinct number*) of G, written $p(G)$, is the minimum number of isometric paths (or beats) needed to cover G. Let $|V(G)| = v$ and define

$$\rho(G, \otimes) = \lim_{n\to\infty} \frac{p(G_{\otimes}^{n})}{v^{n}},$$

provided the limit exists. It is not certain that the limit exists for the lexicographic or categorical products, although she conjectures that the limit does exist in those cases. For the other products, Clarke was able to either determine $\rho(G, \otimes)$ exactly or show that

$$p(G_{\otimes}^{n+1}) < vp(G_{\otimes}^{n}),$$

thereby proving the limit exists.

In [**79**], Fitzpatrick gave the following lower bound.

Theorem 4.6 ([**79**])**.** *If G is a graph of order n, then*

$$\rho(G, \otimes) \geq \frac{1}{\operatorname{diam}(G_{\otimes}^{n}) + 1}.$$

Clarke [**52**] found the following bounds. For the purposes of this table, we will assume that *the graphs have at least two vertices each and are not complete graphs.* The results may be different if any one of these conditions is not met. In addition, we set $|V(G)| = n$.

$\otimes$	$\mathrm{diam}(G \otimes H)$	$\rho(G, \otimes)$
$\Box$	$= \mathrm{diam}(G) + \mathrm{diam}(H)$	unknown
$\times$	$\geq \max\{\mathrm{diam}(G), \mathrm{diam}(H)\}$ $\leq 2\mathrm{diam}(G)$ if $H = G$	$\leq \frac{2p(G)}{n}$ $\geq \frac{1}{2\mathrm{diam}(G)+1}$ if $\chi(G) \geq 3$
$\boxtimes$	$= \max\{\mathrm{diam}(G), \mathrm{diam}(H)\}$	$= \frac{1}{\mathrm{diam}(G)+1}$
$\bigtriangledown$	≤ 2	$= 1/3$
$\bullet$	$= \mathrm{diam}(G)$	unknown
$\equiv$	≤ 2 if $\gamma(G) = \gamma(H) = 1$ ≤ 3 if both $\gamma(G),\ \gamma(H) > 1$ $\leq \max\{\mathrm{diam}(G), \mathrm{diam}(H)\}$	$= 1/3$ if $\mathrm{diam}(G) = 2$ $= 1/4$ otherwise
$\vee$	≤ 2	$= 1/3$
$\Box^c$	≤ 2	$= 1/3$
$\times^c$	≤ 2	$= 1/2$

We now include a few brief comments about some of the interesting cases.

Cartesian Product. Under this product, the first set of graphs that come to mind are the hypercubes, $Q_n = (K_2)^n_\Box$. Fitzpatrick et al. [**81**] showed that

$$\frac{2^n}{n+1} \leq p(Q_n) \leq \frac{3 \cdot 2^{n-1}}{n+1},$$

and that if $n = 2^m - 1$, then the lower inequality is the exact value. They think that the upper bound is far from best possible. Using this result, Clarke shows the following.

Theorem 4.7 ([**52**]). *For a graph G with a perfect matching, we have that $\rho(G, \Box) = 0$.*

An open problem is to improve the upper bound for $p(Q_n)$. Another open problem is to determine if for all graphs G, whether $\rho(G, \Box) = 0$.

Categorical Product. For a graph G, let m be the size of a *maximum matching* in G; that is, a set of pairwise disjoint edges, which is the largest size of all such sets of edges.

Lemma 4.8 ([**52**]). *Let G be a graph with $n > 1$ vertices. Then*

$$p(G \times G) \leq p(G)(2n - 2m).$$

Theorem 4.9 ([**52**]). *If G is a graph with n vertices, then $\rho(G, \times) < \frac{2p(G)}{n}$.*

Clarke also gives the following examples:

$$\frac{1}{2n+1} \leq \rho(P_n, \times) \leq \frac{2}{n+1},$$

$$\frac{1}{3} \leq \rho(K_{2n}, \times) \leq 1,$$

and

$$\frac{1}{2n+1} \leq \rho(C_{2n+1}, \times) \leq \frac{4}{2n+1}.$$

An open problem is to find the exact values for any of these families of graphs.

Strong Product. In this case, we have the following result.

Theorem 4.10 ([**52**]). *For G a graph,*

$$\rho(G, \boxtimes) = \frac{1}{\operatorname{diam}(G) + 1}.$$

The lemma required to prove this is the following.

Lemma 4.11 ([**52**]). *Let G be a graph of order n with an isometric path cover that consists of $m(G) > 0$ paths of length* $\operatorname{diam}(G)$ *and k isolated vertices. Then we have that*

$$\rho(G, \otimes) \geq \frac{m(G)}{n-k}.$$

Symmetric Difference. Suppose G is a complete graph on n vertices. Clarke [**52**] showed that

$$1/3 \leq \rho(G, \triangledown) \leq 1/2.$$

She also showed that

$$\frac{p(G_{\triangledown}^{k+1})}{n^{k+1}} \leq \frac{p(G_{\triangledown}^{k})}{n^{k}},$$

which proves that $\rho(G, \triangledown)$ exists. This leaves us with the unsolved problem of finding $\rho(K_n)$.

Lexicographic Product. There is a *fractional* version of the covering by isometric paths which we discuss now. Given a graph G, let $\mathcal{P}$ be the set of isometric paths in G. Define $\mathbb{R}^{\geq 0}$ to be the set of all non-negative real numbers. Let $w : \mathcal{P} \to \mathbb{R}^{\geq 0}$ be a function that assigns to each path in $\mathcal{P}$ a non-negative weight subject to the constraint that for every vertex in G, the sum of the weights of all the isometric paths that contain it is at least one. Call such a weighting w a *fractional cover* and let $W_w = \sum_{P \in \mathcal{P}} w(P)$. Let

$$p_f = \min\{W_w : w \text{ is a fractional cover}\}.$$

Note that the original problem of covering by isometric paths is equivalent to only allowing weights of 0 and 1. The fractional cover problem is the so-called LP relaxation of the $0 - 1$ problem so that $p_f(G) \leq p(G)$ (for more background on this terminology and fractional graph theory, see [**184**]). For example, a three-claw $G = K_{1,3}$ (see Figure 4.3) has $p(G) = 2$ but $p_f(G) = 3/2$.

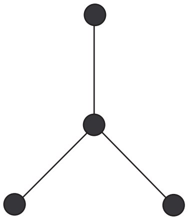


Figure 4.3. The three-claw.

Theorem 4.12 ([**52**]). *If G is a graph on n vertices, then*

$$\rho(G, \bullet) \leq \frac{p_f(G)}{n}.$$

As a consequence of Theorem 4.12, Clarke shows that $\rho(G, \bullet) \leq \frac{3}{8}$, assuming that $\rho(G, \bullet)$ exists.

4.3.2. Cliques. A clique is always a retract and is a cop-win subgraph, so a clique cover would be a retract-cover. The minimum number of cliques needed to cover G is denoted $\theta(G)$. Unfortunately, finding the least number of cliques in a clique cover is an **NP**-complete problem (for background on **NP**-complete problems, we direct the reader to Chapter 5). For example, a clique cover of a C_n would give a upper bound of $n/2$ for the cop number whereas it only takes two isometric paths to cover the cycle. The strong product of P_n with itself is cop-win but there is no clique bigger than K_4 in this product, and the longest isometric path has n vertices so the upper bounds are $n^2/4$ and n, respectively.

The hypercube Q_n never has any cliques bigger than K_2, so that in the product $\theta(Q_n)$ grows large. Does this happen in the other products? As in the previous section, with G being a graph on n vertices, let us introduce a new normalized parameter

$$\rho'(G,\otimes) = \lim_{k\to\infty} \frac{\theta(G^k_{\otimes})}{n^k}.$$

Unfortunately, this turns out to be almost useless since if $\otimes$ is anything other than the Cartesian, categorical, or symmetric difference product, then $\rho'(G,\otimes)=0$. This follows since for all but these three products, $K_n \otimes K_m = K_{mn}$.

It should be noted that $\theta(G) = \chi(\overline{G})$. At the time of writing, there were over 200 articles found in *MathSciNet* when “product” and “chromatic number” were used as search terms and restricted to MSC Primary “05”. We leave it to the reader to survey this extensive literature.

4.3.3. Domination Considerations. The minimum number of cop-win graphs required to form a retract-cover of the vertices of a graph G is called the *cop-cover number* of G and will be denoted $cc(G)$. This was called the *cop-win number* in [**52**] but the name could be confused with the cop number.

The domination number, written $\gamma(G)$, can be re-interpreted in terms of retract-covers, with the subgraphs being the closed neighborhoods of a set of vertices. Immediately, we have that

$$c(G) \le cc(G) \le \min\{\gamma(G), p(G), \theta(G)\}.$$

For products, most of the results below are taken from [**52**], and are probably far from the best possible. A starting point is the following result.

Theorem 4.13 ([**166**]). *If $\otimes$ is greater or equal the strong product in the partial order, and if A and B are dominating sets in G and H, respectively, then $A \times B$ is a dominating set of $G \otimes H$.*

As an immediate corollary we have the following.

Corollary 4.14 ([**166**]). *If $\otimes$ is greater or equal the strong product in the partial order, then*

$$cc(G \otimes H) \leq \gamma(G)\gamma(H).$$

This leaves the categorical, Cartesian, and symmetric difference products. Indeed, a hard open problem is the famous *Vizing's Conjecture:*

$$\gamma(G \Box H) \geq \gamma(G)\gamma(H);$$

see [**110, 195**].

Strong Product. Since the strong product of two cop-win graphs is cop-win, an almost trivial result is that

$$cc(\boxtimes_{i=1}^{k} G_i) \leq \prod_{i=1}^{k} cc(G_i).$$

Lexicographic Product. A little thought gives the following result.

Theorem 4.15 ([**52**]). *If G is cop-win and $H = K_{1,n}$, then $G \bullet H$ is cop-win.*

This can be used to then prove the following.

Theorem 4.16 ([**52**]). *If G and H are graphs, then*

$$cc(G \bullet H) < cc(G)\gamma(H).$$

A special subset is useful for the next products; we find sets A and B such that $A \times B$ dominates $G \otimes H$.

Categorical Product. A *total dominating set* T has the property that every vertex in G is adjacent to a vertex in T, including vertices of T. The minimum cardinality of such sets is denoted by $\gamma_t(G)$. Note that a total dominating set is dominating, so $\gamma(G) \leq \gamma_t(G)$. However, dominating sets need not be total dominating.

Theorem 4.17 ([**52**]). *Let G and H be graphs. Then*

$$cc(G \times H) \leq \gamma_t(G)\gamma_t(H).$$

Equivalence Product. Given a graph G, let

$$u(G) = \min\{|A| : \text{for all } v \in V(G), A \not\subseteq N[v]\}.$$

For any graph of diameter 3 or more, we have that $u(G) = 2$.

Theorem 4.18 ([**52**]). *Let G be a graph with $\gamma(G) > 1$. Then*

$$cc(G \equiv G) \leq u(G)^2.$$

Moreover, if $\operatorname{diam}(G) \geq 3$, *then* $cc(G \equiv G) \leq 4$.

Symmetric Difference. Let $\gamma'(G)$ be the smallest cardinality of a set of vertices that dominate both G and G^c.

Theorem 4.19 ([**52**]). *Let G and H be graphs. Then*

$$cc(G \bigtriangledown H) \leq \min\{\gamma(G)\gamma'(H), \gamma'(G)\gamma(H)\}.$$

Theorem 4.20 ([**52**]). *Let G and H be graphs. Then*

$$cc(G \bigtriangledown H) \leq \min\{\gamma_t(G)\gamma_t(H^c), \gamma_t(G^c)\gamma_t(H)\}.$$

Co-Cartesian. The special set A in this case is an independent set of cardinality 2.

Theorem 4.21 ([**52**]). *Let G be a graph. Then $cc(G\square^c G) \leq 4$.*

4.4. Genus of a Graph

We now consider graphs living in a specified class. We focus on graphs defined on surfaces.

One of the main intuitive approaches in a winning strategy for the cops is for the cops to continually reduce the space that the robber has to move. We now make this more formal. *Guarded vertices* are those that if a robber moved through he would be captured by a cop. The set of guarded vertices of G is its *cops' territory*. In the proofs below, we will be conservative and define a boundary that the cops control similar to a cop guarding an isometric path in Theorem 1.7, with the cop territory on one side and the robber on the other. There could be other vertices that the cops could prevent the robber getting to but in the upcoming proofs the boundary is moved incrementally. If H is the cop territory, then the *robber territory* is the set of vertices in the component of $G - H$ containing the robber. The *unguarded territory* is the set of vertices not in the cop territory. Hence, the robber territory is a subset of the unguarded territory, and may be a proper subset.

In many arguments it turns out that we only need to know where the robber cannot be and expand that region. Any argument that uses this approach is stronger than we actually need since the cops do not require perfect information. Suppose we are in a pursuit game on a graph G where the following properties are satisfied.

(1) The cops on their turn can move along edges.

(2) The robber slides along an edge or passes on his turn.

(3) The cops will capture the robber if a cop and robber occupy the same vertex at any time.

(4) Some oracle tells them where the shadow of the robber is on a subgraph.

Apart from (4), this is reminiscent of searching, sweeping, or decontaminating a graph; see [**7**] and Chapter 9. In those cases though, the intruder (or chemical or biological contamination) is infinitely fast. The proofs for the cop numbers of outerplanar and planar graphs require only this amount of information.

Most surfaces mathematicians work with are the sphere, the doughnut (a sphere with a hole), the pretzel (a sphere with two holes), and so on. Another way of thinking about these is to think of the sphere and add handles. The *genus* of a graph G is the smallest k such that G can be drawn on a sphere with k handles so that distinct edges do not intersect except at common vertices.

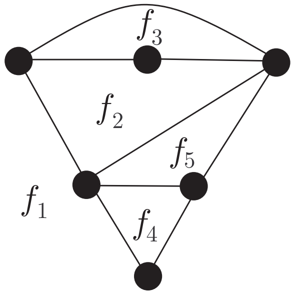


Figure 4.4. A planar graph and its faces. The outer face is f_1.

The planar case (that is, graphs with genus 0) is always used for intuition for surfaces with higher genus, one reason being that locally all the surfaces seem planar. Given an embedding of a planar graph, a *face* is a maximal region X with the property that for all $u, v \in X$, u and v can be joined by a curve which does not touch an edge of the embedding. Every planar graph includes an infinite or *outer* face, which surrounds the graph. See Figure 4.4. One famous result is *Euler's Planar Polytope Formula* which we will translate in graph terms.

Theorem 4.22. *For a fixed planar embedding of a graph G*

$$v - e + f = 2,$$

where $v = |V(G)|$, $e = |E(G)|$, and f is the number of faces in the embedding.

The proof of Theorem 4.22 is rather easy: start with a spanning tree, then notice that

$$v - e + f = v - (v - 1) + 1 = 2.$$

Any edge that is now added leaves v alone, increases e by one and splits a face in two so that f is increased by one.

For genus k surfaces, the formula can be generalized to the *Euler-Poincaré formula*:

$$v - e + f = 2 - 2k.$$

One can imagine a similar proof except one has to determine what the subgraph equivalent to a spanning tree might be. As we shall see, a similar case occurs for Cops and Robbers. Theorem 1.7, which states that an isometric path is 1-guardable, will be extended in Corollary 4.26, to guarding a cycle which is the boundary of a planar region. In the plane, the robber must then be inside or outside of the cycle. Going to higher genus, again, what is the analogue of the cycle?

4.5. Outerplanar Graphs

Before we tackle the cop number of planar graphs, we consider the simpler outerplanar case. A graph G is *outerplanar* if it has an embedding in the plane with the following properties.

(1) Every vertex lies on a circle.

(2) Every edge of G either joins two consecutive vertices around the circle or is a chord across the circle.

(3) If two chords intersect, then they do so at a vertex.

Often the edges are drawn on the "outside" of the circle of vertices but it is equivalent to have them on the inside. We will label the vertices clockwise around the circle $v_0, v_1, \ldots, v_{n-1}$.

Nancy Clarke proved the next result in her doctoral thesis.

Theorem 4.23 ([**52**]). *If G be an outerplanar graph, then $c(G) \leq 2$.*

Since a cycle is an outerplanar graph, not all outerplanar graphs are cop-win, so we need an algorithm to show that two cops suffice. We note that if G is *maximal outerplanar* (that is, has the maximum number of edges so no new circle edge or chord can be added), then the G is cop-win; see Exercise 22 in this chapter.

Proof of Theorem 4.23. Assume first that G has no cut vertices. Suppose that for a given i that v_i is not adjacent to v_{i+1}. We can renumber the subscripts so that $i = 0$. Since G is connected and the degree of v_0 is at least two, then let v_j be the vertex of least index which is adjacent to v_0. The edge v_0v_j prevents any vertex in $\{v_1, v_2, \ldots, v_{j-1}\}$ from being adjacent to any vertex of

$$\{v_{j+1}, v_{j+2}, \ldots, v_{n-1}\},$$

and no vertex of $\{v_1, v_2, \ldots, v_{j-1}\}$ is adjacent to v_0. Therefore, v_j is a cut vertex, which is a contradiction. Hence, we may assume that for all i, v_i is adjacent to both v_{i-1} and v_{i+1}, with subscripts taken modulo n.

If the embedding contains no chords, then it is a cycle and two cops suffice to capture the robber. Let $a_0, a_1, \ldots, a_k$ be the vertices of degree at least 3 in order around the circle. Note that vertices on the cycle between a_i and a_{i+1} are of degree 2 and so *the path* between a_i and a_{i+1} is well defined.

Place the cops C_1 and C_2 on the vertex v_0. If v_0 has degree 2, then it is on a path between a_0 and a_k (renumbering the a_i if necessary); if it has degree 3 or more, then renumber so that $a_0 = v_0$. In either case, we can move C_1 to a_0 and C_2 to a_k so that the robber is not on the path between these two vertices.

We now assume more generally that C_1 and C_2 are on a_i and a_j for some $i < j$, respectively, and that the robber is not on and cannot move to any vertex in $\{v_p, v_{p+1}, \ldots, v_0, \ldots, v_q\}$, where $v_p = a_j$ and $v_q = a_i$. That is, every path from the robber to a vertex in $\{v_p, v_{p+1}, \ldots, v_0, \ldots, v_q\}$ passes through a_i or a_j. Such an area is the cop territory. The idea of the proof now is to show that the cops can increase the cop territory, so that it is eventually all of G and the cops win.

Suppose a_i has a chord to a vertex in the robber territory. Let a_r be a vertex adjacent to a_i which is closest to a_j. If the robber is on the arc of the circle from a_i to a_r, then he cannot move off that arc if C_1 on a_i does not move. Therefore, C_2 can be moved to a_r and the cop territory has increased. If the robber is between a_j and a_r, then C_1 moves from a_i to a_r again increasing the cop territory. A similar analysis holds for a_j. Hence, the only case to consider is when neither a_i nor a_j have an interior edge to the robber territory. In this case, the only paths to the cop territory from the robber are the ones along the cycle incident to a_i and a_j. That is, every path from the robber to the cop territory passes through a_{i+1} or a_j. Hence, moving C_1 along the path from a_i to a_{i+1} does not allow the robber to move into the cop territory, and the cop territory has increased. We will refer to this as the *no-cut-vertex strategy.*

Now suppose that G has at least one cut vertex. Let

$$B(G) = \{G_1, G_2, \ldots, G_m\}$$

be the set of maximal induced subgraphs of G such that each G_i itself has no cut vertices. Note that each G_i will contain a vertex which is a cut vertex of G, and each G_i has at least two vertices. For example, see Figure 4.5.

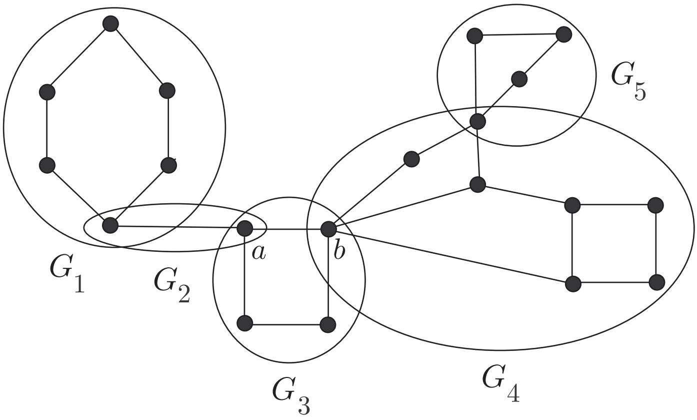


Figure 4.5. The induced subgraphs G_i in an outerplanar graph with cut vertices.

We can retract G onto G_i, for any i, by the mapping described as follows. Let $x \in V(G_i)$ and x be a cut vertex of G. All vertices of G that are disconnected from G_i by the deletion of x are mapped to x. Vertices of G_i are mapped to themselves. (Consider G_3 in Figure 4.5. Vertices of G_1 and G_2 are mapped to a and those of G_4 and G_5 are mapped to b.) Let $\widehat{G_i}$ denote this retract. Since G_i is a subgraph of an outerplanar graph, $\widehat{G_i}$ is also outerplanar. Fix an embedding for each G_i.

Choose some G_i and place the two cops on two vertices in G_i as in the case with no cut vertices. We now use the shadow strategy on $\widehat{G_i}$. Employing the strategy of the case with no cut vertices, the cops will capture the robber's image on $\widehat{G_i}$. Since $|V(G_i)| \geq 2$, at least one more vertex and one more element of $B(G)$ is added to the cop territory. If the robber is actually on G_i, then he has been caught. If not, then the cops have captured the robber's shadow on a vertex x whose deletion separates G_i from the G_k where the robber presently resides. This cut vertex also lies in some G_j that either contains the robber (that is, $j = k$) or contains a cut vertex $y \neq x$ whose deletion separates G_j from G_k (and j is unique). Fix an outerplanar embedding of $\widehat{G_j}$. The cops now execute the no-cut-vertex strategy on $\widehat{G_j}$. Hence, the cops eliminate the subgraphs in $B(G)$, and eventually they capture the robber (rather than just his image). □

We note that the elements of $B(G)$ are called *blocks*, and the decomposition of G into the G_i gives a tree of order m (the vertices are blocks, and blocks are adjacent if they share a cut vertex). A similar approach to bounding the cop number of a general (non-outerplanar) graph was given in [**113**]; see also Exercise 26. We also observe that the converse of Theorem 4.23 does not hold in general, as Figure 4.6 demonstrates.

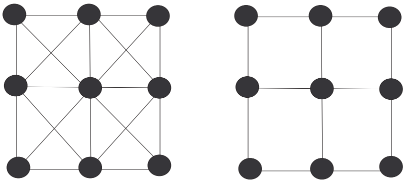


Figure 4.6. The graph on the left is a cop-win non-outerplanar graph, while the graph on the right is non-outerplanar with cop number 2.

4.6. Planar Graphs

Planar graphs have inspired some of the deepest results in graph theory, most notably the Four Color Theorem (which states that every planar graph has chromatic number at most 4; see [**13**] and [**183**]). A graph is *planar* if it can be drawn in $\mathbb{R}^2$ without any two of its edges crossing. For example, a cycle is planar, and so is K_4. The graph K_5 is not planar (see Exercise 21b); nor is $K_{3,3}$. As Kuratowski proved in [**130**], K_5 and $K_{3,3}$ are, in a certain sense, the only obstructions to being non-planar. We *subdivide* an edge uv by replacing it by a two-path uxv, where x is a new vertex of degree 2. A *subdivision* of a graph results by subdividing some subset of its edges.

Theorem 4.24 ([**130**]). *A graph is planar if and only if it does not contain a subgraph which is a subdivision of K_5 or $K_{3,3}$.*

At first glance, there is no reason to believe that a fixed constant number of cops can guard every planar graph. However, Aigner and Fromme [**2**] showed in fact that planar graphs require no more than three cops (and that some actually require three). The proof of this fact, as we will see below, is much more involved than in the outerplanar case. At the time of writing, no one has characterized which planar graphs are cop-win, or those which require two cops and which require three. As we proved in Theorem 4.23, outerplanar graphs require no more than two (and some require two). From Chapter 3 Exercise 13, all graphs of strong isometric dimension 2 require no more than two cops, but there such graphs that are planar but not outerplanar.

To be clear, we make the distinction between a *planar* graph and a *plane* graph, which is an actual planar embedding of a planar graph.

Theorem 4.25 ([**2**]). *If G is a planar graph, then $c(G) \leq 3$.*

The very first step in the proof is fraught with danger if your intuition is not primed correctly. "Choose two vertices of maximum distance from each other, and let P be a shortest path joining them." Clearly, this path partitions the rest of the graph into at least two connected components. That would be false as Figure 4.7 shows. One has to be careful about the assumptions one has when the terms "shortest path" and "planar" are used in the same sentence.

Our proof of Theorem 4.25 follows closely the proof given by Alspach, Li, and Yang [**9**]. From Theorem 1.7 we know that an isometric path is 1-guardable. The extension to a special cycle forms the basis of the proof of Theorem 4.25.

Corollary 4.26 ([**2**]). *Let v, w be distinct vertices of a graph G. Let P_1 and P_2 be two internally disjoint paths from v to w, let P_1 be*

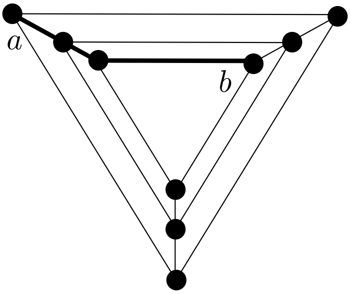


Figure 4.7. Warning: Graph distance may be different than Euclidean distance.

isometric in G, *and let* P_2 *be isometric in* $G-(V(P_1)\setminus\{v,w\})$. *Then* $P_1\cup P_2$ *is* 2*-guardable in* G.

Proof. By Theorem 1.7, we may move one cop named C_1 to guard P_1. Similarly, in $G-(V(P_1)\setminus\{v,w\})$ we may move the second cop, named C_2, to guard P_2. In G the only possible way for the robber to enter P_2 without being captured by C_2 is to move through P_1; but then the robber will have been captured by C_1. □

Before we prove Theorem 4.25, we will need some terminology. Let X be a cycle in a planar graph G, and let u be a vertex of $G-X$. The cycle X partitions the plane into two regions: A_1 containing u and A_2 which does not. Let V_1 denote the vertices contained in A_1, called the *inside* of X with respect to u, and V_2 those in A_2 which we call the *outside* of X with respect to u. The subgraph induced by $V(X)\cup V_1$ is named the *internal subgraph determined by* X, and written $\mathrm{int}(X)$. The subgraph induced by $V(X)\cup V_2$ is named the *external subgraph determined by* X, and written $\mathrm{ext}(X)$. We note that the only way for the robber to pass from the inside to the outside (or vice versa), is to pass through a vertex of the cycle X.

Proof of Theorem 4.25. We proceed by showing that with a finite number of moves, the cops can increase the cop territory. Hence, the robber territory eventually is reduced to the empty set, and so the robber is captured. There are three cases that arise, and in each case, H is the unguarded territory.

(I) Some cop is guarding a shortest path P of a subgraph H of G, and any path from the robber to the cop territory is through a vertex of P.

(II) Two cops guard $P_1 \cup P_2$, where P_1 and P_2 are internally disjoint paths joining the same two vertices, and any path from the robber to the cop territory is through a vertex of $P_1 \cup P_2$. The subgraph H is either the internal or external region of the cycle $P_1 \cup P_2$ (whichever region contains the robber).

(III) Some cop is on a cut vertex v of a subgraph H of G, and any path from the robber to the cop territory is through v.

In some cases we give explicit proofs about the robber's access to the cop territory. We leave it as an exercise for the reader to fill in the details in the other cases (see Exercise 23).

We may assume G is not complete. Let C_1, C_2, and C_3 be the three cops. Choose two vertices of maximum distance from each other, and let P be a shortest path joining them. We first move C_1 to guard P. Thus, after a finite number of moves, we are in case (I) with $H = G$. In particular, one of the three cases has arisen! So the initial strategy of the cops is to first enforce case (I). We then show that this case leads to one of the three cases, but with larger cop territory. We then show that the analogous situation happens when we are in case (II) or case (III), which will complete the proof. (More precisely, we proceed by induction on the order of the cop territory.)

Suppose we are in an instance of case (I). Hence, there is a subgraph H, and there is a cop guarding a shortest path in H given by

$$P = P_1 = v_1 v_2 \cdots v_k,$$

with $k \geq 2$. Without loss of generality, let C_1 be the cop guarding P_1, thus, forcing the robber to remain in a component of $H - P$.

Let Y denote the component of H containing the robber. If there is a unique $v_i \in P_1$ adjacent to some vertex of Y, then v_i is a cut vertex of H. We can move C_1 to v_i. Since C_1 is closer to v_i than the robber, the robber cannot escape from Y. Hence, we are in case (III) (and not case (I)). We then move C_2 to v_i preventing the robber from

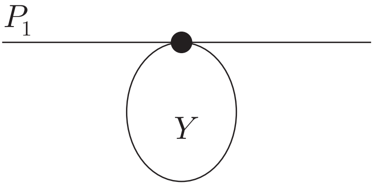


Figure 4.8. A unique vertex v_i adjacent to Y.

leaving Y and freeing the other two cops to move; see Figure 4.8. Now suppose v_i and v_j, $i < j$, have neighbors in Y such that if v_r is another vertex of P_1 with neighbors in Y, then $i < r < j$. Let u_1 be a neighbor of v_i in Y, and let u_2 be a neighbor of v_j in Y. Let P_2 be a shortest path in Y from u_1 to u_2; see Figure 4.9. Move C_2 to guard P_2 in Y, and note that C_1 still guards the subpath P' connecting v_i to v_j in P_1. The robber is either in the internal or external region bounded by the cycle formed by $P' \cup P_2 \cup \{v_iu_1, v_ju_2\}$; in both cases, we are in case (II). The cop territory has become larger even if $u_1 = u_2$.

Now suppose we are in case (II). Without loss of generality, we may assume the robber is located in $\mathrm{int}(X)$, where $X = P_1 \cup P_2$, and the subgraph $\mathrm{ext}(X)$ is guarded. Now H is the subgraph induced by G on $V(\mathrm{int}(X)) \setminus V(X)$. The robber territory is in a component Y of H containing the robber.

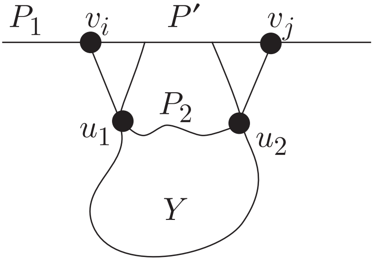


Figure 4.9. At least two vertices of P_1 adjacent to vertices in Y.

If there is only one vertex u of X with a neighbor in Y, then u is a cut vertex of $\text{int}(X)$. We move the free cop to u putting us in case (III) (and not case (II)).

Suppose that both P_1 and P_2 have exactly one vertex each, v_1 and v_2, respectively, with neighbors in Y, so that we are not in the preceding subcase of a cut vertex. Consider the subgraph K, where $V(K) = V(Y) \cup \{v_1, v_2\}$, and

$$E(K) = E(Y) \cup \{v_i u : u \in V(Y),\ v_i u \in E(G) \quad \text{and} \quad i = 1, 2\}.$$

Let P be a shortest path from v_1 to v_2 in K. Since v_1 and v_2 are not adjacent in K, then we have that P includes a vertex of Y. See Figure 4.10. Now move the free cop to guard P. If the robber tries to reach v_1 or v_2, then he will be captured. The cop territory has increased and we are in case (I).

We now consider the case that one of the paths has two or more vertices with neighbors in Y. Without loss of generality, we assume $v_1, v_2 \in V(P_1)$ have neighbors in Y, and any other vertices of P_1 with neighbors in Y lie in the subpath Q of P_1 connecting v_1 and v_2. Analogous to the previous case (with the subgraph K), let P_3 be a shortest path from v_1 to v_2 in Y containing a vertex of Y. Let v_1 be joined to $u_1 \in V(Y)$ on P_3, and v_2 be joined to $u_2 \in V(Y)$ on P_3; see Figure 4.11. Move the free cop so that P_3 is guarded in Y. If the robber is located in the region bounded by $V(P_3) \cup V(Q) \cup \{v_1 u_1, v_2 u_2\}$, then the cop who was guarding P_2 can move to guard Q. We now are in case (II) with the cop who was guarding P_1 free to move.

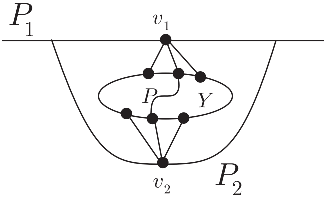


Figure 4.10. The path P in containing a vertex of Y.

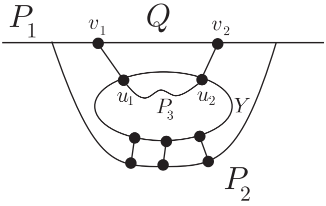


Figure 4.11. The path P_3 in Y.

If the robber is not in the region bounded by $P_3 \cup Q \cup \{v_1u_1, v_2u_2\}$, then he is in the region bounded by P_2, P_3, and the subpaths of P_1 obtained by removing Q. The cop C_2 guards $P_3 - \{v_1, v_2\}$ and now also P_3, since no path in Y is shorter. Define P' to be P_3 along with P_1 minus Q. Thus, we are in case (II) with the paths P_2 and P'. The cop who was guarding P_1 is now free to move. In the last two cases above, the cop territory has increased after reaching the new case. We have now shown that from case (II) we can reduce to one of the three cases with smaller robber territory.

Finally, suppose we are in case (III). Once this case is settled, the proof of the theorem will follow. Let Y be the component of $H - v$ containing the robber. Let u be a vertex of Y which is the largest distance to v (inside Y). Let P be the shortest path from u to v in Y. Now move a free cop to guard P. Note that the robber can never get back into the already guarded territory. With $Y = H$, we are in case (I), and the cop territory has increased. □

The bound in Theorem 4.25 is sharp; we leave it to the reader to show that the dodecahedron needs three cops (see Chapter 1, Exercise 4c).

4.6.1. Graphs of Higher Genus. Less is known about the cop number of graphs with positive genus. As such, our survey of such graphs is brief. The main conjecture in this area is due to Schroeder. In [**185**], Schroeder conjectured that if G is a graph of genus g, then $c(G) \leq g + 3$. Quilliot [**171**] had shown the following.

Theorem 4.27 ([**171**])**.** *If G is a graph of genus g, then $c(G) \leq 2g + 3$.*

In the same paper as the conjecture, Schroeder shows the following.

Theorem 4.28 ([**185**])**.** *If G is a graph of genus g, then*

$$c(G) \leq \left\lfloor \frac{3g}{2} \right\rfloor + 3.$$

A technique that he uses in [**185**] is to partition the embedded graph into planar subregions bounded by two paths and two curves. This is called a *planar trap.* Using Theorem 4.25 he shows that four cops are enough to guard the region. One approach would be to partition the graph into the appropriate regions. This is a bad use of resources since cops remain in a planar trap when they could be free to help in other regions. Schroeder generalized the idea of a planar-trap to now include holes so that cops can be reused. Along the way, he shows the following theorem.

Theorem 4.29 ([**185**])**.** *If G is a graph that can be embedded on a torus, then $c(G) \leq 4$.*

Exercises

1. For each of the associative products $\otimes$ described in the introduction of this chapter, draw the graph $K_2 \otimes K_2$.
2. Draw the products $P_3 \square^c P_3$ and $P_3 \equiv P_3$ by drawing their non-edges.
3. Show that $\max\{c(G), c(H)\} \leq c(G \square H)$.
4. Suppose G and H are cop-win graphs with dismantling sequences $(a_1, a_2, \ldots, a_m)$ and $(x_1, x_2, \ldots, x_n)$, respectively. For each of $\otimes$ being the strong, lexicographic, co-lexicographic, co-categorical, or disjunctive products, prove or disprove that either (or both)

$$((a_1x_1), (a_2x_1), \ldots, (a_mx_1), (a_1x_2), \ldots, (a_mx_n))$$

and

$$((a_1x_1), (a_1x_2), \ldots, (a_1x_n), (a_2x_1), \ldots, (a_mx_n))$$

is a dismantling sequence.

5. Give an example showing that the bound in Theorem 4.5 is sharp.
6. A graph is *prime* with respect to a given graph product if it cannot be represented as the product of two nontrivial graphs.
   (a) Show that every graph has a prime factor decomposition with respect to the Cartesian product.
   (b) Show that the factorization in (a) is not unique if the graphs are disconnected.
7. [**164**] Let G be the Cartesian product of two finite trees. Prove that if the robber cannot stay indefinitely on one vertex of G, then one cop can win on G.
8. Let K_n and K_m be complete graphs. For which products $\otimes$ is $K_n \otimes K_m$ cop-win?
9. Let P and Q be paths. For which products $\otimes$ is $P \otimes Q$ cop-win?
10. Prove Theorem 4.6.
11. Let G be a connected graph with n vertices. For each ordered pair of vertices (a, b), $a \neq b$, let

$$P_{ab} = \{u_0, u_1, \ldots, u_{k_b}\},$$

where $a = u_0$ and $b = u_{k_b}$, be a shortest path from a to b, and let f_{ab} be the retraction map $f_{ab}(x) = u_i$, if $d(x, a) = i < d(a, b)$ and $f_{ab}(x) = u_{k_b}$, if $d(x, a) \geq d(a, b)$. For ease of referencing, order all the ordered pairs and rename the paths and associated paths as P_i and f_i, respectively. Let $H = \boxtimes_{i=i}^{n(n-1)} P_i$. Show that $F(G) \to H$, defined by

$$F(x) = (f_1(x), f_2(x), \ldots, f_{n(n-1)}(x)),$$

is an embedding of G in H.

12. Let G be a connected graph. Suppose G has a homomorphism f with the property that $d(x, f(x)) \geq 2$. Show that G is not a cop-win graph.
13. For a graph G and integer $n > 1$, show that K_n is a retract of $G \times K_n$ if and only if $n \leq \omega(G)$.

14. Prove Theorem 4.7.

15. Prove Theorem 4.9 by first proving Lemma 4.8, then extending it to higher powers.

16. (a) Determine $cc(P)$, where P is the Petersen graph.
    (b) For each integer $k > 0$, give examples of graphs G_k satisfying $cc(G_k) - c(G_k) \geq k$.
    (c) For each integer $k > 0$, give examples of graphs G_k satisfying $\theta(G_k) - c(G_k) \geq k$.

17. Prove Theorem 4.10 using Lemma 4.11.

18. (a) Give an example of a dominating set in graph which is not a total dominating set.
    (b) For each integer $k > 0$, give examples of graphs G_k satisfying $\gamma_t(G_k) - \gamma(G_k) = k$.
    (c) Prove that $\gamma_t(G) \leq 2\gamma(G)$, and $\gamma_t(G) \leq n - \Delta(G) + 1$.
    (d) Determine $\gamma_t(C_n)$ and $\gamma_t(P_n)$.

19. Prove Theorem 4.13.

20. [**164**] If the $C(i)$, where $1 \leq i \leq n$, are cycles of length at least 5, then show that

$$c(\boxtimes_{i=1}^{n} C(i)) \leq n + 1.$$

21. (a) If G is a planar graph, then show that

$$|E(G)| \leq 3|V(G)| - 6.$$

    (b) Prove that K_5 is not planar. (*Hint*: Use part (a).)
    (a) If G is a planar graph with no triangles, then show that $|E(G)| \leq 2|V(G)| - 4$.
    (b) Prove that $K_{3,3}$ is not planar.

22. Show that a maximal outerplanar graph is cop-win.

23. In the proof of Theorem 4.25 show that in all cases the robber cannot enter the cop territory without getting caught.

24. A graph that can be embedded in the torus is called *toroidal.*
    (a) Show that K_7 is toroidal.
    (b) Show that the Petersen graph is toroidal.

25. (a) Find an infinite family of cop-win non-outerplanar graphs.

(b) Find an infinite family of non-outerplanar graphs with cop number 2.

26. [**113**] Recall the set of blocks $B(G)$ defined in the proof of Theorem 4.23 (note that $B(G)$ is well defined for any graphs, even those which are not outerplanar). Show that

$$\max_{G_i \in B(G)} \{c(G_i)\} \le c(G) \le \max_{G_i \in B(G)} \{c(G_i)\} + 1.$$

27. [**113**] A *cut set* S is a set of vertices whose deletion disconnects the graph. Let G be a graph and S a cut set of G, with $G_1, G_2, \dots, G_k$ the components of $G - S$. Prove that

$$c(G) \le \max\{c(G_1), c(G_2), \dots, c(G_k)\} + \gamma(S).$$

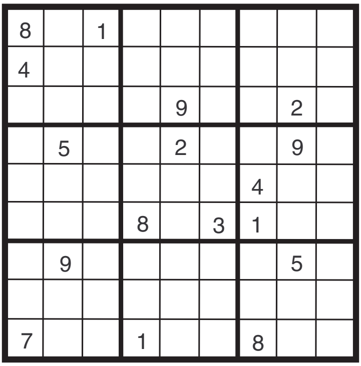
28. [**191**] A graph is *series-parallel* if it has no subgraph isomorphic to a subdivision of K_4. Hence, every outerplanar graph is series-parallel. Prove that every series-parallel graph has cop number at most 2.

Chapter 5

Algorithms

5.1. Introduction

Solving a Sudoku puzzle can be challenging. *Sudoku* is played over a 9×9 grid, divided into nine smaller 3×3 grids called *regions*. Some cells already have numbers in them, and all numbers are from 1 to 9. Each row, column, and region must completed so that they are permutations of 1 to 9. For an example, see Figure 5.1. From the point of view of graph theory, this is equivalent to the proper



8		1						
4								
				9			2	
	5			2			9	
						4		
			8		3	1		
	9						5	
7			1			8		

Figure 5.1. A Sudoku puzzle.

9-coloring of a graph with 81 vertices (one per cell), with two vertices joined if they are in the same row, column, or region. Analogous difficulty comes from solving a large jigsaw puzzle.

However, if you view a solved Sudoku or jigsaw puzzle, you can quickly check the solution with relative ease. In Sudoku, just scan each row, column, and region to determine if the solution is legal. In a jigsaw puzzle, if the final picture is not right, you know almost right away. Everyone agrees that usually checking whether a proposed solution is correct is easier than solving. These examples illustrate a famous dichotomy in theoretical computer science and mathematics between problems in the complexity classes **NP** and those in **P**. Nondeterministic polynomial time (or **NP**) problems are ones where you can easily verify an answer to be correct. In contrast, polynomial time (or **P**) problems are those where finding the answer is easy or fast. Precise definitions for **NP** and **P** will be given in Section 5.2.

Algorithms are pervasive in graph theory. One of the many useful features of finite graphs is that we can design algorithms to compute some of their properties and parameters. Of course, many properties have no polynomial time algorithm to check them (or at least none are thought to exist). No one, for example, would consider an algorithm realistic if it took longer than the expected life-span of the universe to terminate. For example, consider the problem of determining whether there is a *Hamilton cycle* in a given graph; that is, a cycle containing all vertices. You have lists available of all the vertices and edges, and must say either YES or NO (or 1 or 0, ON or OFF, or some other notation for truth and falsity). If given a cycle, then a quick check to determine if all the vertices are in C determines if it is Hamiltonian. See Figure 5.2. There are $n!$ permutations of a vertex-set of order n, so checking all of them is impractical, even for relatively small n. As we will discuss in Section 5.2, this is a famous example of an **NP**-complete problem. In other words, no one is likely ever to find an efficient solution. The same is true for checking whether a graph is 3-colorable (or 9-colorable, as is the case in Sudoku), or has an independent set or clique with a specified cardinality.

In contrast to finding Hamilton cycles, consider an *Eulerian graph*; that is, one where there is a circuit spanning each edge exactly once.

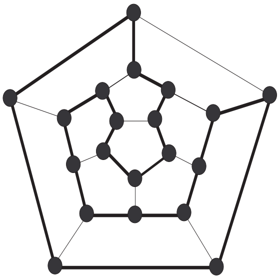


Figure 5.2. A Hamilton cycle in the dodecahedron.

See Figure 5.3. A graph is Eulerian if and only if each vertex is of even degree (see Exercise 1). Checking degrees of vertices is fast, so there is an efficient algorithm for determining whether a graph is Eulerian. We say this property is checkable in polynomial time. The same is true for checking whether a graph is bipartite or connected. Each of these are tractable problems with efficient algorithms.

What about algorithms for recognizing k-cop-win graphs? Fairly little work has been done on this problem, with the exceptions of [**16**], [**22**], [**85**], [**100**], and [**105**]. First, consider the case $k = 1$; that is, recognizing cop-win graphs. As discussed in Chapter 2, cop-win graphs have a pleasant structure in the form of cop-win ordering. It is quite simple to check if a given vertex is a corner: just check its neighbors against the neighbors of all the other vertices. For large

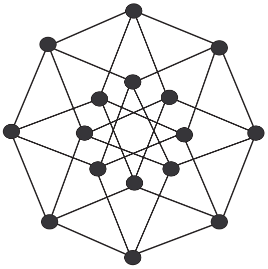


Figure 5.3. The hypercube Q_4 is Eulerian.

graphs this can be lengthy to do by hand, but a computer can perform this comparison fairly quickly. Once you find a corner, try to find another. If there are no corners, then you know the graph is robber-win. Proceeding recursively, you can determine whether there is a cop-win ordering, and hence, whether the graph is cop-win. This is the algorithm for deciding cop-win graphs, and it runs in polynomial time (finding a corner runs in polynomial time, and we need to find at most n corners). Hence, deciding whether a graph is cop-win is in **P**.

The surprising thing is that if we ask whether $c(G) \leq k$ for a fixed integer $k > 1$, the answer can be found in polynomial time. If you read Chapter 2, then perhaps this is less surprising owing to the elimination ordering for k-cop-win graphs recently introduced in [**57**]. See Section 5.3 and Theorem 5.1. Theorem 5.1 was known fairly early on, and the proof of it was published first in [**16**]. We present a different approach, inspired by the strong products used in Chapter 2.

In the preceding paragraphs, we take k as fixed and not part of the input. But what if you do not fix k? Despite Theorem 5.1, it was recently shown in [**85**] that if k is not fixed and part of the input, then the problem of determining if $c(G) \leq k$ is **NP**-hard. **NP**-hard problems are the most difficult problems in **NP**, so this is bad news for computing the cop number of a graph. The authors of [**85**] use a reduction to the well-known **NP**-hard domination problem. See Theorem 5.5.

We do not assume that the reader has much or any background in algorithms. For this reason, our next section is devoted to giving an overview of **P**, **NP**, reductions, and algorithms in graphs.

5.2. Background on Complexity

The purpose of this section is to give an introduction to graph algorithms and their complexity. It can be skipped by those versed in the topic, although we set out notation that will be used throughout the chapter. All of the algorithms we consider (unless otherwise stated) focus on graphs or *graph decision problems*. That is, the *input* is a (typically undirected, simple, and always finite) graph G, where we are given some data structure representing G. Sometimes we include a positive integer k as an additional input. For us, as elsewhere in this book, G is connected. We could represent G by an adjacency matrix or adjacency list. If G is order n, then the *adjacency matrix* of G has (i, j) entry equaling 1 if i and j are joined, and 0 otherwise. An *adjacency list* is consists of n lists, one for each vertex v, so that the list of v enumerates all the neighbors of v. For example, the adjacency matrix of the 4-cycle C_4 is

$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix},$$

and has the adjacency list

$$\begin{aligned} 1&: 2, 4 \\ 2&: 1, 3 \\ 3&: 2, 4 \\ 4&: 1, 3. \end{aligned}$$

In practical implementations of algorithms, the choice of data structure is extremely important. In theory, we do not care as much about which data structure is used to represent the graph, so long as the graph is completely and efficiently represented. We are also less interested in the encoding of the algorithm; it could be pseudocode (see Algorithm 1 below) or simply a set of instructions written in prose.

An *algorithm* is a method or procedure for solving a problem. It is purely deterministic: at any stage during an algorithm, the method tells you exactly what you need to do to get to the next stage. Algorithms are encoded by a finite set of instructions which dictate what happens at any step in the computation. All the algorithms we consider terminate, and are executed in a finite number of steps. Problems solved by an algorithm are *decidable*; otherwise, they are *undecidable.* We only consider decidable problems here; see Exercise 20.

The *output* of an algorithm deciding a graph decision problem is simply YES or NO. For example, the following is a graph decision problem.

EULERIAN: Is G Eulerian?

As discussed in the introduction of this chapter, there is a fast algorithm for solving this problem, based simply on checking degrees of vertices (recall again that G is connected). To make "fast" precise, we define the *time complexity* (or just *complexity*) of a graph decision problem (or just *problem*) to be the minimum worst-case running time over all possible algorithms solving the problem as a function of the length of the input. (There is an analogous notion of space complexity, which we will not define.) Here *running time* (or *cost*) is defined as the number of steps needed for the algorithm to conclude either yes or no, and the length of the input is usually taken to be the number of vertices, which we take as n a positive integer. The actual unit of time taken for a given step is irrelevant; we simply treat it as an indivisible discrete unit of time. For a function $f : \mathbb{N}^+ \rightarrow \mathbb{N}^+$, we say a problem has *complexity* f if the time complexity of a problem with length of input n equals $f(n)$.

A problem is *solvable in polynomial time* if its complexity is $O(n^m)$, for some integer $m \geq 0$. A *polynomial time algorithm* is defined in an analogous way. The set of all problems solvable in polynomial time is denoted by **P**. Hence, checking if a graph is Eulerian is in **P**. It is not hard to show that checking whether a graph is bipartite is in **P** (see Exercise 2a). The class **P** can be more precisely defined as decision problems recognized by a deterministic Turing machine in polynomial time. We do not pursue this more formal approach, but the reader is directed to [**116**] and [**138**] for further reading.

The *Cobham-Edmond* or extended *Church-Gödel thesis* states that the feasible graph decision problems are exactly those which are in **P**; see [**116**] and [**138**]. In particular, if a particular problem is not in **P**, then the thesis implies there is no "fast" algorithm to solve it. Much of our intuition and experience agrees with the thesis, which is widely accepted by experts in complexity theory. Of course, an algorithm with running time $O(n^{200})$, while polynomial, is impractical even for small n; constants hidden in the "big Oh" notation may also make the running time too long for practical purposes.

NP (or *non-deterministic polynomial*) is the set of decision problems such that if the answer is YES, then there is a *certificate* (or *argument* or *proof*) of this fact that can be checked in polynomial time. The certificate has length a polynomial sized function of the input. Every YES input to the problem has at least one certificate (possibly many), and each NO input has none. Intuitively, **NP** is the set of decision problems where we can verify a YES answer quickly if we are given a certificate. So the Sudoku and jigsaw puzzle analogies in the introduction of this chapter capture the essence of **NP**: you can tell quickly if a proposed solution (the solution is the certificate in either case) is correct. But as those who tried the Sudoku puzzle above can attest, it could take a long time to find a solution! An alternative definition of **NP** consists of those decision problems decided by a non-deterministic Turing machine; see [**116**] and [**138**]. We omit this more formal definition in favour of the one with certificates, which is an equivalent and often more useful formulation.

The class **NP** is large. For example, following problem is in **NP**.

HAMILTONIAN: Is G Hamiltonian?

The certificate here is simply a Hamilton cycle. We can quickly verify that the proposed solution is an actual Hamilton cycle. Each of the following problems is in **NP**, and we leave it to Exercise 9 to find the certificates. (Note that k is specified in the input and it is not fixed.)

CLIQUE: Given an integer $k \geq 2$, is there a k-clique in G?

3-COLORING: Is G 3-colorable?

DOMINATION: Given an integer $k \geq 2$, is it true that $\gamma(G) \leq k$?

INDEPENDENT SET: Given an integer $k \geq 2$, is there a k-co-clique in G?

Both **P** and **NP** are examples of *complexity classes*. How these classes fit together or overlap is not well understood. It is evident that

P$\subseteq$ **NP**, since a certificate of a problem in **P** is a polynomial time algorithm used to solve the problem. One of the deepest current open problems in mathematics and computer science is whether **P**=**NP**. Most experts think the answer is no, but no one has yet proven that there is a problem in **NP** that is not in **P**. The **P** versus **NP** problem is one of the Clay Millennium Prize problems, whose solution comes with a million dollar prize. However, the authors think there are easier ways to earn a million dollars!

A problem is **NP**-*hard* if a polynomial-time algorithm for it would imply a polynomial-time algorithm for every problem in **NP**. Hence, if an **NP**-hard problem were in **P**, then **P**=**NP**. So it is unlikely that any **NP**-hard problem is in **P**! **NP**-hard problems are at least as hard as any problem in **NP**.

An **NP**-*complete problem* is one which is **NP**-hard and in **NP**. Each of the five problems listed above are **NP**-hard and in **NP**, and so are **NP**-complete. The *Halting problem*, written HALTING, asks if given a program and an input, whether the program will eventually halt on that input. The Halting problem is **NP**-hard but not in **NP** (see [**193**] and Exercise 20). Unless **P**=**NP**, the classes of **P** and **NP**-complete problems are disjoint. Again, most experts think that no **NP**-complete problem is in **P**, but there is no proof of this fact yet. See Figure 5.4 for a depiction of generally accepted relations between these complexity classes.

The method to prove that a problem is **NP**-hard is to use a suitable reduction. A *reduction from problem* X *to* Y is a polynomial time algorithm $\mathcal{A}$ which transforms inputs of X to equivalent inputs of Y. Hence, given an input G to problem X, the algorithm $\mathcal{A}$ produces an input $\mathcal{A}(G)$ to problem Y, such that G is a YES input of X if and only if $\mathcal{A}(G)$ is a YES input of Y. We write $X \leq Y$ if there is reduction from X to Y. If we reduce X to Y, then we are establishing that roughly X is no harder than Y. Hence, if Y is **P**, then so is X. If X is **NP**-hard, then so is Y. Using the already mentioned pool of **NP**-hard problems, we can use this as a tool to show that the problem is **NP**-hard.

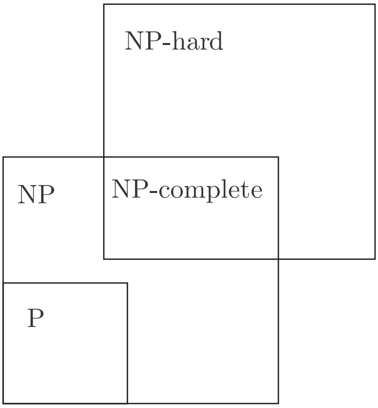


Figure 5.4. Inclusion among complexity classes, assuming **P**≠**NP**.

k-COP NUMBER: Given a positive integer k, for the input (G, k) is $c(G) \leq k$?

See Theorem 5.5. In contrast, as shown in Theorem 5.1. the following problem is in **P**:

k-FIXED COP NUMBER: Given a fixed positive integer k, for the input G is $c(G) \leq k$?

The main difference between the two problems is that in k-COP NUMBER the integer k may be a function of n, and so it grows with n. In k-FIXED COP NUMBER, k is fixed and not part of the input, and so it is independent of n.

Reductions often require ingenuity. The first reductions were introduced by Cook and Levin. A *Boolean formula* is a logical expression over *Boolean variables* (that can take values in $\{0, 1\}$) with *connectives* $\vee$, $\wedge$, and $\neg$. A variable and its negation are *literals.* A disjunction of literals is a *clause.* Without loss of generality, we assume that Boolean formulas are in *conjunctive normal form* consisting of the conjunction of clauses. For example,

$$(x \vee y \vee \neg z) \wedge (x \vee y) \wedge (\neg x \vee z)$$

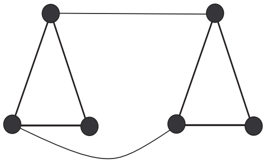


Figure 5.5. The graph $G(\Phi)$, where $\Phi = (x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z)$.

is a Boolean formula. Let SAT denote the set of all satisfiable Boolean formulas (namely those formulas for which there is a Boolean assignment to the variables which gives it the value 1). In his pioneering paper from 1971, Cook introduced **NP**-complete problems and showed that SAT is **NP**-complete [**64**]. (Finding the first **NP**-complete problem was a crucial step in finding others; thousands of **NP**-complete problems are now known!) Levin [**137**] independently defined **NP**-complete problems and showed that a variant of SAT is **NP**-complete. Karp [**126**] showed that 21 central problems (including proper coloring) are **NP**-complete, igniting a firestorm of interest in the topic. We note that the specialization 3-SAT, where each clause in a Boolean expression contains exactly three literals, is also **NP**-complete; see Exercise 16.

From this, one can derive that many graph problems such as 3-COLORING and HAMILTONIAN are **NP**-complete. Reductions in graph algorithms usually use some kind of *gadget* or *auxiliary graph* construction. As an easy example, CLIQUE can be reduced to INDEPENDENT SET by taking the complement of the input graph. For a less trivial example, we can reduce 3-SAT to INDEPENDENT SET by the following reduction. For a given Boolean formula Φ with clauses

$$X_1, X_2, \ldots, X_k,$$

we form a graph $G(\Phi)$ such that there is a k-independent set in $G(\Phi)$ if and only Φ is satisfiable. To accomplish this, for each clause X_i, construct a 3-clique and associate each vertex with a unique literal in X_i. Two vertices are joined in different triangles if the literals they correspond to are negations of each other. See Figure 5.5.

If Φ is satisfiable, then given any truth assignment satisfying Φ, we can find an independent set in $G(\Phi)$ by choosing for each clause a vertex corresponding to a literal that is satisfied. The converse (that there is an independent set in $G(\Phi)$ of cardinality k implies that Φ is satisfiable) is left as Exercise 10.

There is much more background on **P** and **NP** than the sketch given here. For an excellent overview of the classes **P** and **NP**, see the books [**94**], [**138**], and [**172**], and the survey [**198**].

We introduce another complexity class which contains **NP**, and will be useful in our discussion of algorithmic properties of the cop number. **EXPTIME** (short for *exponential time*), is the set of decision problems of complexity $O(2^{p(n)})$. It is known that

$$\mathbf{P}\subseteq\mathbf{NP}\subseteq\mathbf{EXPTIME}$$

(see [**172**]). To see the latter inclusion, note that each **NP** problem can be solved in exponential time by a brute force search for the certificate. It is generally believed that $\mathbf{NP}\subsetneq\mathbf{EXPTIME}$**.** We note in passing (see [**172**]) that $\mathbf{P}\subsetneq\mathbf{EXPTIME}$. A problem is **EXPTIME**-*complete* if it is in **EXPTIME** and every problem in **EXPTIME** reduces to it (that is, it is at least as hard as any problem in **EXPTIME)**. An algorithm runs in *sub-exponential time* if it has complexity $2^{o(n)}$. A well-known sub-exponential algorithm is for the following problem:

ISOMORPHISM: Given graphs G and H, is it the case that $G \cong H$?

The ISOMORPHISM problem is known to run in time $2^{O(\sqrt{n \log n})}$ [**143**], and is one of the few **NP** problems not known to be either in **P** or **NP**-complete.

5.3. Polynomial Time with k Fixed

Our main goal is to show that there is a polynomial-time algorithm that can determine whether $c(G) \le k$ assuming that k is fixed (that is, not a function of $|V(G)|$). Define the following decision problem.

k-FIXED COP NUMBER: Given a fixed positive integer k, for the input G is $c(G) \leq k$?

We prove the following theorem.

Theorem 5.1. *The problem FIXED COP NUMBER is in **P**.*

Theorem 5.1 first appeared in [**16**], and was later reproved in [**105**]. Another proof of the result follows implicitly from the characterizations given in [**57**] (see Theorem 2.11 of Chapter 2).

The approach we take to proving Theorem 5.1 comes from [**22**], and relies on characterizing the game via strong products, in a way not dissimilar to the methods of Chapter 2, or that used implicitly in [**105**]. However, the methods of [**105**] using strong products of graphs is different from ours. Given a graph G, recall that the kth *strong power of* G, written $G^k_{\boxtimes}$, is the strong product of G with itself k times. That is, vertices are k-sequences of vertices of G, and two sequences are joined if they are joined or equal in each coordinate. As in Chapter 2, we identify the positions of k-many cops in G with a single vertex in $G^k_{\boxtimes}$. The definition of the strong power allows us to simulate movements of the cops in G by movements of a single cop in $G^k_{\boxtimes}$.

See [**119**] for additional background on strong products of graphs. For a set X, define 2^X to be the set of subsets of X. For $S \subseteq V(G)$, define $N_G[S]$ to be the union of the closed neighbor sets of vertices in S.

Theorem 5.2 ([**22**])**.** *Suppose $k \geq 1$ is an integer. Then $c(G) > k$ if and only if there is a mapping $f : V(G^k_{\boxtimes}) \to 2^{V(G)}$ with the following properties.*

(1) *For every $u \in V(G^k_{\boxtimes})$,*

$$\emptyset \neq f(u) \subseteq V(G) \setminus N_G[u].$$

(2) *For every $uv \in E(G^k_{\boxtimes})$,*

$$f(u) \subseteq N_G[f(v)].$$

Proof. Let k cops play on G. If R has a winning strategy (that is, $c(G) > k$), then for $u \in V(G^k_{\boxtimes})$ define $f(u)$ to be the set of all vertices

$r \in V(G)$ such that if the cops start from the initial position u, then robber can start from r and win the game.

For the proof of item (1), since R has a winning strategy, $f(u)$ is non-empty for every $u \in V(G^k_\boxtimes)$. To show that $f(u) \subseteq V(G) \setminus N_G[u]$, assume r is in $f(u)$. Then r cannot be in $N_G[u]$; otherwise, C can capture the robber, which contradicts the fact that R can win the game starting from this position.

To prove item (2), let uv be an edge in $E(G^k_\boxtimes)$ and fix $z \in f(u)$. The robber can win if the cops are in u and the robber is in z. Since $uv \in E(G^k_\boxtimes)$, C can move the cops from u to v. Since R has a winning strategy, R must be able to move the robber from z to a vertex z' that is adjacent or equal to z. Therefore, $z' \in f(v)$. Since every vertex z of $f(u)$ is either in $f(v)$ or has a neighbor z' in $f(v)$, we have $f(u) \subseteq N_G[f(v)]$.

For the reverse direction, assume now that a mapping f exists with properties (1) and (2). We show that R has a strategy to avoid capture. Let $u^{(0)} \in V(G^k_\boxtimes)$ be the positions of the k cops in round 0; that is, $u_i^{(0)} \in V(G)$ is the position of the ith cop, for all $1 \leq i \leq k$. In round 0, the robber R moves to an arbitrary vertex in $f(u^{(0)})$. This is possible, because the first property of f demands that $f(u^{(0)}) \neq \emptyset$. In round 1 the cops cannot capture the robber since by the first property of f, the vertices of $f(u^{(0)})$ have distance at least two from any cop in $u^{(0)}$.

We argue that for all $t \geq 0$ the robber can move safely to $f(u^{(t)})$ in round t, where $u^{(t)}$ is the position of the k cops in round t. Suppose this claim is true for $t \leq m$. We prove that the claim is true for $m+1$. In each round a cop can move to an adjacent vertex, so

$$u^{(m)}u^{(m+1)} \in E(G^k_\boxtimes).$$

Therefore, by the second property of f, $f(u^{(m)}) \subseteq N_G[f(u^{(m+1)})]$. Hence, the robber at $f(u^{(m)})$ can move to a vertex in $f(u^{(m+1)})$ in round $m+1$ and avoid capture. □

We now consider a polynomial-time algorithm for determining whether $c(G) \leq k$ based on Theorem 5.2. We express this argument in *pseudocode*, which is a compact way of summarizing the steps in the algorithm. It also gives us an easy way to reference the lines of the algorithm when analyzing the running time.

Algorithm 1 CHECK k-COP NUMBER

```
Require: $G = (V, E)$, $k \geq 1$
 1: initialize $f(u)$ to $V(G) \setminus N_G[u]$, for all $u \in V(G^k_{\boxtimes})$
 2: repeat
 3:     for all $uv \in E(G^k_{\boxtimes})$ do
 4:         $f(u) \leftarrow f(u) \cap N_G[f(v)]$
 5:         $f(v) \leftarrow f(v) \cap N_G[f(u)]$
 6:     end for
 7: until the value of $f$ is unchanged
 8: if there exists $u \in V(G^k_{\boxtimes})$ such that $f(u) = \emptyset$ then
 9:     return $c(G) \leq k$
10: else
11:     return $c(G) > k$
12: end if
```

For those less familiar with pseudocode, we give a high-level description of Algorithm 1. By Algorithm 1 we want to find a function f that has both properties of Theorem 5.2. Initially we set $f(u)$ to the largest possible set; that is,

$$f(u) = V(G) \backslash N_G[u].$$

The set $f(u)$ cannot be any larger set because of the first property of Theorem 5.2. The main idea is that we change the value of $f(u)$ from S to S' only if we have a *proof* that the elements of $S \backslash S'$ cannot be in $f(u)$. In other words, at each step, we have a proof that if any function, say $g(u)$, satisfies Theorem 5.2, then $g(u) \subseteq f(u)$ (this is trivially true after the initialization). A proof that an element is not in $f(u)$ can be the following: if x is not in $N_G[f(v)]$ and uv is an edge, then, according to the second property of Theorem 5.2, x cannot be in $f(u)$. Thus, we keep removing elements like x from $f(u)$ until there is no such element. Then $c(G) > k$ if and only if the final value of f satisfies both properties of Theorem 5.2.

The following theorem gives Theorem 5.1 as a corollary.

Theorem 5.3 ([**22**])**.** *Algorithm* 1 *runs in time* $O(n^{3k+3})$.

Proof. We may determine if there exists a mapping f with properties stated in Theorem 5.2 using Algorithm 1. It is clear that if the algorithm terminates, it will answer correctly; either it finds a function f with properties stated in Theorem 5.2, or no such f exists because nothing from $f(u)$ will be removed unless it is necessary. In other words, for any mapping f' with properties stated in Theorem 5.2, we will have $f'(u) \subseteq f(u)$ for all $u \in V(G^k_\boxtimes)$, where f is the mapping found by Algorithm 1. Hence, if $f(u) = \emptyset$ for some u, there is no mapping with the stated properties. The running time of Algorithm 1 is at most $O(n^{3k+3})$, since the repeat loop in lines 2–7 iterates at most $O(n^{k+1})$ times (lines 2–7 form the bottleneck here). To see this, in each iteration except the last one, the cardinality of $f(u)$ will be decreased for at least one u. Hence, in each iteration, we are checking all the edges of the $G^k_\boxtimes$ (lines 3–6). For each edge, we are computing intersections (which has complexity $O(n)$) and neighborhoods (of complexity $O(n^2)$). Therefore, in each iteration the running time is

$$\begin{aligned} |E(G^k_\boxtimes)|O(n+n^2) &= O(n^{2k})O(n^2). \\ &= O(n^{2k+2}). \end{aligned}$$

It follows that the total running time is

$$O(n^{k+1})O(n^{2k+2}) = O(n^{3k+3}). \qquad \square$$

Algorithm 1 can be modified to run in a more efficient way in time $O(n^{2k+3})$. See Exercise 3. As our main goal is proving Theorem 5.1, we omit a discussion of this algorithm (see [**22**]). Note that the algorithm in [**105**] for answering $c(G) \leq k$ also runs in time $O(n^{2k+3})$ (although this was only implicit in [**105**]).

As noted in [**85**], Theorem 5.1 helps show that computing the cop number is sub-exponential time. Define

k-COP NUMBER: Given a positive integer k, for the input (G, k) is $c(G) \leq k$?

Recall that in this problem, k is not fixed and may be function of n.

Theorem 5.4 ([**85**]). *The problem* COP NUMBER *is solvable in sub-exponential time.*

As remarked in [**85**], Theorem 5.4 is interesting, as many **NP**-hard problems are believed not to be solvable in sub-exponential time.

5.4. NP-hard with k Not Fixed

We sketch the proof from [**85**] that computing the cop number is **NP**-hard. (See also [**84**].) Recall the following decision problem from Section 5.2.

k-COP NUMBER: Given a positive integer k, for the input (G, k) is $c(G) \leq k$?

Note that k may be a function of n. For example, this question decides whether $c(G) \leq \lfloor c\sqrt{n} \rfloor$, where $c > 0$ is a constant. As we know from Meyniel's conjecture from Chapter 3, this is a much harder problem than determining whether say, a graph is 2-cop-win. We prove the following theorem.

Theorem 5.5 ([**85**]). *The problem k-*COP NUMBER *is **NP**-hard.*

Theorem 5.5 does not say that k-COP NUMBER is in **NP**; that is an open problem! See Section 5.5 below. The approach we take to proving Theorem 5.5 is to use a reduction from the following **NP**-complete problem:

DOMINATION: Given an integer $k \geq 2$, is it true that $\gamma(G) \leq k$?

We first describe a certain family of bipartite graphs. This family is critical for the reduction, and any particular graph in the family can be constructed in polynomial time. For positive integers n, m, and r, define a bipartite graph $H(m, n, r)$ with size rmn^2. The vertex classes are X and Y, so that $|X| = |Y| = nm$. The set X is partitioned into sets $U_1, U_2, \ldots, U_n$, and Y is partitioned into sets $W_1, W_2, \ldots, W_n$, where $|U_i| = |W_i| = m$ for all i. Define $H_{i,j}$ to be the subgraph of $H(m, n, r)$ induced by $U_i \cup W_j$, and let $\deg_{i,j}(u)$ be the degree of the vertex u in $H_{i,j}$. See Figure 5.6.

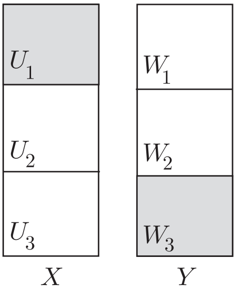


Figure 5.6. The graph $H(m, 3, r)$. The induced subgraph $H_{1,3}$ is shaded.

We omit the proof of the following theorem—whose proof is contained in [**85**]. We do mention that $H(m, n, r)$ is the output of an algorithm with polynomial complexity in m, n, and r.

Theorem 5.6 ([**85**]). *If*

$$m \geq 2n(r+1)\frac{(n(r+1)-1)^6 - 1}{(n(r+1)-1)^2 - 1},$$

then $H(n, m, r)$ can be constructed with the following properties.

(1) *There is an algorithm which, given a set $X \cup Y$ as in the definition of $H(m, n, r)$ with no edges, constructs $H(m, n, r)$ in time $O(rmn^2)$ on $X \cup Y$.*

(2) *For all $u \in V(H_{i,j})$ and all i and j,*

$$r - 1 \leq \deg_{i,j}(u) \leq r + 1.$$

(3) *For all vertices u,*

$$\deg(u) \leq n(r+1).$$

(4) *The girth of $H(m, n, r)$ is at least* 6.

Proof of Theorem 5.5. Let G be a graph with n vertices, labeled as $\{v_1, \ldots, v_n\}$. Let $r = k + 2$, and define

$$m = \left\lceil 2n(r+1)\frac{(n(r+1)-1)^6 - 1}{(n(r+1)-1)^2 - 1} \right\rceil .$$

For each v_i add $2m$ vertices, and make these new vertices joined to all the neighbors of v_i in G and to v_i itself. Half of these new $2m$ vertices form the set U_i, while the other half form the set W_i. Apply Theorem 5.6 (1) to construct in polynomial time in k and n, the bipartite graph $H(m,n,r)$ on $X \cup Y$, where

$$X = \bigcup_{i=1}^{n} U_i \text{ and } Y = \bigcup_{i=1}^{n} W_i.$$

We denote the resulting graph (constructed in polynomial time in n and k) by G'.

We claim that $\gamma(G) \leq k$ if and only if $c(G') \leq k$. Once this claim is established, the reduction from DOMINATION follows, as does the proof of the theorem.

Now suppose that S is a dominating set of cardinality at most k in G. It is straightforward to see that if we place the cops on S in G', then the robber is captured in the first round.

Next assume that $\gamma(G) > k$. We show that $c(G') > k$.

If we play with k cops in round 0, then the robber has a *safe position* (that is, a vertex not joined to a cop) v_i in G (as $\gamma(G) > k$; note that we consider G to be an induced subgraph of G'). By Theorem 5.6 (2), the cops are joined or equal to at most $kn(r+1)$ vertices in U_i. Since $|U_i| = m$, we have that

$$m = \left\lceil 2n(r+1)\frac{(n(r+1)-1)^6-1}{(n(r+1)-1)^2-1} \right\rceil > kn(r+1).$$

Therefore, there is a safe vertex $u_i \in U_i$ for the robber. Let $R = u_i$ in round 0. We proceed by induction on the number of rounds, assuming for an induction hypothesis that $R = u \in X \cup Y$ in round $t \geq 0$ and is safe from the cops. Without loss of generality, we assume that $u \in X$ by symmetry.

The induction step has a proof reminiscent of the proof of Theorem 1.3 from Chapter 1. If in round $t+1$ the R is not joined to a cop, then the robber passes. Otherwise, the robber proceeds as follows. As in the base case, there is a vertex v_j not in $N[C]$. By Theorem 5.6 (2) the vertex u has at least $r - 1 = k + 1$ neighbors in W_j. By Theorem 5.6 (4), the girth of $H(m,n,r)$ is at least 6. Hence, any given cop is joined to at most one neighbor of u. As there are only k cops, at least one of these neighbors z is not in $N[C]$. The robber moves to z and is safe for another round. □

5.5. Open Problems

Our algorithmic knowledge of the COP NUMBER problem is somewhat limited. We do not even know if the problem is in **NP**! The deepest conjecture on the algorithmic properties of cop number is due to Goldstein and Reingold [**100**].

Conjecture: The problem COP NUMBER is **EXPTIME**-complete.

Goldstein and Reingold [**100**] proved that the version of the Cops and Robbers game on directed graphs is **EXPTIME**-complete. They also proved that the version of the game on undirected graphs when the cops and the robber are given their initial positions is also **EXPTIME**-complete. We do not include the proofs of these results (which relies on a reduction to the so-called *Alternating Boolean Formula* (or *ABF*) decision problem, which is **EXPTIME**-complete), as they would lead us too far afield. If this conjecture is true, then COP NUMBER is not in **NP** unless **NP**=**EXPTIME**. However, the common wisdom is that **NP** is a proper subset of **EXPTIME**, and so it is unlikely that COP NUMBER is in **NP**. Hence, a weaker conjecture than the above is to show the following.

Conjecture: The problem COP NUMBER is not **NP**-complete.

A graph G is a *split graph* if $V(G)$ can be partitioned into sets C and I, such that C is a clique, and I is an independent set. See Figure 5.7. Some algorithmic problems which are **NP**-hard on general graphs are in **P** for split graphs. In particular, the independence number and treewidth of a split graph can be computed in linear time (see [**33**]). However, it was proved in [**85**] that COP NUMBER remains **NP**-hard when restricted to split graphs. They also proved that the problem is in **P** for *interval graphs*, which are the intersection graphs of intervals on the real line. Many other graph classes were studied with the property that **NP**-hard problems become polynomial time solvable; see [**33**]. It would be interesting to investigate the complexity of computing the cop number in those classes.

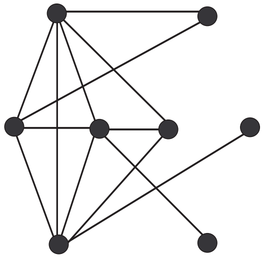


Figure 5.7. A split graph with $C = K_4$ and $I = \overline{K_4}$.

Approximation algorithms are an important part of complexity theory; see Vazirani [**194**]. Approximation algorithms compute near-optimal solutions. An algorithm has *approximation ratio* of $f(n)$, if for any input of cardinality n, the cost C' of its solution is within a multiplicative factor $f(n)$ of the cost C of an optimal solution. More precisely,

$$\max\left\{\frac{C}{C'}, \frac{C'}{C}\right\} \leq f(n).$$

A $f(n)$-*approximation algorithm* has approximation ratio $f(n)$. All the algorithms given in this chapter are optimal, and so are simply 1-approximation algorithms. It was proved in [**85**] that there is a constant $c > 0$ such that there is no polynomial time algorithm to approximate $c(G)$ within a multiplicative factor $c \log n$, unless $\mathbf{P} = \mathbf{NP}$. A problem posed in [**85**] is whether for some $\varepsilon > 0$ there is an $n^{1-\varepsilon}$-approximation algorithm for the COP NUMBER problem.

Exercises

1. Prove that a graph is Eulerian if and only if each vertex is even degree.
2. (a) Show that the problem of deciding whether a graph is bipartite is in **P**.

   (b) Repeat part (a), but for deciding whether a graph is connected.

3. [**22**] Devise an algorithm which runs in time $O(n^{2k+3})$ solving the problem FIXED COP NUMBER.

4. Cops and Robbers is defined analogously on *directed* graphs. Consider a decision problem where the input is a finite directed graph. By modifying the proof of Theorem 5.1, show that the problem FIXED COP NUMBER for digraphs is in **P**.

5. [**105**] For a digraph G and k a positive integer, define the *k-game digraph of G*, written $\Gamma_k(G)$, by first forming the disjoint union of G and $G^k_{\boxtimes}$ (note that strong products are defined analogously for directed graphs). Now add two vertices u and v. Add directed edges from u to each vertex in G and $G^k_{\boxtimes}$; add directed edges from v to G and from each vertex of $G^k_{\boxtimes}$ pointing to v. For each vertex $x = (x_1, \ldots, x_k)$ of $G^k_{\boxtimes}$, add directed edges from x to each outneighbor of x_i in G, for $1 \leq i \leq k$.
   (a) Draw the graph $\Gamma_2(C_4)$, and explain why it is cop-win.
   (b) Prove that G is k-cop-win if and only if the digraph $\Gamma_k(G)$ is cop-win.
   (c) Use (b) to devise a polynomial time algorithm that decides the problem FIXED COP NUMBER.

6. (a) Prove that a planar graph has a vertex of degree at most 6.
   (b) Use (a) to give a polynomial time algorithm which gives a proper 6-coloring of a planar graph.

7. An *edge-labeled graph* is one whose edges are assigned positive integer weights. A *minimum spanning tree* (or *MST*) in an edge-labeled graph is a spanning tree whose sum of edge labels is as small as possible (there may be more than one). Find a polynomial time algorithm which produces an MST in an edge-labeled graph. (*Hint*: Use a greedy algorithm.)

8. Prove that the reduction relation $\leq$ is transitive: if $A \leq B$ and $B \leq C$, then $A \leq C$.

9. Find certificates to show that each of the following problems is in **NP**. Note that k is a fixed positive integer and is part of the input.

CLIQUE: Given an integer $k \geq 2$, is there a k-clique in G?

3-COLORING: Is G 3-colorable?

DOMINATION: Given an integer $k \geq 2$, is it true that $\gamma(G) \leq k$?

INDEPENDENT SET: Given an integer $k \geq 2$, is there a k-co-clique in G?

10. Show that if there is an independent set in the $G(\Phi)$ of cardinality k, then the Boolean formula Φ is satisfiable.
11. A *vertex cover* of a graph is a set of vertices that touches every edge in the graph. The problem VERTEX COVER problem is, given a positive integer k, to decide whether there exists a vertex cover of cardinality at most k in a given graph. Show that VERTEX COVER is **NP**-complete. (*Hint*: Reduce from INDEPENDENT SET.)
12. In the MAXCUT problem, given a positive integer k, we must decide whether there is a subset of vertices S such that there are at least k edges that have one endpoint in S and the other endpoint in $V\backslash S$. Prove that MAXCUT is **NP**-complete.
13. Show that the greedy algorithm is a 2-approximation algorithm for MAXCUT.
14. Give an $O(n^2)$ algorithm for deciding if a graph is cop-win.
15. Prove that deciding whether a given planar graph is 3-colorable is **NP**-complete.
16. Show that SAT$\leq$ 3-SAT.
17. The *complement* of a decision problem is the decision problem resulting from reversing the YES and NO answers. The complexity class **coNP** is defined to be the set of problems whose complement is in **NP**.
    (a) Show that **P**$\subseteq$**NP**$\cap$ **coNP**. Problems in **NP**$\cap$ **coNP** are said to have a *good characterization;* see, for example, [**142**]. It is widely thought that **P**=**NP**$\cap$ **coNP.**
    (b) Show that the problem of whether a graph is cop-win has a good characterization.

18. (For those readers with some programming background.) Implement Algorithm 1 for small values of k, such as $k = 1, 2, 3$.

19. [**83**] Prove that a graph is split if and only if no induced subgraph is a cycle on four or five vertices, or a pair of disjoint edges.

20. A decision problem is *undecidable* if there is no algorithm that solves the problem.

    (a) Prove that there are uncountably many undecidable decision problems.

    (b) [**193**] The *Halting Problem*, written HALTING, has as input the description of a program, and decides whether or not the program halts. Show that HALTING is undecidable.

21. [**85**] Prove Theorem 5.4. (*Hint*: Use Theorems 5.3 and 3.6.)

Chapter 6

Random Graphs

6.1. Introduction

Randomness is a fundamental property in both mathematics and nature. The probabilistic method—championed by Erdős and Rényi in their pioneering work on the subject in the 1960s (see [**73**], [**74**], and [**75**])—remains one of the most powerful tools in graph theory. Random graphs play an important role both as displaying asymptotic and typical properties of graphs, and as a beautiful theory in their own right. See the books [**6, 18, 123**], or Chapter 3 of [**21**] for more discussion on random graphs. We saw one application of the probabilistic method in Chapter 3 in the proof of Meyniel's conjecture for diameter 2 graphs (see Theorem 3.10). In this chapter, our approach is different: we choose a graph at random by randomly drawing edges, independently and with given probability, then play the game there. Results about the cop number then become fuzzy: the cop number is a random variable over a probability space, and theorems aim to estimate this variable. Graph theoretic problems in random graphs often reduce them to analytic ones. The good news for those without extensive probabilistic background is that we need only a few basic properties on the expected value (such as its linearity and Markov's inequality) to accomplish the estimates.

The edge probabilities p in random graphs are often taken as varying with the order of the graph. We focus on cases for edge probabilities of random graphs, going from less to more complex arguments. In Section 6.2 we consider the simplest case when p is a constant, independent of n. In Section 6.3 we discuss results for variable $p = p(n)$, and for both sparse and dense random graphs. The most interesting and definitive result on the cop number of a random graph is the recent striking and surprising Zig-Zag Theorem of Łuczak and Prałat [**142**]. We present the Zig-Zag Theorem and a discussion of its proof in Section 6.4. We finish with a discussion of the cop number of random graphs in models of the web graph.

6.1.1. Probabilistic Tools and Random Graphs. Readers with little or no experience with random graphs should be relieved to hear that all that is required is elementary probability theory (usually taught in a first undergraduate course on the topic), and some calculus. Readers wishing a detailed background on discrete probability theory should consult [**102**].

Random variables and expectation were introduced in Chapter 3. For example, familiar graph parameters, such as size, chromatic number, and cop number, become random variables in random graphs. Expectation has the following elementary but useful property, which follows from the definitions.

Theorem 6.1 (Linearity of expectation). *Suppose that X, Y, and X_i, where $1 \leq i \leq n$, are random variables defined on a probability space. Let c_i, where $1 \leq i \leq n$, be real numbers. Then*

$$\mathbb{E}\left(\sum_{i=1}^{n} c_i X_i\right) = \sum_{i=1}^{n} c_i \mathbb{E}\left(X_i\right).$$

A basic but very useful inequality on random variables is the following.

Theorem 6.2 (Markov's inequality). *Let $X \geq 0$ be a random variable on a probability space with sample space S. If c is a positive real number, then*

$$\mathbb{P}(X \geq c) \leq \frac{\mathbb{E}(X)}{c}.$$

Proof. The proof follows by the following inequalities:

$$\begin{aligned}\mathbb{E}(X) &= \sum_{s\in S}\mathbb{P}(\{s\})X(s)\\ &\geq \sum_{\substack{s\in S,\\ X(s)\geq c}}\mathbb{P}(\{s\})X(s)\\ &\geq \sum_{\substack{s\in S,\\ X(s)\geq c}}\mathbb{P}(\{s\})c\\ &= c\mathbb{P}(X\geq c). \quad \square\end{aligned}$$

A *binomial random variable* X with parameters n and p, written $X \in Bi(n,p)$, satisfies

$$\mathbb{P}(X=i)=\binom{n}{i}p^i(1-p)^{n-i},$$

where $\mathbb{P}(A)$ stands for the probability of the event A. As a simple but illustrative example, if we toss a fair coin n times and X counts the number of heads, then $X\in Bi(n,\frac{1}{2})$ and the probability of obtaining i heads equals

$$\mathbb{P}(X=i)=\binom{n}{i}\left(\frac{1}{2}\right)^n.$$

If X is a $Bi(n,p)$, then $\mathbb{E}(X)=np$.

For a binomial random variable, the probability of deviating from the mean exponentially tends to 0 the larger the distance from the expectation. This is made precise by the following set of inequalities, called the *Chernoff bound*. For a proof, see Theorem 2.1 and Corollary 2.3 of [**123**].

Theorem 6.3. *Let X be a binomial random variable $X\in Bi(n,p)$ with $\mathbb{E}(X)=np$. If $\varepsilon\leq 3/2$, then*

$$\mathbb{P}\left(|X-\mathbb{E}(X)|\geq\varepsilon\mathbb{E}(X)\right)\leq 2\exp\left(-\frac{1}{3}\varepsilon^2\mathbb{E}(X)\right).$$

We now give a formal definition of random graphs. Define a probability space on graphs of a given order $n\geq 1$ as follows. Fix a vertex set V consisting of n distinct elements, usually taken as $[n]=\{1,2,\ldots,n\}$, and fix $p\in[0,1]$. Define the space of *random* *graphs of order n with edge probability p*, written $G(n,p)$ with sample space equaling the set of all $2^{\binom{n}{2}}$ (labeled) graphs with vertex set V, and

$$\mathbb{P}(G) = p^{|e(G)|}(1-p)^{\binom{n}{2}-|e(G)|}.$$

Informally, we may view $G(n,p)$ as the space of graphs with vertex set V, so that two distinct vertices are joined independently with probability p. Even more informally: toss a (biased) coin to determine the edges of your graph. Hence, V does not change, but the number of edges is not fixed: it varies according to a binomial distribution with expectation $\binom{n}{2}p$. Despite the fact that $G(n,p)$ is a space of graphs, we will abuse language and call it *the random graph of order n with edge probability p*. An alternative and equivalent definition of $G(n,p)$ using product spaces may be found in Exercise 9.

We consider the cases when p is fixed, and when it is a function of n. Graph parameters, such as the cop number, become random variables in $G(n,p)$. For notational ease, if X is a parameter of graphs, we will write $X(G(n,p))$. Hence, we will refer to the cop number of $G(n,p)$ simply by $c(G(n,p))$.

We say that an event holds *asymptotically almost surely* (or *a.a.s.* for short) if it holds with probability tending to 1 as $n \to \infty$. For example, if p is constant, then a.a.s. $G(n,p)$ is diameter 2 and not planar. (See Exercise 2.)

6.2. Constant p and $\log n$ Many Cops

An elementary upper bound for the cop number is

$$c(G) \leq \gamma(G), \tag{6.1}$$

where $\gamma(G)$ is the domination number of G. In general graphs, the inequality (6.1) is far from tight (consider a path, for example). In random graphs $G(n,p)$ with constant p, we will see that both the cop number and domination number equal $\Theta(\log n)$.

The domination number of $G(n,p)$ was first studied in Dreyer's doctoral thesis [**70**]. The following result of [**70**] gave asymptotic bounds for the domination number of a finite random graph with p a fixed constant. For $p \in (0,1)$ or $p = p(n) = o(1)$, define

$$\mathbb{L}n = \log_{\frac{1}{1-p}} n.$$

Theorem 6.4. *Let $0 < p < 1$ be fixed and $q = \frac{1}{1-p}$. For every real $\varepsilon > 0$, a.a.s.*

$$(1-\varepsilon)\mathbb{L}n \le \gamma(G(n,p)) \le (1+\varepsilon)\mathbb{L}n.$$

In particular,

$$\gamma(G(n,p)) = \Theta(\log n).$$

We note that the domination number for random graphs in the more general context when p is a function of n was studied in [**196**]. They proved that a.a.s. $\gamma(G(n,p))$ equals one of two values:

$$\lfloor \mathbb{L}n - \mathbb{L}\big((\mathbb{L}n)(\log n)\big) + 1 \rfloor$$

or

$$\lfloor \mathbb{L}n - \mathbb{L}\big((\mathbb{L}n)(\log n)\big) + 2 \rfloor.$$

The cop number of $G(n,p)$ was studied in [**28**] for constant p, where the following result was proved.

Theorem 6.5. *Let $0 < p < 1$ be fixed. For every real $\varepsilon > 0$, a.a.s.*

$$(1-\varepsilon)\mathbb{L}n \le c(G(n,p)) \le (1+\varepsilon)\mathbb{L}n. \tag{6.2}$$

In particular,

$$c(G(n,p)) = \Theta(\log n).$$

If we consider the case $p = 1/2$, then $G(n,p)$ corresponds to the case of uniformly choosing a labeled graph of order n from the space of all such graphs. Hence, Theorem 6.5 may be interpreted as saying "most" finite graphs of order n have cop number approximately $\log n$. In other words, the cop number of most graphs is fairly small, and is much smaller than Meyniel's bound $O(\sqrt{n})$. One view of Theorem 6.5 supplies some evidence that Meyniel's conjecture is in fact true (although we already know it holds for diameter 2 graphs by Theorem 3.10, and a.a.s. $G(n,p)$ is diameter 2).

The upper bound in (6.2) follows from Theorem 6.4. The proof of Theorem 6.5 thus follows by establishing the lower bound for the cop number of $G(n,p)$. For this, we use an adjacency property. For

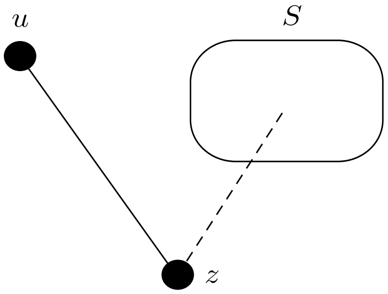


Figure 6.1. The $(1,k)$-e.c. property.

a positive integer k, a graph is $(1,k)$-*existentially closed* (or $(1,k)$-*e.c.*) if for each k-element subset S of vertices of G and vertex u distinct from S, there is a vertex $z \notin S$ not joined to any vertex in S and joined to u. See Figure 6.1. Note that a $(1,k)$-e.c. graph has minimum degree at least $k+1$.

Proof of Theorem 6.5. If G is $(1,k)$-e.c., then $c(G) > k$. To see this, in a given round suppose that there are k cops C. The robber uses the $(1,k)$-e.c. to find a neighboring vertex z not joined or equal to a vertex in C. The vertex acts as an "escape route" for the robber, who can evade capture for one more round.

Let $0 < \varepsilon < 1$ be fixed, and let

$$k = \lfloor (1-\varepsilon)\mathbb{L}n \rfloor.$$

Define $c = \log(\frac{1}{1-p})$, and d by

$$d = \frac{1-\varepsilon}{c}.$$

Then $c, d > 0$ and $0 < cd < 1$. The probability that G is not $(1,k)$-e.c. is at most

$$\begin{aligned}
&n^{k+1}(1-p(1-p)^k)^{n-k-1}\\
&= \exp(\log(n^{k+1}(1-p(1-p)^k)^{n-k-1}))\\
&\leq \exp\left((d\log n + 1)\log(n) + (n - d\log n - 1)\log\left(1 - \frac{p}{n^{cd}}\right)\right)\\
&= o(1),
\end{aligned}$$

where the last line follows since $\log\left(1-\frac{p}{n^{cd}}\right) < 0$ for all n. Hence, a.a.s.

$$c(G) \geq (1-\varepsilon)\mathbb{L}n. \qquad \square$$

As we will see in the next section, $c(G(n,p))$ actually *concentrates on* $\mathbb{L}n$: that is,

$$c(G(n,p)) = (1+o(1))\mathbb{L}n.$$

6.3. Variable p and Bounds

The more difficult problem of determining the cop number of $G(n,p)$, where $p = p(n)$ is a function of n, was left open in [**28**]. We now consider the works of [**19**] and [**31**] on the cop number of $G(n,p(n))$ when $p(n)$ is a function of n. Throughout this section, we will abuse notation and refer to p rather than $p(n)$.

6.3.1. Dense Random Graphs. We now consider the cop number of *dense* random graphs, with average degree pn at least $\sqrt{n}$. The main results of this subsection are summarized in the following theorem, proved in [**31**].

Theorem 6.6 ([**31**]). (1) *Suppose that $p \geq p_0$ where p_0 is the smallest p for which*

$$p^2/40 \geq \frac{\log\left((\log^2 n)/p\right)}{\log n}$$

holds. Then a.a.s.

$$\mathbb{L}n - \mathbb{L}\left((p^{-1}\mathbb{L}n)(\log n)\right) \leq c(G(n,p)) \leq \mathbb{L}n - \mathbb{L}\left((\mathbb{L}n)(\log n)\right) + 2.$$

(2) *If $(2\log n)/\sqrt{n} \leq p = o(1)$ and $\omega(n)$ is any function tending to infinity, then a.a.s.*

$$\mathbb{L}n - \mathbb{L}\left((p^{-1}\mathbb{L}n)(\log n)\right) \leq c(G(n,p)) \leq \mathbb{L}n + \mathbb{L}(\omega(n)).$$

An example of a suitable p in item (1) of the theorem would be $p = (\log n)^{-1/4}$. By Theorem 6.6, we have the following corollary which gives a concentration result for the cop number. In particular, for a wide range of p, the cop number of $G(n,p)$ concentrates on just the one value $\mathbb{L}n$.

Corollary 6.7 ([**31**]). *If* $p = n^{-o(1)}$ *and* $p < 1$, *then a.a.s.*

$$c(G(n,p)) = (1 + o(1))\mathbb{L}n.$$

In Section 6.2 above, we established bounds in Theorem 6.5 for the cop number of $G(n,p)$ when p is constant. From Theorem 6.6 (1) it follows that if p is a constant, then we have the concentration result that

$$c(G(n,p)) = \mathbb{L}n - 2\mathbb{L}\log n + \Theta(1) = (1 + o(1))\mathbb{L}n.$$

We supply an important corollary of Theorem 6.6, which is the first step closer towards the Zig-Zag Theorem (see Theorem 6.16).

Corollary 6.8 ([**31**]). *If* $d = np = n^{\alpha + o(1)}$, *where* $1/2 < \alpha \leq 1$, *then a.a.s.*

$$c(G(n,p)) = \Theta(\log n/p) = n^{1-\alpha+o(1)}.$$

To better understand Corollary 6.8, we define a function $f : (0,1) \to \mathbb{R}$ by

$$f(x) = \frac{\log \overline{c}(G(n, n^{x-1}))}{\log n},$$

where $\overline{c}(G(n, n^{x-1}))$ is the median of the cop number for $G(n,p)$. Hence, the graph of f is the log-log plot of the median of the cop number of $G(n,p)$. Note that for a real number $x \in (0,1)$, the expected degree of $G(n, n^{x-1})$ is

$$pn = nn^{x-1} = n^x.$$

The reader should take a moment and verify that the following straight line with negative slope, which plots the function f, depicts the conclusion of Corollary 6.8. See Figure 6.2. Note that Meyniel's conjecture for random graphs would suggest that the graph of f never goes above the line $y = 1/2$. The remainder of the graph plotted in Figure 6.2 will be more closely examined in the next subsection and in Section 6.4.

Proof of Theorem 6.6. We consider the upper bounds in each of items (1) and (2). The upper bound in (1) is implied by the following result proved in [**196**].

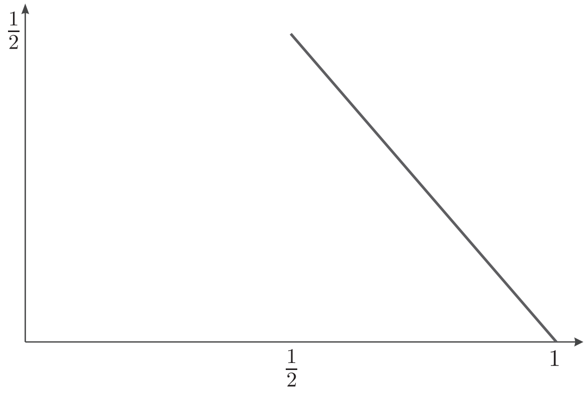


Figure 6.2. The graph of f, so far.

Theorem 6.9. *Suppose that $p \geq p_0(n)$, where p_0 is the smallest p for which*

$$p^2/40 \geq \frac{\log\big((\log^2 n)/p\big)}{\log n}$$

holds. Then a.a.s.

$$\lfloor \mathbb{L}n - \mathbb{L}\big((\mathbb{L}n)(\log n)\big)\rfloor + 1 \leq \gamma(G(n,p)) \leq \lfloor \mathbb{L}n - \mathbb{L}\big((\mathbb{L}n)(\log n)\big)\rfloor + 2.$$

For the upper bound in item (2), the proof follows by the following claim.

Claim 1. If $p = o(1)$ and $\omega(n)$ is any function tending to infinity with n, then a.a.s.

$$\gamma(G(n,p)) \leq \lceil \mathbb{L}n + \mathbb{L}(\omega(n))\rceil\,.$$

To prove Claim 1, we note that since $p = o(1)$ we have that

$$\mathbb{L}n = \frac{\log n}{-\log(1-p)} = (1+o(1))\frac{\log n}{p}\,. \tag{6.3}$$

Let

$$k = \lceil \mathbb{L}n + \mathbb{L}(\omega(n))\rceil\,.$$

If $k > n$, then the claim is trivial, so we assume $k \leq n$. Then the probability that $\gamma(G(n,p)) \leq k$ is bounded from below by the probability that any fixed set of k vertices is a dominating set. But the latter probability is equal to

$$\begin{aligned}
\big(1-(1-p)^k\big)^{n-k} &\geq 1-(n-k)(1-p)^k \\
&\geq 1-n(1-p)^k \\
&\geq 1-n(1-p)^{\mathbb{L}n+\mathbb{L}(\omega(n))} \\
&= 1-\frac{1}{\omega(n)} \\
&= 1-o(1)\,.
\end{aligned}$$

We now consider the lower bounds in items (1) and (2). Both lower bounds in Theorem 6.6 will follow once we prove the following claim.

Claim 2. If $p > (2\log n)/\sqrt{n}$ and

$$k = \big\lfloor \mathbb{L}n - \mathbb{L}\big((p^{-1}\mathbb{L}n)(\log n)\big)\big\rfloor\ , \tag{6.4}$$

then a.a.s. $G \in G(n,p)$ is $(1,k)$-e.c.

To prove Claim 2, assume that $p = o(1)$. Then

$$\begin{aligned}
k &= \mathbb{L}n - \mathbb{L}\left((1+o(1))\frac{\log^2 n}{p^2}\right) \\
&= \mathbb{L}n - 2\mathbb{L}\left((1+o(1))\frac{\log n}{p}\right).
\end{aligned}$$

Fix a k-subset S of vertices and a vertex u not in S. Since edges are chosen independently, the probability that no suitable vertex can be found for this particular S and u is

$$(1-p(1-p)^k)^{n-k-1}. \tag{6.5}$$

Let X be the random variable counting the number of S and u for which no suitable x can be found. By (6.5) and the linearity of expectation we have that

$$\begin{aligned}
\mathbb{E}(X) &= \binom{n}{k}(n-k)\left(1-p(1-p)^k\right)^{n-k-1} \\
&\le n^{k+1}\left(1-\frac{(\mathbb{L}n)(\log n)}{n}\right)^{n(1-(\mathbb{L}n)/n)} \\
&= n^{k+1}\exp\left(-(\mathbb{L}n)(\log n)(1-(\mathbb{L}n)/n)\right)(1+o(1)) \\
&= n^{k+1}\exp\left(-(\mathbb{L}n-(\mathbb{L}n)^2/n)(\log n)(1+o(1))\right) \\
&\le n^{k+1}\exp\left(-\left(k+\frac{2\log\log n}{p}-\frac{2\log^2 n}{p^2 n}\right)(\log n)(1+o(1))\right) \\
&= n^{k+1}\exp\left(-\left(k+\frac{2\log\log n}{p}\right)(\log n)(1+o(1))\right) \\
&= \exp\left((k+1)\log n-\left(k+\frac{2\log\log n}{p}\right)(\log n)(1+o(1))\right) \\
&= o(1),
\end{aligned}$$

where the second inequality follows by (6.3). The proof now follows by Markov's inequality (Theorem 6.2). □

Prałat [**175**] considered the case when the cop number of a random graph is constant.

Theorem 6.10 ([**175**]). *Fix k a positive integer. Let*

$$p = p(n) = 1-\left(\frac{k\log n + a_n}{n}\right)^{\frac{1}{k}}.$$

Then the following items hold.

(1) *If $a_n \to -\infty$, then a.a.s. $c(G(n,p)) \le k$.*

(2) *If $a_n \to a \in \mathbb{R}$, then the probability that $c(G(n,p)) = k$ tends to*

$$1-\exp(-e^{-a}/k!),$$

and $c(G(n,p)) = k+1$, otherwise.

(3) *If $a_n \to \infty$, then a.a.s. $c(G(n,p)) \ge k+1$.*

One byproduct of Theorem 6.10 (2) is that for each integer $k > 0$, there exists a graph with cop number exactly k. Note that the proof of Theorem 6.10 (2) is non-constructive.

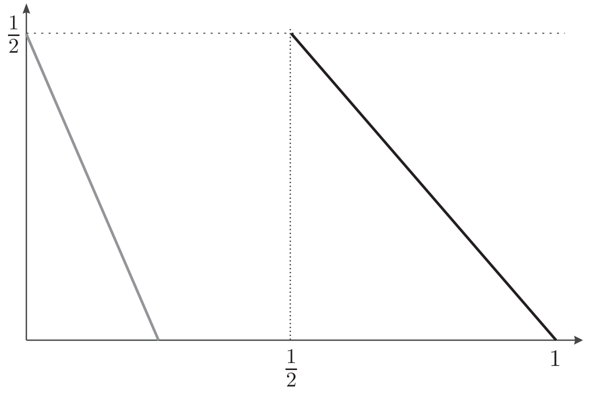


Figure 6.3. Bounds on the graph of f.

6.3.2. Sparse Random Graphs. Recent work by Bollobás, Kun, and Leader [**19**] establishes the following bounds on the cop number in the sparse case, when the expected degree is $np = O(n^{1/2})$.

Theorem 6.11. *If* $p(n) \geq 2.1 \log n/n$, *then a.a.s.*

$$\frac{1}{(np)^2} n^{\frac{1}{2}\frac{\log\log(np)-9}{\log\log(np)}} \leq c(G(n,p)) \leq 160000\sqrt{n}\log n\,. \tag{6.6}$$

In particular, Theorem 6.11 proves Meyniel's conjecture for random graphs, up to a logarithmic factor of n from the upper bound in (6.6). We now know a bit more about the function f from the previous subsection. See Figure 6.3.

What does the rest of the graph of f look like? The surprising answer is in the next section. Read onwards! We sketch a proof of the upper bound of Theorem 6.11.

Lemma 6.12. *Let* $p \in (0,1)$, *and let* k *and* n *be integers such that* $k \leq pn$. *Then the following holds:*

$$\sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i} \leq \exp\left(-\frac{(k-pn)^2}{2pn}\right).$$

For a vertex x and integer $r \geq 0$, define $N_r(x)$ to be the set of vertices of distance at most r from x. To establish an upper bound

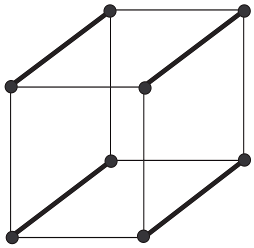


Figure 6.4. A perfect matching (represented by bold edges) in the cube.

for the cop number, we use the strategy of *surrounding the robber*. In other words, if for some r, each vertex of $N_r(R)$ is joined to a unique cop, then the robber cannot leave $N_r(R)$. This amounts to an application of the following famous Hall's theorem (see [**68**] or [**197**]). We note that another application of this theorem will be used in the next section in the proof of the upper bound of the Zig-Zag Theorem; see Theorem 6.16. A *matching* in a graph is a set of edges no two of which share an endpoint, and a matching is *perfect* if it is a spanning subgraph. See Figure 6.4.

For a set S of vertices, define

$$N(S) = \bigcup_{x \in S} N(x).$$

Theorem 6.13. *In a bipartite graph with colors* B_1 *and* B_2, *if for all* $S \subseteq B_1$ *the condition*

$$|S| \leq |N(S)| \tag{6.7}$$

is satisfied, then G *contains a matching of size* $|B_1|$.

We make precise the notion of surrounding the robber in the following lemma. Define

$$N_{r+1}(S) = \bigcup_{x \in S} N_{r+1}(x).$$

Lemma 6.14. *Let T be a set of vertices in a graph G of order n. If for each vertex x of G there is an integer r such that for all $S \subseteq N_r(x)$*

$$|S| \leq |T \cap N_{r+1}(S)|,$$

then $c(G) \leq |T|$.

Proof. To prove the lemma, we need to give a winning strategy with $|T|$ cops. Place the cops on T in round 0. By Theorem 6.13 for every $u \in N_r(R)$ we may assign a distinct cop in $N_{r+1}(R)$. The cops then move to their assigned vertices in at most $r+1$ moves. But then the robber is "surrounded": the robber cannot leave $N_r(R)$ in r rounds, and so is eventually captured there. □

The following expansion-type lemma (whose proof is omitted) follows from Lemma 6.14.

Lemma 6.15. *Let $c > 0$ be an integer. If there is an integer $r > 0$ such that for every $x \in V(G)$ and $S \subseteq N_r(x)$, then the following inequality holds:*

$$10|S| \log n \leq \frac{c}{n}|N_{r+1}(S)|,$$

then $c \geq c(G)$.

Proof of Theorem 6.11. As we mentioned above, we only sketch the proof of the upper bound of (6.6). In fact, we prove the following slightly stronger statement, which implies the upper bound of (6.6). For $\varepsilon \in (0,1)$ and $p > 2(1+\varepsilon)\log n/n$, a.a.s.

$$c(G(n,p)) < \max\left\{\frac{1}{\varepsilon}, 160000\right\}\sqrt{n}\log n.$$

Fix S a set of vertices. Then

$$\mathbb{E}(|N_1(S)|) = |S| + (n - |S|)(1 - (1-p)^{|S|})).$$

By Lemma 6.12, for every S we have that

$$\mathbb{P}\left(||N_1(S)| - \mathbb{E}(|N_1(S)||) \leq \sqrt{2\log n(1+\frac{\varepsilon}{2})(pn+1)|S|}\right)$$
$$\leq \mathbb{P}\left(||N_1(S)| - \mathbb{E}(|N_1(S)||) \leq \frac{4-\varepsilon}{4}(pn+1)\sqrt{|S|}\right)$$

is at least $1-n^{-\frac{2+\varepsilon}{2}|S|}$. Hence, we may assume that these events occur for all S.

Let

$$r = \left\lfloor \frac{\log(1000\sqrt{n})}{\log(pn+2)} \right\rfloor . \tag{6.8}$$

If $|S| \geq (pn+1)^2$, then $|N_1(S)| \leq (pn+2)|S|$. Further, for S of any size,

$$\begin{aligned} |N_1(S)| &\leq (pn+1)|S| + (pn+1)\sqrt{|S|} \\ &\leq 2(pn+2)|S|. \end{aligned}$$

By induction on k, we have that

$$|N_k(S)| \leq 4(pn+2)^k,$$

from which it follows that

$$|N_r(x)| \leq 4000\sqrt{n}.$$

We have that

$$\mathbb{E}(N(T)) > n(1-\exp(-p|T|))$$

for all T. If

$$p|T|(\log n + 1) < 1, \tag{6.9}$$

then

$$\mathbb{E}(N_1(T)) > \left(1 - \frac{1}{\log n}\right) pn|T|.$$

Condition (6.9) holds for $T = N_k(S)$ if $S \subseteq N_r(x)$ for some vertex x and $k \leq r-2$. Note that for n large, if $p|S|(\log n + 1) \leq 1$, then

$$|N_1(S)| \geq \frac{\varepsilon}{5} pn|S|.$$

Further, if $|N_1(S)|p(\log n + 1) < 1$, then for n large

$$|N_2(S)| \geq \frac{\varepsilon}{6} p^2 n^2 |S|.$$

If $3 \leq k \leq r-2$, then

$$|N_{k+1}(S)| \geq \left(1 - \frac{1}{\log n}\right)(pn-1)|N_k(S)|. \tag{6.10}$$

By using (6.10) for $k = 2, 3, \ldots, r-2$, we have that

$$\begin{aligned} |N_{r-1}(S)| &\geq \frac{\varepsilon}{6}\left(1 - \frac{1}{\log n}\right)^{r-1}\left(\frac{pn-1}{pn+2}\right)^{r-1}(pn+2)^{r-1}|S| \\ &= (1+o(1))\frac{\varepsilon}{6}(pn+2)^{r-1}|S|. \end{aligned}$$

Now,

$$|N_r(S)| \geq n(1 - \exp(-p|N_{r-1}(S)|)).$$

For large n, the right-hand side is at least $n/4$ if $p|N_r(S)| > 1/2$, or else at least

$$(2 - 2\exp(-1/2))pn|N_{r-1}(S)| > 1/2(pn+2)|N_r(S)|.$$

It follows that

$$|N_r(S)| \geq \max\{n/4, 1/2(pn+2)|N_{r-1}(S)|\}.$$

In a similar fashion, we have for large n that

$$\begin{aligned} |N_{r+1}(S)| &\geq \max\{n/4, 1/4(pn+2)|N_r(S)|\} \\ &\geq \max\left\{n/4, \frac{\varepsilon}{96}(1+o(1))(pn+2)^{r+1}|S|\right\} \\ &\geq \max\left\{n/4, \frac{\varepsilon}{100}(1+o(1))(pn+2)^{r+1}|S|\right\} \\ &\geq \max\left\{n/4, 10\varepsilon(1+o(1))\sqrt{n}|S|\right\}, \end{aligned}$$

where the last inequality follows by the choice of r in (6.8). The proof now follows by Lemma 6.15:

$$\begin{aligned} c(G) &< 10n\log n \max_{\substack{x\in V,\\ S\subseteq N_r(x)}} \frac{|S|}{|N_{r+1}(S)|} \\ &\leq 10n\log n \max\left\{\frac{1}{10\varepsilon\sqrt{n}}, \frac{4000\sqrt{n}}{n/4}\right\} \\ &= \max\left\{\frac{1}{\varepsilon}, 160000\right\}\sqrt{n}\log n, \end{aligned}$$

as desired. □

6.4. The Zig-Zag Theorem

Based on the results of Section 6.3 it would be natural to assume that the cop number of $G(n,p)$ is close to $\sqrt{n}$ also for $np = n^{\alpha+o(1)}$, where $0 < \alpha < 1/2$. The so-called Zig-Zag Theorem of Łuczak and Prałat [**142**] demonstrates that the actual behaviour of $c(G(n,p))$ is much more complicated.

Theorem 6.16. *Let* $0 < \alpha < 1$*, and* $d = d(n) = np = n^{\alpha+o(1)}$*.*

(1) *If* $\frac{1}{2j+1} < \alpha < \frac{1}{2j}$ *for some* $j \geq 1$*, then a.a.s.*

$$c(G(n,p)) = \Theta(d^j)\,.$$

(2) *If* $\frac{1}{2j} < \alpha < \frac{1}{2j-1}$ *for some* $j \geq 1$*, then a.a.s.*

$$\Omega\left(\frac{n}{d^j}\right) = c(G(n,p)) = O\left(\frac{n\log n}{d^j}\right)\,.$$

With Theorem 6.16 available, for the final time, we come back to the plot of the function $f : (0,1) \to \mathbb{R}$ defined by

$$f(x) = \frac{\log \overline{c}(G(n,n^{x-1}))}{\log n},$$

where $\overline{c}(G(n,n^{x-1}))$ is the median of the cop number for $G(n,p)$. See Figure 6.5, which justifies the theorem's moniker. In particular,

$$f(x) = \left\{ \begin{array}{cl} \alpha j & \text{if } \frac{1}{2j+1} \leq \alpha < \frac{1}{2j} \text{ for some } j \geq 1, \\ 1-\alpha j & \text{if } \frac{1}{2j} \leq \alpha < \frac{1}{2j-1} \text{ for some } j \geq 1. \end{array} \right.$$

A few things become more transparent when studying Figure 6.5. First, there are infinitely many values of x (such as $x = 1/2$, $1/4$, $1/6$, and so on) where $c(G(n,p)) = \Theta(\sqrt{n})$. Second, the cop number is far from being a monotonic function of p: it exhibits increasing oscillation as x tends to 0. Last but not least, it gives an independent verification of the Meyniel bound for $G(n,p)$ random graphs, up to a log factor.

As the proof of the lower bound for the cop number in Theorem 6.16 is more technical, we only give the proof of the upper bound. Nevertheless, for completeness, we now give a high level overview of the proof of the lower bound. For the lower bound, we show that regardless of how the cops move, the robber can move keeping all cops

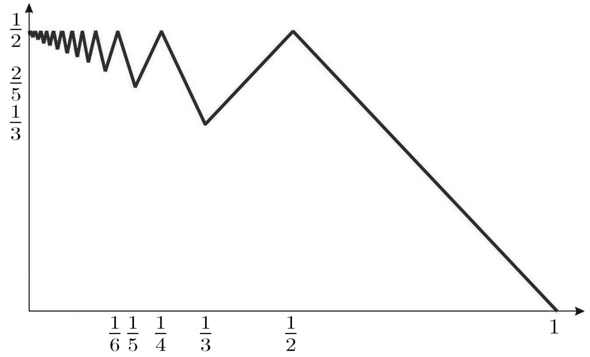


Figure 6.5. The zig-zag-shaped graph of the cop number of $G(n, p)$.

within distance at least one. Moreover, the robber is able to maintain the property that only a small fraction of all neighbors within distance i (where $i \geq 1$) are occupied by a cop. This is enough to set up an inductive proof which ensures that a.a.s. the robber can move indefinitely without capture.

The upper bound for $c(G(n, p))$ in Theorem 6.16 follows from the following theorem from [**142**].

Theorem 6.17. *Let $j \geq 1$, and $d = d(n) = np$.*

(1) *If $n^{1/(2j+1)} \leq d \leq n^{1/(2j)}$ and $\gamma = \lceil n \log n / d^{2j+1} \rceil$, then a.a.s.*

$$c(G(n, p)) = O\big(d^j \gamma\big) .$$

(2) *If $n^{1/(2j+2)} \leq d \leq n^{1/(2j+1)}$, then a.a.s.*

$$c(G(n, p)) = O\Big(\frac{n \log n}{d^{j+1}}\Big) .$$

In order to derive the upper bound for $c(G(n, p))$ in Theorem 6.16, the cops use the following strategy. First, distribute the cops uniformly at random. (The number of cops that are required depends on the parameter p.) We show that regardless of the first move of the robber, the cops can move toward the robber so that eventually the robber is surrounded, and is captured after another few moves.

As in the proof of Lemma 6.14, the proof here heavily relies on Hall's bipartite matching theorem, which is Theorem 6.13.

We need the following lemma, whose proof is left as an exercise (see Exercise 7).

Lemma 6.18. *Let* $0 < \alpha < 1$, *and* $d = d(n) = np = n^{\alpha+o(1)}$. *Then a.a.s. for every* $1 \leq i \leq 1/(2\alpha)$ *and vertices* $v_1, v_2, \ldots, v_k$ *we have that*

$$\left| \bigcup_{j=1}^{k} N_{i+1}(v_j) \right| \geq 0.5 \min\{k(0.1d)^{i+1}, n\}.$$

Proof of Theorem 6.17. We only sketch the proof of (1) as the proof of item (2) is similar. We play with

$$\beta n = 5000(10d)^j \gamma$$

many cops, and show they a.a.s. have a winning strategy in $G(n,p)$. Suppose that to each vertex u of $N_j(R)\backslash N_{j-1}(R)$ we can assign a unique cop in $N_{j+1}(u)$. Then the cops can move to these vertices u, and so after at most $j+1$ rounds, the robber is surrounded in $N_j(R)$. See Figure 6.6. Then the robber is eventually captured there.

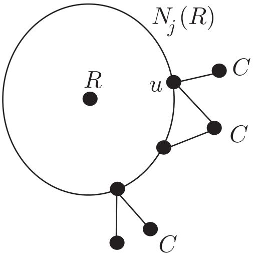


Figure 6.6. Surrounding the robber.

To be able to assign a cop as in the previous paragraph a.a.s., we apply Hall's theorem. Fix a vertex v and $S \subseteq N_j(v)\backslash N_{j-1}(v)$ with $|S| = k$. Let

$$k_0 = \max\{k : (0.1d)^{j+1}k < n\}.$$

By Lemma 6.18, if $k \le k_0$, then the number of cops occupying

$$\bigcup_{u \in S} N_{i+1}(u)$$

is bounded below by the *Bernoulli random variable* $B(M, \beta)$ (that is, the random variable takes on values 1 or 0 with probabilities β or $1 - \beta$, respectively) where $M \ge 0.1(0.1d)^{j+1}k$. In particular, the expectation of $B(M, \beta)$ is just $M\beta$, and

$$M\beta \ge 50k \log n.$$

By the Chernoff bound (see Theorem 6.3), it follows that the probability that there are fewer than k cops in $\bigcup_{u \in S} N_{j+1}(u)$ is less than $\exp(-4k \log n)$. Now, since

$$\begin{aligned} \sum_{k=1}^{k_0} \binom{|N_j(v)|}{k} \exp(-4k \log n) &\le \sum_{k=1}^{n} n^k \exp(-4k \log n) \\ &= O(n^{-2}) \end{aligned}$$

with probability $1 - O(n^{-2})$, condition (6.7) is satisfied for all sets of cardinality at most k_0.

If $k_0 \le k \le |N_j(v)| \le 2d^j$, then the Chernoff bound implies that the number of cops in $\bigcup_{u \in S} N_{j+1}(u)$ is at least

$$\frac{1}{4} n\beta \ge 50d^j > |N_j(v)|$$

with probability at least $1 - \exp(-4d^j)$. As

$$\begin{aligned} \sum_{k=k_0+1}^{|N_j(v)|} \binom{|N_j(v)|}{k} \exp(-4d^j) &\le 2d^j 2^{2d^j} \exp(-4d^j) \\ &= O(n^{-2}), \end{aligned}$$

condition (6.7) is satisfied with probability $1 - O(n^{-2})$ in the case $R = v$. In a similar fashion, a.a.s. there is a perfect matching between $N_{i+1}(R) \backslash N_i(R)$ and $N_i(R) \backslash N_{i-1}(R)$ for all $1 \le i \le j$. In particular, the cops can a.a.s. continue to surround the robber and capture him in j moves. □

More work on the cop number of random graphs remains to be done. The behaviour of the cop number is still unknown for sparse graphs (close to the connectivity threshold) or for the giant component of $G(n, p)$ below this threshold. It would be interesting also to determine the cop number of random regular graphs.

One approach is to consider properties of almost all k-cop-win graphs; that is, considering properties of randomly chosen k-cop-win graphs (with the uniform distribution). For this, it is equivalent to work in the probability space $G(n, 1/2)$. Let **cop-win** be the event in $G(n, 1/2)$ that the graph is cop-win, and let **universal** be the event that there is a universal vertex. If a graph has a universal vertex w, then it is cop-win; in a certain sense, graphs with universal vertices are the simplest cop-win graphs. The probability that a random graph is cop-win can be estimated as follows:

$$\begin{aligned}\mathbb{P}(\textbf{cop-win}) \geq \mathbb{P}(\textbf{universal}) &= n2^{-n+1} - O(n^2 2^{-2n+3}) \\ &= (1 + o(1))n2^{-n+1}.\end{aligned}$$

A recent surprising result of [**29**] shows this lower bound is in fact the correct asymptotic value for $\mathbb{P}(\textbf{cop-win})$.

Theorem 6.19 ([**29**]). *In $G(n, 1/2)$, we have that*

$$\mathbb{P}(\textbf{cop-win}) = (1 + o(1))n2^{-n+1}.$$

Hence, almost all cop-win graphs contain a universal vertex. This is not obvious on first glance!

Corollary 6.20 ([**29**]).

$$\mathbb{P}(\textbf{universal} \mid \textbf{cop-win}) = 1 - o(1).$$

For $k > 1$, it is conjectured in [**29**] that almost all k-cop-win graphs in fact have a dominating set of cardinality k.

6.5. Cops and Robbers in the Web Graph

Real-world complex networks are active topics of investigation by both mathematicians and empiricists alike. Perhaps the most famous example of such a network is the web graph, where vertices correspond to web pages and edges correspond to hyperlinks. The web graph has over one trillion vertices, with many billions of pages being added each day. Complex networks range from on-line social networks such as the friendship graph in Facebook, to the protein-protein interaction networks in living cells. For more discussion on complex networks, see the books [**21**] and [**49**].

One of the most important and well-studied properties observed in the web graph and complex networks are power-law degree distributions. Given an undirected graph G and a non-negative integer k, we define $N_{k,G}$ by

$$N_{k,G} = |\{x \in V(G) : \deg_G(x) = k\}|.$$

The *degree distribution* of G is the sequence $(N_{k,G} : 0 \le k \le t)$, where t is the order of G. We say that the degree distribution of G follows a *power law* if for degrees $k > 0$,

$$\frac{N_{k,G}}{t} \sim k^{-\beta}, \tag{6.11}$$

for a fixed real constant $\beta > 1$ called the *exponent*. If G possesses a power law degree distribution, then we simply say G is a *power law graph*. If we take logarithms on both sides of (6.11), then the relationship is expressed as

$$\log(N_{k,G}) \sim \log(t) - \beta \log(k).$$

Hence, in the log-log plot, we obtain a straight line with slope $-\beta$. See Figure 6.7.

Based on their crawl of the domain of Notre Dame University, Albert, Barabási, and Jeong [**4**] claimed that the web graph exhibited a power law in-degree distribution, with $\beta = 2.1$. An independent crawl corroborating the findings of [**4**] was reported from IBM researchers [**129**], which considered 40 million web pages from 1997 data. The exponent of $\beta = 2.1$ was further corroborated by a larger crawl of the entire web (including 200 million web pages) reported in Broder et al. [**35**]. Power laws are now known to be pervasive in a variety of real-world networks such as on-line social networks such as the friendship graphs in Facebook, LinkedIn, and Twitter (see [**25**]), and in biological networks such as protein-protein interaction networks (see Chapter 2 of [**21**]).

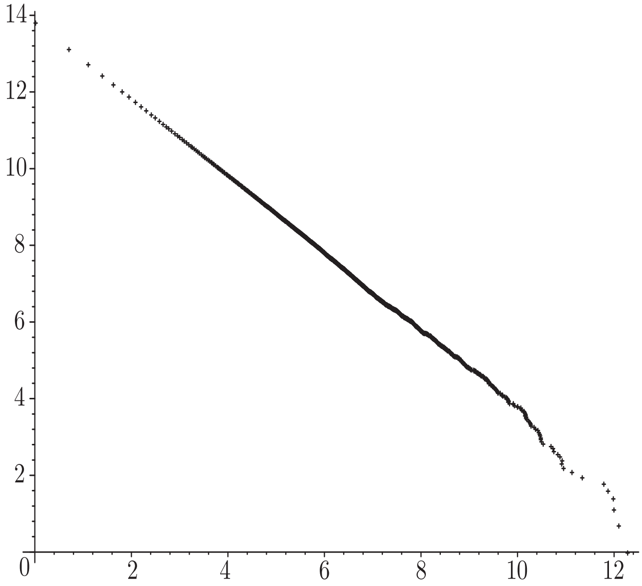


Figure 6.7. The log-log plot of the degree distribution of a power law graph.

We now consider a model for the web graph proposed by Chung and Lu; see [**49**]. Let

$$\mathbf{w} = (w_1, \dots, w_n)$$

be a sequence of n real non-negative real numbers. We define a random graph model, written $G(\mathbf{w})$, whose vertices are integers in $[n]$. Each potential edge between i and j is chosen independently with probability $p_{ij} = w_i w_j \rho$, where

$$\rho = \frac{1}{\sum_{i=1}^{n} w_i}.$$

We will always assume that

$$\max_i w_i^2 < \sum_{i=1}^{n} w_i,$$

which implies that $p_{ij} \in [0, 1)$ (see Exercise 11). The model $G(\mathbf{w})$ is referred to as *random graphs with given expected degree sequence*

w. Observe that $G(n,p)$ may be viewed as a special case of $G(\mathbf{w})$ by taking **w** to be equal to the constant n-sequence

$$(pn, pn, \dots, pn).$$

Given $\beta > 2$, $d > 0$, and a function $M = M(n) = o(\sqrt{n})$ (with M tending to infinity with n), we consider the random graph with given expected degrees $w_i > 0$, where

$$w_i = ci^{-\frac{1}{\beta-1}} \tag{6.12}$$

for i satisfying $i_0 \leq i < n + i_0$. The term c depends on β and d, and i_0 depends also on M; namely,

$$c = \left(\frac{\beta-2}{\beta-1}\right) dn^{\frac{1}{\beta-1}}, \qquad i_0 = n\left(\frac{d}{M}\left(\frac{\beta-2}{\beta-1}\right)\right)^{\beta-1}. \tag{6.13}$$

As shown in [**49**], a.a.s. random graphs with the expected degrees satisfying (6.12) and (6.13) follow a power law degree distribution with exponent β, average degree $(1 + o(1))d$, and maximum degree $(1 + o(1))M$. So-called *random power law graphs* are an important example of an off-line web graph model; see [**21, 49**] for more details.

One of the main results of this section, proved in [**31**], is that the cop number of random power law graphs is a.a.s. $\Theta(n)$. Hence, these results are suggestive that in power law graphs, on average a large number of cops are needed to secure the network.

Theorem 6.21 ([**31**]). *For a random power law graph $G(\mathbf{w})$ with exponent $\beta > 2$ and average degree d, a.a.s. the following hold.*

(1) *If X is the random variable denoting the number of isolated vertices in $G(\mathbf{w})$, then*

$$\begin{aligned} c(G(\mathbf{w})) &\geq X \\ &= (1+o(1))n\int_0^1 \exp\left(-d\frac{\beta-2}{\beta-1}x^{-1/(\beta-1)}\right)dx. \end{aligned}$$

(2) *For $a \in (0,1)$, define*

$$f(a) = a + \int_a^1 \exp\left(-d\frac{\beta-2}{\beta-1}a^{(\beta-2)/(\beta-1)}x^{-1/(\beta-1)}\right)dx.$$

Then

$$c(G(\mathbf{w})) \leq (1+o(1))n \min_{0<a<1} f(a).$$

See Exercise 12 for more on the integrals in Theorem 6.21. While Theorem 6.21 suggests a large number of cops are needed to secure complex networks against intruders, by item (1) it is the abundance of isolated vertices that makes the cop number equal to $\Theta(n)$. To overcome the issue with isolated vertices, we consider restricting the movements of the cops and robber to the subgraph induced by sufficiently high degree vertices.

Fix $\beta \in (2,3)$. Define the *dense-core* of a graph G, written $\widehat{G}$, as the subgraph induced by the set of vertices of degree at least $n^{1/\log\log n}$. Random power law graphs with $\beta \in (2,3)$ are referred to as *octopus* graphs in [**49**], since the dense-core has low diameter $O(\log\log n)$, while the overall diameter is $O(\log n)$. Since the expected degree of vertex i in $G(\mathbf{w})$ is

$$w_i = \frac{\beta-2}{\beta-1} d n^{1/(\beta-1)} i^{-1/(\beta-1)},$$

vertices with expected degree at least $n^{1/\log\log n}$ have labels at most

$$i_N = \left(\frac{\beta-2}{\beta-1} d\right)^{\beta-1} n^{1-(\beta-1)/\log\log n}.$$

The order of the dense-core is written N. By the Chernoff bound,

$$N = (1+o(1))i_N - i_0 = (1+o(1))i_N = \Theta(n^{1-(\beta-1)/\log\log n}),$$

provided that $\log M \gg (\log n)/\log\log n$. Thus,

$$n = N^{1+(\beta-1)/\log\log N + \Theta(1)/\log^2\log N}. \tag{6.14}$$

We consider the cop number of the dense-core of random power law graphs. As vertices in the dense-core informally represent the *hubs* of the network, you might think that the cop number of the dense-core is of smaller order than the dense-core itself. This intuition is made precise by the following theorem, which provides a sublinear upper bound for the cop number of the dense-core.

Theorem 6.22 ([**31**]). *For a random power law graph $G(\mathbf{w})$ with power law exponent $\beta \in (2,3)$ a.a.s.*

$$\begin{aligned} c(\widehat{G(\mathbf{w})}) &\le N^{1-(1+o(1))(\beta-1)(3-\beta)/(\beta-2)\log\log n} \\ &= o(N). \end{aligned}$$

Proof of Theorem 6.21. For the lower bound, we exploit the fact that the cop number is bounded from below by the number of isolated vertices: one cop is needed per isolated vertex. In general power law graphs, there may exist an abundance of isolated vertices, even as much as $\Theta(n)$ many. We show that this is the case for random power law graphs.

The probability that the vertex i for $i_0 \leq i < n + i_0$ (that is, the vertex i corresponds to the expected degree w_i) is isolated is equal to

$$\begin{aligned} p_i &= \prod_{j, j \neq i} (1 - w_i w_j \rho) \\ &= \prod_{j, j \neq i} \exp\left(-w_i w_j \rho (1 + o(1))\right) \\ &= \exp\left(-w_i \rho \sum_{j, j \neq i} w_j (1 + o(1))\right) \\ &= \exp\left(-w_i (1 + o(1))\right). \qquad (6.15) \end{aligned}$$

Let X_i be the indicator random variable for the event that the vertex i is isolated. Then

$$\mathbb{P}(X_i = 1) = 1 - \mathbb{P}(X_i = 0) = p_i$$

for $i_0 \leq i < n + i_0$.

Let X be the number of isolated vertices in $G(\mathbf{w})$. As

$$X = \sum_{i_0 \leq i < n + i_0} X_i,$$

it follows from (6.15) that the expected value of X is

$$\begin{aligned} \sum_{i_0 \leq i < n + i_0} p_i &= (1 + o(1)) n \int_0^1 \exp\left(-c(xn)^{-1/(\beta - 1)}\right) dx \\ &= (1 + o(1)) n \int_0^1 \exp\left(-d \frac{\beta - 2}{\beta - 1} x^{-1/(\beta - 1)}\right) dx\,. \end{aligned}$$

By Theorem 6.3, the number of isolated vertices in $G(\mathbf{w})$ is a.a.s. equal to

$$X = (1 + o(1)) n \int_0^1 \exp\left(-d \frac{\beta - 2}{\beta - 1} x^{-1/(\beta - 1)}\right) dx.$$

Hence, item (1) of Theorem 6.21 follows.

For the proof of the upper bound in item (2) of Theorem 6.21, we give a bound on the domination number. Fix a constant $a \in (0,1)$ and consider the set $A \subseteq V$ of the first $\lfloor an \rfloor - i_0 + 1 = (1+o(1))an$ vertices; that is,

$$A = \{i_0, i_0+1, \ldots, \lfloor an \rfloor\}.$$

Let $B \subseteq V \setminus A$ denote the set of vertices that do not have a neighbor in A. Then $D = A \cup B$ is a dominating set of G, and we estimate the cardinality of D.

Consider the vertex i, where $an < i < n + i_0$. Since $i_0 = o(n)$, there is $b \in (0,1]$ such that $i = (1+o(1))bn$. The probability that i does not have a neighbor in A is equal to

$$\begin{aligned} q_i &= \prod_{j \in A} (1 - w_i w_j \rho) \\ &= \exp\left(-(1+o(1)) w_i \rho \sum_{j < an - i_0} w_j \right) \\ &= \exp\left(-(1+o(1)) c(bn)^{-1/(\beta-1)} (dn)^{-1} n \int_0^a c(xn)^{-1/(\beta-1)} dx \right) \\ &= \exp\left(-(1+o(1)) d \Big(\frac{\beta-2}{\beta-1}\Big)^2 b^{-1/(\beta-1)} \int_0^a x^{-1/(\beta-1)} dx \right) \\ &= \exp\left(-(1+o(1)) d \frac{\beta-2}{\beta-1} b^{-1/(\beta-1)} a^{(\beta-2)/(\beta-1)} \right). \end{aligned}$$

Define

$$I(a) = \int_a^1 \exp\left(-d \frac{\beta-2}{\beta-1} a^{(\beta-2)/(\beta-1)} x^{-1/(\beta-1)} \right) dx.$$

Thus, using Chernoff's bound (Theorem 6.3), we obtain that a.a.s.

$$|B| = (1+o(1)) n I(a),$$

and that a.a.s.

$$\begin{aligned} |D| &= |A \cup B| \\ &= (1+o(1)) n (a + I(a)). \end{aligned}$$

As this holds for every $a \in (0,1)$, the proof of item (2) follows. □

Proof of Theorem 6.22. We give an upper bound for $c(\widehat{G(\mathbf{w})})$ by using a dominating set. The probability that there is an edge between two given vertices in the dense-core is at least

$$\begin{aligned} p_{\min} &= w^2_{i_N(1+o(1))}\rho \\ &= (n^{2/\log\log n}/dn)(1+o(1)) \\ &= (N^{(3-\beta)/\log\log N+\Theta(1)/\log^2\log n})/N \qquad (6.16) \\ &= (N^{(1+o(1))(3-\beta)/\log\log N})/N\,. \end{aligned}$$

Hence, $\widehat{G(\mathbf{w})}$ contains a random graph $G(N, p_{\min})$. Let $\omega(N)$ be any function tending to infinity with N. Thus, using Claim 1 in the proof of Theorem 6.9, any set of cardinality

$$\begin{aligned} k_N &= \left\lceil \log_{1/(1-p_{\min})} N + \log_{1/(1-p_{\min})} \omega(N) \right\rceil \\ &= (1+o(1))\frac{\log N}{p_{\min}} \\ &= \frac{N\log N}{N^{(1+o(1))(3-\beta)/\log\log N}} \\ &= N\exp\left(-(1+o(1))(3-\beta)(\log N)/\log\log N + \log\log N\right) \\ &= N\exp\left(-(1+o(1))(3-\beta)(\log N)/\log\log N\right) \\ &= N/N^{(1+o(1))(3-\beta)/\log\log N} \\ &= o(N) \end{aligned}$$

is a dominating set a.a.s. As this holds for any set of cardinality k_N, we obtain a smaller dominating set by considering only vertices with the largest expected degree. Consider the subset of vertices $U = \{i_0, i_0+1, \dots, k\}$ of the first $k - i_0 + 1$ vertices, $k \gg i_0$. Then

$$\sum_{i=i_0}^{k} \omega_i = c\int_{i_0}^{k} i^{-1/(\beta-1)}di + O(1) = (1+o(1))c\frac{\beta-1}{\beta-2}k^{(\beta-2)/(\beta-1)}.$$

Hence, the probability that vertex i does not have a neighbor in U is equal to

$$\begin{aligned} q(i) &= \prod_{j=i_0}^{k} (1 - \omega_i \omega_j \rho) \\ &= \exp\left(-(1+o(1))\omega_i \rho \sum_{j=i_0}^{k} \omega_j \right) \\ &= \exp\left(-(1+o(1)) \frac{\beta-2}{\beta-1} dn^{(3-\beta)/(\beta-1)} i^{-1/(\beta-1)} k^{(\beta-2)/(\beta-1)} \right). \end{aligned}$$

It is straightforward to see that for all vertices i in the dense-core,

$$\begin{aligned} q(i) &\leq q\big(i_N(1+o(1))\big) \\ &= \exp\left(-(1+o(1)) n^{(2-\beta)/(\beta-1)+1/\log\log n} k^{(\beta-2)/(\beta-1)} \right). \end{aligned}$$

Therefore, in order to make the right-hand side of the latter equation equal to $o(n^{-1})$, it is enough to take

$$\begin{aligned} k &= n^{1-(\beta-1)/(\beta-2)\log\log n} \log^{2(\beta-1)/(\beta-2)} n \\ &= n^{1-(1+o(1))(\beta-1)/(\beta-2)\log\log n} \\ &= N^{1-(1+o(1))(\beta-1)(3-\beta)/(\beta-2)\log\log n}. \end{aligned}$$

Now, the expected number of vertices that are not dominated by U is $o(1)$, and the assertion follows from Markov's inequality. The upper bound now follows. □

An interesting open problem is to determine the asymptotic value of the cop number of the dense-core of random power law graphs. In [**31**], it was shown that

$$c(\widehat{G(\mathbf{w})}) \geq N^{(1+o(1))(3-\beta)/\log\log N},$$

and so the bounds we have are not tight. Of course, many other models for the web graph and complex networks have been introduced, such as the preferential attachment and copying models. See Chapter 4 of [**21**], and [**49**] for an overview of such models. An intriguing direction of research would be estimating the cop number in models for complex networks.

Exercises

1. Prove Theorem 6.1.
2. Prove the following properties hold a.a.s. in $G(n,p)$, where p is a constant.
   (a) For a fixed integer $m > 0$, the *m-e.c. property*: for all disjoint sets of vertices A and B with $|A \cup B| = m$ (one of A or B may be empty), there is a vertex z not in $A \cup B$ such that z is joined to each vertex of A and to no vertex of B.
   (b) Diameter 2.
   (c) Non-planar.
   (d) Contains a complete graph of order $\log n / \log(1/p)$.
   (e) The degree of each vertex is $pn + O(\sqrt{pn}\log n)$.
   (f) All independent sets have less than $n^{1/3}$ vertices.
3. Define the probability space $G(\mathbb{N}, p)$ to be graphs with vertex set of positive integers, and each distinct pair of integers is joined independently with probability p. We will call this space *the infinite random graph.* Determine the cop number of $G(\mathbb{N}, p)$.
4. For a random variable X, define its *variance* by
$$\mathrm{Var}(X) \quad = \quad \mathbb{E}(X^2) - \mathbb{E}(X)^2.$$
   (a) Derive *Chebyshev's inequality*: if $c > 0$ is a real number, then
$$\mathbb{P}(|X - \mathbb{E}(X)| \geq c) \leq \mathrm{Var}(X)/c^2.$$
   (b) (*The second moment method*) Show that if
$$\mathrm{Var}(X) = o(\mathbb{E}(X)^2),$$
   then a.a.s. $X > 0$.
   (c) Prove that if $1/(np) = o(1)$, then a.a.s. $G(n,p)$ contains a triangle. (*Hint*: Use the second moment method.)
5. Show that if $X \in Bi(n,p)$, then $\mathbb{E}(X) = np$ and
$$\mathrm{Var}(X) = np(1-p).$$
6. For each integer $k > 0$, give explicit examples of $(1,k)$-e.c. graphs.
7. [**142**] Prove Lemma 6.18.

8. (a) Prove Hall's theorem, Theorem 6.13. (*Hint*: Prove the contrapositive.)

   (b) Show that a d-regular bipartite graph contains a perfect matching.

9. This exercise establishes an alternative, formal definition to $G(n, p)$. Given a finite set of probability spaces

$$\{(S_i, \mathcal{F}_i, \mathbb{P}_i) : 1 \leq i \leq m\},$$

we may form the *product space* $(S, \mathcal{F}, \mathbb{P})$ with sample space $\prod_{i=1}^m S_i$, events $\prod_{i=1}^m \mathcal{F}_i$, and for an event $A = \prod_{i=1}^m A_i$,

$$\mathbb{P}(A) = \prod_{i=1}^{m} \mathbb{P}_i(A_i).$$

   (a) Prove that the product space is a probability space.

   (b) Fix n a positive integer, $p \in [0, 1]$, and $i \in [n]^2$. Let $S_i = \{0_i, 1_i\}$ and $\mathbb{P}_i(\{1_i\}) = p$ and $\mathbb{P}_i(\{0_i\}) = 1 - p$. Verify that for a fixed i this gives rise to a probability space $(S_i, \mathcal{F}_i, \mathbb{P}_i)$. The corresponding product space over all $i \in [n]^2$, written $G(n, p)$, is called *a random graph with n vertices and edge probability p.*

   (c) For $i \in [n]^2$, define E_i to be the event consisting of the set of $s \in S$ whose ith coordinate is 1_i. Prove that the events $\{E_i : i \in [n]^2\}$ are mutually independent, and $\mathbb{P}(E_i) = p$. We may therefore identify the probability space $G(n, p)$ with graphs whose vertex set is $[n]$, and whose edges are chosen independently with probability p.

10. Let $0 < d < 1$ and $p = d/n$. Prove that $c(G(n, p)) = \Theta(n)$.

11. Verify that if $\max_i w_i^2 < \sum_{i=1}^n w_i$ in $G(\mathbf{w})$, then $p_{ij} \in [0, 1)$.

12. Show that

$$\int_0^1 \exp\left(-d\frac{\beta-2}{\beta-1}x^{-1/(\beta-1)}\right) dx$$
$$= (d(\beta-2))^{\beta-1}(\beta-1)^{2-\beta}\Gamma\left(1-\beta, d\frac{\beta-2}{\beta-1}\right),$$

where $\Gamma(\cdot, \cdot)$ is the incomplete gamma function.

13. The *Iterated Local Transitivity (ILT) model* has found application in the study of on-line social networks; see [**25**]. The ILT model deterministically generates finite, simple, undirected graphs $(G_t : t \geq 0)$. The only parameter of the model is the initial graph G_0, which is a fixed finite graph. Assume that for a fixed $t \geq 0$, the graph G_t has been constructed. To form G_{t+1}, for each vertex $x \in V(G_t)$, add its *clone* x', such that x' is joined to x and all of its neighbors at time t.
    (a) Find exact formulas for $|V(G_t)|$ and $|E(G_t)|$ for $t > 0$.
    (b) Show that for all $t \geq 0$, $\gamma(G_t) = \gamma(G_0)$.
    (c) Show that for all $t \geq 0$, $c(G_t) = c(G_0)$.

14. Using the probabilistic method, show that a graph G of order n contains a dominating set with cardinality at most
$$n\left(\frac{1+\log(\delta+1)}{\delta+1}\right).$$

15. [**175**] For a fixed positive integer k, let
$$p < 1 - \left(\frac{(k+1)\log n + \log\log n}{n}\right)^{\frac{1}{k}}.$$
Show that a.a.s. $c(G(n,p)) > k$. (*Hint*: Derive that a.a.s. $G(n,p)$ is $(1,k)$-e.c.)

16. The *random tournament of order n with edge probability* $p \in (0,1)$, written $T(n,p)$, has vertices $[n]$, and if $1 \leq i < j \leq n$ an arc (i,j) is chosen independently with probability p; otherwise, the (j,i) is chosen (with probability $1-p$). Determine the asymptotic value of $c(T(n,p))$, for p a constant.

17. For positive integers m and n, and $p \in (0,1)$, define the random bipartite graph $G(m,n,p)$ to have $m+n$ vertices partitioned into an m-set and n-set, so each pair u and v of different colors is joined independently with probability p.

    Determine the asymptotic value of $c(G(m,n,p))$, for p a constant.

Chapter 7

Infinite Graphs

7.1. Introduction

Infinite graphs exhibit properties often quite different than finite ones. In this regard, the cop number is no exception. For example, the *ray* (that is, the one-way infinite path) has infinite cop number: one robber can always stay ahead of finitely many cops. See Figure 7.1. Infinite trees—unlike their finite counterparts—need not be cop-win.

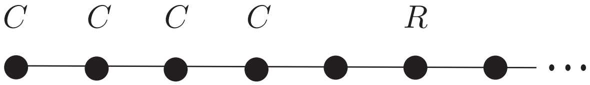


Figure 7.1. Finitely many cops can always be evaded on a ray.

Perhaps not surprisingly, less work has been done on infinite cop-win graphs. We survey that work in this chapter. Along the way, we will see that basic questions about the cop number of infinite graphs remain mostly unanswered. For instance, what can be said about cop-win infinite graphs? Can we characterize graphs with infinite cop number? What are sufficient conditions that force an infinite cop number? How are vertex-transitive k-cop-win graphs behaved?

One of the most influential infinite graphs is the infinite random graph, written $\mathcal{R}$. (Usually, the infinite random graph is written R, which conflicts with our notation for the robber. For this reason, we use the notation $\mathcal{R}$.) While the cop number of $\mathcal{R}$ is simple to calculate (it is countably infinite), consideration of $\mathcal{R}$ and its limit structure gives rise to the notion of cop density. We introduce $\mathcal{R}$ in Section 7.2, and cop density in Section 7.3. Two results from [**28**] stand out here: The first is that the cop density of $\mathcal{R}$ (and more generally, any graph satisfying a certain adjacency property) can be any real number in $[0, 1]$; see Theorem 7.9. The second is the unexpected connection between cop density of any graph and spanning subgraphs of $\mathcal{R}$; see Theorem 7.11.

Finite chordal graphs are cop-win, owing to their simplicial vertices. However, as proven in Section 7.4, there are examples of infinite chordal diameter 2 graphs which are robber-win. The results in this section originate from the paper of [**104**].

We finish with a discussion of vertex-transitive cop-win graphs. Recall that a graph G is *vertex-transitive* if for all pairs of vertices u and v of G, there is an automorphism f of G, so that $f(u) = v$. Roughly, any two vertices behave the same in vertex-transitive graphs. See Figure 7.2 for the isomorphism types of vertex-transitive connected graphs of order 7. Once again, things break down in the infinite case. Finite vertex-transitive cop-win graphs are cliques, as observed first in [**167**] (see Exercise 3 in Chapter 1). Based on work in [**27**], we show that for all infinite cardinals κ, there are 2^κ-many non-isomorphic cop-win vertex-transitive graphs. To put this into perspective, there are 2^κ-many isotypes of graphs of order κ.

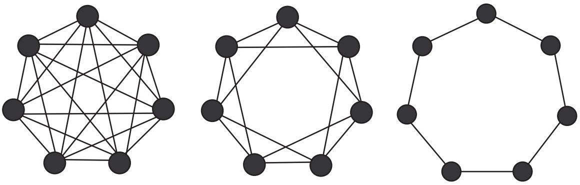


Figure 7.2. The connected vertex-transitive graphs of order 7.

As an ending to this introduction, we note that for infinite bipartite cop-win graphs, we can say quite a bit. As a warm-up, we prove the following theorem characterizing bipartite cop-win graphs in all cardinalities.

Theorem 7.1. *For a graph G, the following are equivalent.*

(1) *The graph G is a cop-win tree.*

(2) *The graph G is a rayless tree.*

(3) *The graph G is cop-win and bipartite.*

Proof. It is evident that a tree containing a ray has infinite cop number, and a rayless tree is cop-win. So we show only that items (1) and (3) are equivalent, with the direction (1) implying (3) being trivial. For direction (3) implying (1) suppose that G is cop-win and bipartite. Deleting an end-vertex leaves another cop-win and bipartite graph. We may therefore sequentially delete end-vertices (possibly requiring a transfinite sequence of such deletions), resulting in another cop-win and bipartite graph G' with no end-vertices. Now consider the second-to-last move of the cop before the robber is captured. We have that R is a corner, so $N[R] \subseteq N[C]$. If the vertex R has a neighbor $x \neq C$, then C is joined to x, resulting in a triangle. Hence, R is only joined to C and so is an end-vertex. But G' has no end-vertices, which is a contradiction. Hence G' is empty, and G is a tree. □

7.2. Introducing the Infinite Random Graph

Something interesting happens to $G(n, p)$ random graphs when you allow a (countably) infinite set of vertices. Define the probability space $G(\mathbb{N}, p)$ to be graphs with vertex set of non-negative integers, and each distinct pair of integers is joined independently with probability p. We will call this space *the infinite random graph* or the Rado graph, after Richard Rado who wrote an important paper [**180**] on its properties. Erdős and Rényi [**75**] discovered the following beautiful theorem about the infinite random graph, which demonstrates once again the radical difference often evident when we move from the finite to infinite.

Theorem 7.2. *With probability* 1, *the graph* $G(\mathbb{N}, p)$ *is unique up to isomorphism.*

The proof of Theorem 7.2 relies on what is called the *back-and-forth* method, introduced by Cantor in his proof that the rationals are the unique isomorphism type of countable linear order which is dense with no endpoints [**42**]. We first show that $G(\mathbb{N}, p)$ satisfies a certain adjacency property with probability 1. Define a graph to be *existentially closed* or *e.c.* if for all finite disjoint sets of vertices A and B (one of which may be empty), there is a vertex z joined to all of A and to no vertex of B. We say that z is *correctly joined to* A and B. See Figure 7.3 for a visualization of the e.c. property.

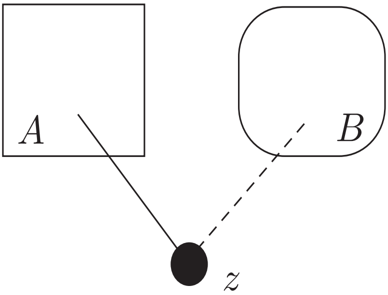


Figure 7.3. The e.c. property.

Theorem 7.3. *With probability* 1, $G(\mathbb{N}, p)$ *is e.c.*

Proof. Consider disjoint sets of vertices A and B in G with $|A| = i$ and $|B| = j$. For a given $z \notin A \cup B$, the probability that z is not correctly joined to A and B is

$$1 - p^i(1-p)^j.$$

The probability that no vertex of G is correctly joined to A and B is therefore

$$\lim_{n\to\infty} \left(1 - p^i(1-p)^j\right)^{n-(i+j)} = 0.$$

The proof follows as there are only countably many choices of A and B, and the countable union of probability 0 events has probability 0. □

In passing, we note that it is not difficult to show that an e.c. graph has diameter 2, with each vertex of infinite degree. The following theorem, along with Theorem 7.3, proves Theorem 7.2.

Theorem 7.4. *If G and H are e.c. graphs, then $G \cong H$.*

Proof. We define a chain of partial isomorphisms f_n between induced subgraphs of G and H, so that the limit of the chain F is an isomorphism of G with H. The mappings f_n are defined inductively.

Let

$$V(G) = \{x_i : i \in \mathbb{N}\} \text{ and } V(H) = \{y_j : j \in \mathbb{N}\}.$$

Define the mapping f_0 to be the isomorphism from $\langle x_0 \rangle_G$ to $\langle y_0 \rangle_H$.

For an induction hypothesis, suppose that f_t is defined for some $t \geq 0$, and is an isomorphism. If t is even, then suppose that f_t has domain containing $\{x_0, \ldots, x_t\}$, and if t is odd, f_t has range containing $\{y_0, \ldots, y_t\}$.

We consider cases depending on whether t is even or odd. If t is even, then we "go forward". Let x be the vertex of least index not in the domain of f_t. By the induction hypothesis, the vertex $x \notin \{x_0, \ldots, x_t\}$ and so it could be x_{t+1}. We now define a mapping f_{t+1} whose domain is the domain of f_t and x (which ensures that x_{t+1} is in the domain of f_{t+1}). Suppose that the neighbors of x in the domain of f_t consist of the finite set S.

In H, by the e.c. property there is a vertex y joined to $f_t(S)$ and to no other vertex of the finite range of f_t. Define f_{t+1} to be the mapping extending f_t mapping x to y. It is straightforward to see that f_{t+1} preserves adjacencies and non-adjacencies by the choice of y. Further, it is a bijection with the desired range.

If t is odd, then we "go back" by a similar, and so omitted argument as going forward.

For all integers $t \geq 1$, we have that

(1) If t is even, then $\{x_0, \ldots, x_t\}$ is in the domain of f_t;

(2) If t is odd, then $\{y_0, \ldots, y_t\}$ is in the range of f_t;

(3) f_{t+1} extends f_t;

(4) f_t is an isomorphism.

Define the mapping $F : G \to H$ by

$$F = \bigcup_{t \in \mathbb{N}} f_t;$$

more explicitly, $F(x) = f_i(x_i)$. Then F is well defined by item (3), and is an isomorphism by (1), (2), and (4). □

What is this seemingly mysterious isomorphism type $\mathcal{R}$? How do we construct it deterministically? Define a graph R^* as follows. Let R_0 be a K_1. Assume that for a non-negative integer $t \geq 0$, the graph R_t is defined and finite. To form R_{t+1}, the idea is to add all possible "extensions" of R_t by one vertex. Hence, for each subset $S \subseteq V(R_t)$ (possibly empty) add a vertex z_S joined only to the vertices of S. The sets $\{V(R_t) : t \in \mathbb{N}\}$ and $\{E(R_t) : t \in \mathbb{N}\}$ are well-ordered sets or *chains*, so we can define

$$V(R^*) = \bigcup_{t \in \mathbb{N}} V(R_t), \qquad E(R^*) = \bigcup_{t \in \mathbb{N}} E(R_t).$$

We write $\lim_{t \to \infty} R_t = \mathcal{R}^*$, and say that $\mathcal{R}^*$ is the *limit of the chain* $(R_t : t \in \mathbb{N})$. Limits are powerful tools for proving results about infinite graphs. The graph $\mathcal{R}^*$ is e.c., and so is isomorphic to $\mathcal{R}$ by Theorems 7.3 and 7.4. To see this, choose R_t large enough to contain both A and B. A vertex correctly joined to A and B may be found in R_{t+1}.

Although $\mathcal{R}^*$ is an explicit construction of $\mathcal{R}$, the infinite random graph also arises naturally from the following arithmetic construction, given first by Rado [**180**] . Define a graph G with vertices $\mathbb{N}$. A vertex m is joined to n if 2^m occurs in the (unique) base 2 expansion of n, or if 2^n is in the base 2 expansion of m. Using an elementary argument (see Exercise 6), G is e.c., and so is isomorphic to $\mathcal{R}$.

Other representations of $\mathcal{R}$ exist, a few of which are outlined in the exercises (see Exercises 5 and 6). The graph $\mathcal{R}$ has a myriad of other interesting properties. For example, the following result follows by a variation of back-and-forth.

Theorem 7.5. *The graph $\mathcal{R}$ is* universal*: all countable graphs embed in $\mathcal{R}$.*

Proof. Let G be a fixed countable graph, and let

$$V(G) = \{x_i : i \in \mathbb{N}\}.$$

We name the subgraph induced by $\{x_i : 0 \le i \le t\}$ as G_t. Hence, $G = \lim_{t\to\infty} G_t$. We embed each G_t into $\mathcal{R}$ by induction, so each embedding extends the previous. To accomplish this, we go "forth" only. More precisely, let $f_0 : G_0 \to \mathcal{R}$ be any fixed embedding. Suppose that for an integer $t \ge 0$, there is an embedding $f_t : G_t \to \mathcal{R}$ extending f_0. The vertex x_{t+1} is joined to some set S of vertices in G_t. By the e.c. property, there is a vertex z of $\mathcal{R}$ joined only to $f_t(S)$ in G_t. Define the mapping $f_{t+1} : G_{t+1} \to R$ by extending f_t so that x_{t+1} is mapped onto z. The map f_{t+1} is an embedding by the choice of z. The mapping $F = \lim_{t\to\infty} f_t$ is an embedding of G into $\mathcal{R}$. □

A graph G is *homogeneous* if each partial isomorphism between finite induced subgraphs extends to an automorphism of G. For example, each clique is homogeneous, as is the 5-cycle (while not even the 6-cycle is homogeneous). We may view homogeneity as the strongest form of symmetry a graph can possess; for example, a homogeneous graph is both vertex- and arc-transitive.

The graph $\mathcal{R}$ is homogeneous by a back-and-forth argument (see Exercise 10a). We mention in passing that the homogeneous countable graphs have been completely classified. The classification of finite homogeneous graphs was completed independently by Sheehan [**189**], Gardiner [**92**], and Gol'fland and Klin [**99**].

Theorem 7.6. *A finite homogeneous graph is isomorphic to a disjoint union of complete graphs, a complete multipartite graph,* C_5, *or the graph* $K_3 \square K_3$.

For an integer $n \ge 3$, the *Henson graph* H_n is the unique isomorphism type of countably infinite K_n-free graph satisfying the following adjacency property: for each (K_{n-1})-free induced subgraph S, there is a vertex joined to S. By back-and-forth arguments, the graph H_n is universal for all K_n-free graphs, and is homogeneous. These graphs were first discovered and studied by Henson [**112**]. Lachlan and Woodrow [**132**] classified the countably infinite homogeneous graphs.

Theorem 7.7 ([**132**]). *The countably infinite homogeneous graphs are, up to isomorphism, the following.*

(1) *The graphs αK_β (α disjoint copies of complete graphs of order β), where α and β are cardinals with the property that $\alpha + \beta = \aleph_0$.*

(2) *The complements of graphs of* (1).

(3) *The Henson graphs H_n, $n \geq 3$.*

(4) *The complements of graphs of* (3).

(5) *The graph $\mathcal{R}$.*

7.3. Cop Density

The cop number of $\mathcal{R}$ is infinite; that is, $\mathcal{R}$ is *infinite-cop-win.* To see this, note that if the cops occupy a set B of vertices, with the robber at a vertex A, a vertex z in $\mathcal{R}$ correctly joined to A and B supplies an "escape route" for the robber. This sketch will be made more precise in Theorem 7.8, where we generalize this result to the uncountable class of so-called strongly 1-e.c. graphs.

Owing to our approach in this section, we make the following proviso:

All graphs in this section are countable.

When dealing with countable graphs, we exploit the fact that they are limits of chains of finite graphs. So we can often exploit properties of finite graphs (which tend to be better understood) to study countable graphs, supplying us a fair amount of control which may be lacking in higher cardinalities. To analyze the cop number of infinite graphs, we consider the *cop density* of a finite graph first introduced in [**28**]. In 2006, Geňa Hahn suggested the notion of cop density to the first author. Define

$$D_c(G) = \frac{c(G)}{|V(G)|}.$$

Note that $D_c(G)$ is a rational number in $[0, 1]$. The closer D_c is to one, the denser the cops are in the graph. We extend the definition of D_c to infinite graphs by considering limits of chains of finite graphs.

In this way, the cop density for infinite graphs is a real number in $[0, 1]$.

Every countable graph G is the limit of a chain of finite graphs, and there are infinitely many distinct chains with limit G. Suppose that $G = \lim_{n\to\infty} G_n$, where $\mathcal{C} = (G_n : n \in \mathbb{N})$ is a fixed chain of induced subgraphs of G. We say that $\mathcal{C}$ is a *full chain for* G. Define

$$D(G, \mathcal{C}) = \lim_{n\to\infty} D_c(G_n),$$

if the limit exists (and then it is a real number in $[0, 1]$). This is the *cop density of* G *relative to* $\mathcal{C}$; if $\mathcal{C}$ is clear from context, we refer to this as the *cop density of* G. We will only consider graphs and chains where this limit exists. Indeed, if the cop number of G is infinite, then for some chain the cop density equals 1 (see Theorem 7.8 (2) and Theorem 7.11). The *upper cop density of* G, written $UD(G)$, is defined as

$$\sup\{D(G, \mathcal{C}) : \mathcal{C} \text{ is a full chain for } G\}.$$

Note that $UD(G)$ does not depend on the chain, and is a parameter of G.

We illustrate these parameters with some examples. If G is a ray, then we may take $\mathcal{C}$ to be $(P_n : n \in \mathbb{N})$. As $c(P_n) = 1$, we have that $D(G, \mathcal{C}) = 0$. Let G be the disjoint union of infinitely many 4-cycles $\{C_4^{(i)} : i \in \mathbb{N}\}$, and let G_n be the disjoint union of the first n $C_4^{(i)}$. If $\mathcal{C} = (G_n : n \in \mathbb{N})$, then $D(G, \mathcal{C}) = \frac{1}{2}$. If G is an infinite clique, then $UD(G) = 0$, while $UD(H) = 1$ if H is an infinite co-clique.

If we insist that all the elements of the chain $\mathcal{C}$ are connected, then the situation for cop density changes radically. By Frankl's bound, $c(G) = o(|V(G)|)$ (see Theorem 1.6), it follows that

$$D(G, \mathcal{C}) = 0.$$

We consider the following weakening of the e.c. property. For a positive integer n, a graph G *is strongly* n*-e.c.* if for all disjoint finite sets of vertices A and B from G with $|A| \leq n$, there is a vertex z correctly joined to A and B. The following result, proved in [**28**], finds connections between infinite-cop-win graphs and the strongly 0- and 1-e.c. properties.

Theorem 7.8 ([**28**]). (1) *If G is strongly* 1*-e.c., then $c(G)$ is infinite.*

(2) *If $c(G)$ is infinite, then G satisfies the strongly* 0*-e.c. property. In particular, G is a spanning subgraph of $\mathcal{R}$.*

Proof. (1) Given only finitely many cops in G, we describe a winning strategy for the robber. By the strongly 0-e.c. property (which follows from the strongly 1-e.c. property), R may choose a vertex not joined to a vertex of C. Suppose after the cops' nth move, where $n \geq 0$ is fixed, the robber has not been captured. By the strongly 1-e.c. property, there is a vertex the robber can move to that is not joined to the vertices occupied by the cops.

(2) The robber has a winning strategy if there are only finitely many cops. Hence, no matter what finite set of vertices S the cops first choose to occupy, the robber can evade capture. It follows that there is a vertex $x \notin S$ that is not joined to any vertex of S. The second statement of (2) follows from the well-known fact that a strongly 0-e.c. graph is a spanning subgraph of $\mathcal{R}$ (see Exercise 9). □

We now prove the main theorem of this section that if G is strongly 1-e.c., then the cop density of G may be *any* real number in $[0, 1]$. This property applies, therefore, to a large number of graphs: for each $n \geq 0$, there are $2^{\aleph_0}$ many non-isomorphic countable graphs that are strongly n-e.c. but not strongly $(n+1)$-e.c.; see Theorem 4.1 of [**24**].

Theorem 7.9 ([**28**]). *If G is strongly* 1*-e.c., then for all $r \in [0, 1]$, there is a chain $\mathcal{C}$ in G such that $D(G, \mathcal{C}) = r$.*

Proof. Let $(p_n : n \in \mathbb{N})$ be a sequence of rationals in $[0, 1]$ such that $\lim_{n\to\infty} p_n = r$, with $p_0 = 1$. For example, if r has a decimal expansion $0.r_1r_2r_3\cdots$, then we can choose $p_i = 0.r_1r_2\cdots r_i$. We construct a chain $\mathcal{C} = (G_n : n \in \mathbb{N})$ in G such that $G = \lim_{n\to\infty} G_n$, and with the property that $D_c(G_n) = p_n$. Enumerate $V(G)$ as $\{x_n : n \in \mathbb{N}\}$.

We proceed inductively on n. For $n = 0$, let G_0 be the subgraph induced by x_0. Then

$$\frac{c(G_0)}{|V(G_0)|} = 1 = p_0.$$

Fix $n \geq 1$, suppose the induction hypothesis holds for all $k \leq n$, and let $p_{n+1} = \frac{a}{b}$, where a, b are positive integers. Further suppose for an inductive hypothesis that $\{x_0, \ldots, x_n\} \subseteq V(G_n)$. Without loss of generality, as $r \in [0, 1]$ we may assume $a < b$, and $\gcd(a, b) = 1$.

We add vertices to G_n in stages. Define G'_{n+1} to be the graph induced by $V(G_n) \cup \{x_{n+1}\}$. Suppose that $c(G'_{n+1}) = a'$ and also that $|V(G'_{n+1})| = b'$. If $\frac{a'}{b'} = \frac{a}{b}$, then let $G_{n+1} = G'_{n+1}$. Otherwise, we add some new vertices to adjust the density $D_c(G'_{n+1})$.

Each time an isolated vertex is added to a graph, the cop number increases by one. Adding an end-vertex to a graph does not change the cop number. We may assume that $\frac{a'}{b'} < \frac{a}{b}$ by adding an appropriate number of end-vertices. In this way, b' will become larger, while a' will remain unchanged.

We may add an arbitrary finite number of isolated vertices and end-vertices to G'_{n+1} by the strongly 1-e.c. property. We add x isolated vertices and y end-vertices to G'_{n+1} to form G_{n+1} so that

$$D_c(G_{n+1}) = \frac{c(G_{n+1})}{|V(G_{n+1})|} = \frac{a}{b}.$$

This is possible if we can solve the equation

$$\frac{a}{b} = \frac{a' + x}{b' + x + y},$$

which is equivalent to

$$(b - a)x - ay = ab' - a'b. \tag{7.1}$$

Note that $ab' - a'b > 0$, since otherwise, $ab' \leq a'b$ which is contrary to hypothesis. Hence, we obtain a linear Diophantine equation

$$cx + dy = e,$$

where $c = b - a > 0$, $d = -a < 0$, and $e = ab' - a'b > 0$. As $\gcd(b - a, -a) = \gcd(a, b) = 1$, (7.1) has infinitely many solutions. The general integer solution of (7.1) is

$$x = x_0 - at, y = y_0 - (b - a)t, \tag{7.2}$$

where (x_0, y_0) is a particular fixed solution, and t is an integer. (For example, we may take $(x_0, y_0) = (-a', a' - b')$.) As the coefficients of t in (7.2) are both negative, we may choose an appropriate $t < 0$ to ensure an integer solution of (7.1) (x, y) with $x, y \geq 0$. This completes the induction step in constructing G_{n+1}.

As $\{x_0, \ldots, x_n\} \subseteq V(G_n)$ for all $n \in \mathbb{N}$, we have that $\mathcal{C} = (G_n : n \in \mathbb{N})$ is a full chain for G. Further,

$$D(G, \mathcal{C}) = \lim_{n \to \infty} p_n = r. \qquad \square$$

As the infinite random graph $\mathcal{R}$ is e.c. (and so is strongly 1-e.c.) we have the following corollary.

Corollary 7.10 ([**28**]). *For all $r \in [0, 1]$, there is a chain $\mathcal{C}$ in $\mathcal{R}$ such that $D(\mathcal{R}, \mathcal{C}) = r$.*

Our next result completely characterizes the upper cop density of a graph G: $UD(G)$ takes on one of the two values 0 or 1, and equals 1 exactly when G is strongly 0-e.c. This fact, proven in [**28**], is somewhat unexpected. Even more unexpected is the connection with cop density and the infinite random graph.

Theorem 7.11 ([**28**]). *The following are equivalent.*

(1) $UD(G) = 1$.

(2) $UD(G) > 0$.

(3) *G is strongly* 0*-e.c.*

(4) *G is a spanning subgraph of $\mathcal{R}$.*

Proof. Since (1) implies (2) is immediate, and (3) being equivalent to (4) is Exercise 9, we prove that (2) implies (3) and (3) implies (1).

For (2) implies (3), suppose for the contrapositive that G is not strongly 0-e.c. Then there is some finite set S of vertices of G with the property that each vertex not in S is joined to some vertex of S; in other words, S is a finite dominating set for G. Let

$$\mathcal{C} = (G_n : n \in \mathbb{N})$$

be a fixed full chain of finite graphs in G, and suppose that n_0 is the least integer n where $S \subseteq V(G_n)$. Fix $t \geq n_0$. Then $c(G_t) \leq |S|$, since S is a dominating set for G and hence, G_t. Thus,

$$D_c(G) \leq \frac{|S|}{|V(G_t)|} = o(1).$$

Hence, $UD(G) = 0$.

For (3) implies (1), enumerate $V(G)$ as $\{x_i : i \in \mathbb{N}\}$. Fix a countable sequence of real numbers $\epsilon_n \in (0,1)$, such that $\lim_{n\to\infty} \epsilon_n = 1$ and $\epsilon_0 = 1$. It is sufficient to inductively construct a full chain

$$\mathcal{C} =(G_n : n \in \mathbb{N})$$

of finite induced subgraphs in G satisfying the following conditions for all $n \in \mathbb{N}$:

(a) $x_n \in V(G_n)$;

(b) $\frac{c(G_n)}{|V(G_n)|} \geq \epsilon_n$.

If items (a) and (b) hold, then

$$UD(G) \geq \lim_{n\to\infty} D_c(G_n) \geq \lim_{n\to\infty} \epsilon_n = 1,$$

and so $UD(G) = 1$.

Let G_0 be the subgraph induced by $\{x_0\}$. Then G_0 satisfies items (a) and (b) above. Suppose G_n has been constructed. Add x_{n+1} to G_n (if it is not already there) to form the induced subgraph G'_{n+1}. If $\frac{c(G'_{n+1})}{|V(G'_{n+1})|} \geq \epsilon_{n+1}$, then let $G_{n+1} = G'_{n+1}$. Otherwise, suppose that

$$\frac{c(G'_{n+1})}{|V(G'_{n+1})|} = \frac{p}{q} < \epsilon_{n+1}.$$

By the strongly 0-e.c. property of G, we may add k isolated vertices to G'_{n+1} to form G_{n+1}, where k is a positive integer that is to be determined. Then

$$\frac{c(G_{n+1})}{|V(G_{n+1})|} = \frac{p+k}{q+k}.$$

We choose k so that $\frac{p+k}{q+k} \geq \epsilon_{n+1}$, which holds so long as

$$k \geq \frac{\epsilon_{n+1}q - p}{1 - \epsilon_{n+1}}.$$ □

The following corollary gives a necessary condition for G to have infinite cop number, and follows directly by Theorems 7.8 and 7.11.

Corollary 7.12 ([**28**]). *If $c(G)$ is infinite, then $UD(G) = 1$.*

The converse of Corollary 7.12, however, is false in a strong sense. For each real number $r \in [0,1]$, there is a graph $G(r)$ with $c(G(r)) = 1$, so that for some full chain $\mathcal{C}$ in $G(r)$, $D(G(r), \mathcal{C}) = r$. See Exercise 12.

7.4. Infinite Chordal Graphs

As another instance in which results from finite graphs do not translate to infinite ones, we consider chordal graphs. Let C_n be a cycle in G, with $n \geq 4$. A *chord* of C_n is an edge in G between two vertices in C_n which are not adjacent in C_n. Roughly, a chord acts as a short-cut across the cycle. See Figure 7.4. The graph G is *chordal* if each cycle of length at least 4 has a chord. See Figure 7.5. Chordal graphs are sometimes called *triangulated graphs*, for obvious reasons. A vertex of G is *simplicial* if its neighborhood induces a complete graph. Every finite chordal graph contains at least two simplicial vertices, and the deletion of a simplicial vertex leaves a chordal graph; see Exercise 4 in Chapter 1. As a simplicial vertex is a corner, we derive that a finite chordal graph is dismantlable and thus, is cop-win.

However, an infinite tree containing a ray is chordal but not cop-win. Such trees have infinite diameter. Inspired by a question of Martin Farber which asked if infinite chordal graphs (more generally, bridged graphs) of finite diameter are cop-win, it was shown in [**104**] that there exist infinite chordal graphs of diameter 2 that are not cop-win. How paradoxical! The difficulty lies in finding examples with finite diameter.

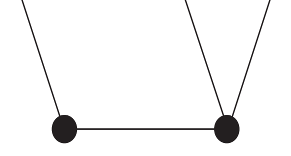


Figure 7.4. A cycle with a chord.

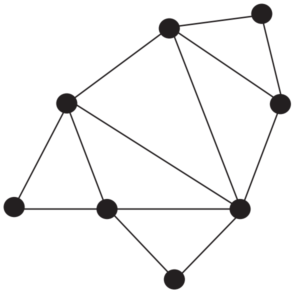


Figure 7.5. A chordal graph.

Theorem 7.13 ([**104**]). *For each infinite cardinal κ, there exist chordal, robber-win graphs of order κ with diameter* 2.

To prove Theorem 7.13, we consider a graph built from words or strings over a binary alphabet. Define our alphabet to be $B = \{1, 2\}$ (any two element set would do), and let B_n be the set of *words of length n* with letters in B; that is, elements of B_n are strings

$$b_1 b_2 \cdots b_n,$$

where the b_i are either 1 or 2. Let B^* be the set of all words over B, with empty word ε. For words u, v, we write uv for the concatenation of u and v. We define an order on B^* by $u \leq v$ if u is an initial segment of v. Hence,

$$1121 \leq 1121112 \text{ and } 22 \leq 2212221.$$

This is just the standard lexicographic order over words. Observe that 1 and 2 are incomparable. Note that $(B^*, \leq)$ forms an ordered tree with root ε. See Figure 7.6.

We make this ordered tree into a graph G^* in the following way. To form the vertices of G^*, for each word u, replace each vertex of B^* by a copy of $B^* \times B^*$ called B_u. We identify vertices of G^* as triples $x = (u, v, w)$, where u, v, and w are words over B. Define $\pi_1(x) = v$ and $\pi_2(x) = w$. Transform each B_u into a complete graph, and add an edge between each vertex of B_u and each vertex of B_{uk}, where $k \in B$. To finish the description of the edge set of G^*, add the following set

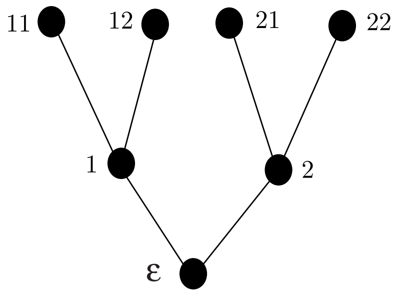


Figure 7.6. The first three rows of the rooted tree B^*.

of edges:

$$X = \{xy : x \in B_u, y \in B_v, v = ukz, \text{ for some } k \in B,$$
$$z \in B^* \backslash \{\varepsilon\}, \text{ and } z \leq \pi_k(x)\}.$$

We use the following lemma whose proof is left as an exercise (see Exercise 15). From Chapter 4, recall the lexicographic product of graphs G and H, written $G \bullet H$. One way to think of $G \bullet H$ is by replacing each vertex of G by a copy of H. Edges in each copy of H remain unchanged, and if vertices of G are joined, then there are all edges between corresponding copies of H. See Figure 7.7.

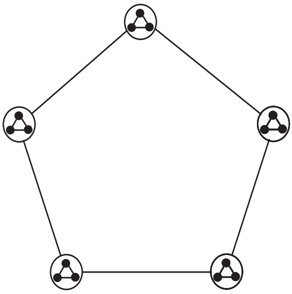


Figure 7.7. The graph $C_5 \bullet K_3$. There are all edges present between neighboring K_3's on the 5-cycle.

Lemma 7.14 ([**104**]). *If G is chordal, robber-win, and diameter 2, then so is $G \bullet K$, where K is a clique.*

We now prove the main theorem of this section.

Proof of Theorem 7.13. We first consider the case when κ is countably infinite. We show that G^* has all the desired properties. We first show that G^* is of diameter 2. Fix non-joined vertices $x \in B_u$ and $y \in B_v$. In particular, $u \neq v$. Let w be the longest common initial segment of u and v. If $w = u$ (the case when $w = v$ is similar), then $v = ukz$, for $k \in B$ and $z \in B^*$. Since x and y are not joined, we have that z is not empty. But then x and y are joined to (u, z, z). Otherwise, $w \neq u, v$ and without loss of generality,

$$u = w1i, v = w2j$$

for words w, i and j. If i and j are both empty, then u and v have as a common neighbor any vertex in B_w. Otherwise, they are both joined to (w, i, j).

We next show that G^* is robber-win. Otherwise, one cop wins; without loss of generality (see Exercise 16) we can assume that the cop's initial position is in B_ϵ. Suppose that the robber is in B_u which is distance 2 from C. We prove that the robber can maintain distance 2 from the cop after each of his moves (and thus, has a winning strategy). It is enough to show that if $z \in B_v$ is joined to R, then there is a neighbor w of R of distance 2 to z. This is straightforward if $u = v$, so we consider the cases that $u \neq v$ and either $u \leq v$ or $v \leq u$. If $u \leq v$, then $v = uks$, where $k \in B$ and s is some word. But then any vertex w in $B_{u(3-k)}$ is joined to R but not z. Now if $v \leq u$ and $u = vks$. Note that $st \not\leq \pi_k(z)$ for some $t \in B$. Hence, any vertex w of B_{ut} has the desired property.

We finish the proof by showing that G^* is chordal. We define the following relation $(u, v, w) \sim (u', v', w')$ if u and u' are comparable; that is, $u \leq u'$ or $u' \leq u$. Note that edges in G^* only occur between comparable copies of B_u and $B_{u'}$. Suppose that

$$C_n = x_0 x_1 \cdots x_{n-1}$$

is a cycle in G^* with $x_i = (u_i, v_i, w_i)$, and $n \geq 4$. Without loss of generality, assume that u_0 is the minimum of the x_i with respect to $\leq$. If $u_i = u_0$ for all i, then C_n is within the clique B_{u_0}, and so C_n has a chord. Hence, assume that there is a vertex x_i with $u_i > u_0$. We say that an edge x_ix_{i+1} (working (mod n)) of C_n is *up* if $u_i \leq u_{i+1}$ and *down* if $u_{i+1} \leq u_i$. It is possible for an edge to be both up and down if $u_i = u_{i+1}$.

The cycle C_n must contain an up edge x_ix_{i+1} followed by a down edge $x_{i+1}x_{i+2}$. We show that x_ix_{i+2} is a chord, which finishes the proof that G^* is chordal. We could have that $u_i = u_{i+1}$ or $u_{i+1} = u_{i+2}$, but in either case x_ix_{i+2} is an edge. Hence, there are $k, k' \in B$ and $s, s' \in B^*$ such that

$$u_iks = u_{i+1} = u_{i+2}k's'.$$

It follows that $x_i \sim x_{i+2}$. Suppose that $u_i \leq u_{i+2}$ (the other case $u_{i+2} \leq u_i$ is similar and so omitted). If $u_{i+2} = u_i$ or $u_{i+2} = u_{i+1}k$, then x_ix_{i+2} is an edge of G^*. The only remaining case is that $u_{i+2} = u_ikt$, where $t \in B^*$ is a non-empty word, and $s = tt'$. But then $x_ix_{i+1} \in X$, and $tt' \leq \pi_k(x_i)$. Hence, $t \leq \pi_k(x_i)$ and x_ix_{i+2} is an edge (from those in X).

We now generalize our examples to any infinite cardinality κ. Define $G_\kappa \cong G^* \bullet K_\kappa$. Then

$$|V(G_\kappa)| = \kappa|V(G^*)| = \kappa,$$

as $|V(G^*)| \leq \kappa$. By Lemma 7.14, the graph G_κ has diameter 2, is chordal, and is robber-win. □

7.5. Vertex-transitive Cop-win Graphs

As we have seen from Theorem 7.13, the cop number of infinite graphs behaves rather differently than in the finite case. Vertex-transitive cop-win finite graphs are cliques. (See Exercise 3 in Chapter 1.) However, this fails badly in the infinite case, which is the focus of this section.

A class of graphs is *large* if for each infinite cardinal κ there are 2^κ many non-isomorphic graphs of order κ in the class. In other words, a large class contains as many as possible non-isomorphic graphs of each infinite cardinality. For example, the classes of all graphs, all trees, and all k-chromatic graphs for k finite are large. Recall that a graph G is *vertex-transitive* if for each pair of vertices x and y there is an automorphism of G mapping x to y. For example, each clique is vertex-transitive, as is a hypercube. We describe the result of [**27**] showing that for any integer $k > 0$ there are large families of k-cop-win graphs that are vertex-transitive. This result reinforces the divide between the theories of cop number of finite and infinite graphs.

Theorem 7.15 ([**27**]). *The class of cop-win, vertex-transitive graphs with the property that the cop can win in two moves is large.*

Before we prove Theorem 7.15, we recall some properties of the strong product of a set of graphs over a possibly infinite index set. Let I be an index set. The *strong product* of a set $\{G_i : i \in I\}$ of graphs is the graph $\boxtimes_{i\in I} G_i$ defined by

$$\begin{aligned} V(\boxtimes_{i\in I} G_i) &= \{f : I \to \bigcup_{i\in I} V(G_i) : f(i) \in V(G_i) \text{ for all } i \in I\}, \\ E(\boxtimes_{i\in I} G_i) &= \{fg :\ f \neq g \text{ and for all } i \in I, \\ f(i) &= g(i) \text{ or } f(i)g(i) \in E(G_i)\}. \end{aligned}$$

For background on strong products, we refer the reader to [**119**]. Strong products exhibit unusual properties if there are infinitely many factors. An elementary but instructive example is the following. The graph

$$G = \boxtimes_{i\geq 1} P_i$$

is not connected even though each factor is. To see this, label the vertex set of each P_i by $[i]$. Then the vertex f of G with $f(i) = 1$ for all $i \geq 1$, is in a different connected component than the vertex g, where $g(i) = i$ for all $i \geq 1$.

An issue with the last example was that it contained vertices which differed in infinitely many coordinates. We now consider connected components of strong products. In particular, we allow vertices to differ in only finitely many coordinates. Fix a vertex $f \in V(\boxtimes_{i\in I} G_i)$ and define the *weak strong product* of $\{G_i : i \in I\}$ *with base* f as the subgraph $\boxtimes_f^I G_i$ of $\boxtimes_{i\in I} G_i$ induced by the set of all $g \in V(\boxtimes_{i\in I} G_i)$ such that $\{i \in I : g(i) \neq f(i)\}$ is finite. The graph $\boxtimes_f^I G_i$ is connected if each factor is, and if $|I| \leq \kappa$ and $|V(G_i)| \leq \kappa$ for each $i \in I$, then $|V(\boxtimes_f^I G_i)| \leq \kappa$ (see Exercise 19b). For $i \in I$, the *projection mapping* $\pi_i : \boxtimes_f^I G_i \to G_i$ is defined by $\pi_i(g) = g(i)$.

When all the factors are isomorphic to some fixed graph G, we refer to a *power* of G. Let $\{G_i : i \in I\}$ be a set of isomorphic copies of G. Denote by $\boxtimes^I G$ the strong product $\boxtimes_{i \in I} G_i$. If $f \in V(\boxtimes^I G)$ is fixed, denote by G^I_f the weak strong product $\boxtimes^I_f G$ with base f. One particular power of a graph is of special interest as it allows us to construct vertex-transitive graphs out of non-transitive ones. Let κ be a cardinal, and let H be a graph of order κ. Let $I = \kappa \times V(H)$, and define $f : I \to V(H)$ by $f(\beta, v) = v$. The power H^I_f of H with base f will be called the *canonical power* of H and will be denoted by H^H.

Since automorphisms of the factors applied coordinate-wise yield an automorphism of the product, it follows that the weak strong product of vertex-transitive graphs is vertex-transitive (see Exercise 4). However, the following technical lemma from [**27**] demonstrates the paradoxical property that if there are infinitely many factors, the weak strong product may be vertex-transitive even if none of the factors are!

Lemma 7.16 ([**27**]). *If H is an infinite graph, then the canonical power of H^H is vertex-transitive.*

Proof. We prove that for all $g \in V(H^H)$ there is an automorphism ψ_g of H^H which maps f to g. The lemma follows from this claim.

Fix $g \in V(H^H)$. Without loss of generality, we may assume that $g \neq f$ as in that case ψ is the identity map. The families $\{g^{-1}(v)\}_{v \in V(H)}$ and $\{f^{-1}(v)\}_{v \in V(H)}$ partition I and

$$|g^{-1}(v)| = |f^{-1}(v)| = \kappa,$$

since g and f differ in at most finitely many values. It follows that for each $v \in V(H)$ there exists a bijection $\phi_v : g^{-1}(v) \longrightarrow f^{-1}(v)$. Using the maps ϕ_v, we define a map $\phi : I \longrightarrow I$ by

$$\phi(\beta, v) = \phi_{g(\beta,v)}(\beta, v).$$

Then ϕ is a bijection since for each (β, v), $\phi_{g(\beta,v)}$ is a bijection, and the sets $g^{-1}(v)$ for $v \in V(H)$ partition I. Observe that $g(\beta, v) = f(\phi(\beta, v))$.

We now define ψ_g as follows. For a vertex $h \in V(H^H)$, set $\psi_g(h) = \hat{h}$ with $\hat{h}(\beta, v) = h(\phi(\beta, v))$. We claim first that ψ_g is a bijection. To see this, note that if $h \neq h'$, then $h(\beta, v) \neq h'(\beta, v)$ for some $(\beta, v) \in I$. Hence,

$$\hat{h}(\phi^{-1}(\beta, v)) = h(\beta, v) \neq h'(\beta, v) = \hat{h'}(\phi^{-1}(\beta, v)).$$

Also note that for any h, if h' is defined by $h'(\beta, v) = h(\phi^{-1}(\beta, v))$ then $\psi_g(h') = h$.

As $\psi_g(f) = g$, the claim of the lemma follows once we verify that ψ_g preserves adjacencies and non-adjacencies. Let $h, h' \in V(H^H)$. We have that $hh' \in E(H^H)$ if and only for any $(\beta, v) \in I$, either $h(\beta, v) = h'(\beta, v)$, or the two images of (β, v) are adjacent in H. Since ϕ is a permutation of I, and by definition of ψ, this happens if and only if $h(\phi(\beta, v))$ and $h'(\phi(\beta, v))$ are either identical or adjacent for every $(\beta, v) \in I$. In particular, $hh' \in E(H^H)$ if and only if $\hat{h}\hat{h'} \in E(H^H)$. □

With Lemma 7.16 in hand, we now turn to the proof of the main theorem of this section.

Proof of Theorem 7.15. For each infinite cardinal κ there are 2^κ many non-isomorphic trees of order κ (see Exercise 20a). Fix a tree T of order κ, and let $\hat{T}$ be the graph formed by adding a universal vertex u to T. The canonical power $\hat{T}^{\hat{T}}$ of $\hat{T}$ has order κ and is vertex-transitive by Lemma 7.16.

A single cop may win on $\hat{T}^{\hat{T}}$ as follows. The cop C will initially occupy the base vertex $c_0 = f$ and the robber R some vertex r_0. For all but finitely many indices $(\beta, v) \in I$, we have that $r_0(\beta, v) = c_0(\beta, v)$. We define the next position c_1 of the cop coordinate-wise: if $c_0(\beta, v)$ is joined or equal to $r_0(\beta, v)$, let $c_1(\beta, v) = c_0(\beta, v)$, otherwise, let $c_1(\beta, v) = u$. When the robber moves from r_0 to r_1, we have $c_1(\beta, v)$ joined or equal to $r_1(\beta, v)$ for every $(\beta, v) \in I$, so the cop captures the robber by moving to $c_2 = r_1$. Note that we have shown that cop captures the robber in at most two moves.

All that is left to show is that if T and T' are not isomorphic, then $\hat{T}^{\hat{T}}$ and $\hat{T'}^{\hat{T'}}$ are not isomorphic. We examine maximal cliques in the canonical powers; the trick is to build an auxiliary graph from cliques. Note first that the maximal cliques in $\hat{T}^{\hat{T}}$ are simply products of triangles in every factor (the clique number of each factor is three, after all). Formally, if K is a maximal clique in $\hat{T}^{\hat{T}}$, then for every $(\beta, v) \in I$, there exist two adjacent vertices $s_{(\beta,v)}, t_{(\beta,v)}$ of T such that

$$K = \{g \in V(\hat{T}^{\hat{T}}) : g(\beta, v) \in \{u, s_{(\beta,v)}, t_{(\beta,v)}\} \text{ for all } (\beta, v) \in I\}.$$

Therefore, if $P = K \cap K'$ is a maximal proper intersection of maximal cliques in $\hat{T}^{\hat{T}}$, there exists an index $(\beta_0, v_0) \in I$ and a vertex s_0 of T such that

$$P = \left\{ g \in V(\hat{T}^{\hat{T}}) : g(\beta, v) \in \left\{ \begin{array}{l} \{u, s_0\} \text{ if } (\beta, v) = (\beta_0, v_0) \\ \{u, s_{(\beta,v)}, t_{(\beta,v)}\} \text{ otherwise} \end{array} \right\} \right\}.$$

Let $\mathcal{V}(T)$ be the set of all maximal proper intersections of maximal cliques in $\hat{T}^{\hat{T}}$. We construct an auxiliary graph $H(T)$ with vertex set $\mathcal{V}(T)$ by putting an edge between two cliques $P, Q \in \mathcal{V}(T)$ if $P \cup Q$ is a maximal clique of $\hat{T}^{\hat{T}}$. By the above characterization, $H(T)$ consists of disjoint copies of T, one for each index (β_0, v_0) in I. Thus, if $T \ncong T'$, then $H(T) \ncong H(T')$. □

Theorem 7.15 generalizes to the case for $k \geq 2$ with a little more help from graph products. As the reader is no doubt aware, unique factorization into primes is a fundamental property of the integers (it is indeed often referred to as the *Fundamental Theorem of Arithmetic*). There is a well-developed literature on analogous unique factorization-type theorems for graph products (where graphs play the role of integers, and ordinary products of integers are graph products). For some isomorphic products, the factors are isomorphic up to reordering. In particular, we employ the following unique factorization theorem for Cartesian products; see Theorem B.9 of [**119**].

Theorem 7.17 ([**119**]). *Suppose that A, B, and C are graphs,with A finite and connected. If*

$$A \boxtimes B \cong A \boxtimes C,$$

then $B \cong C$.

Corollary 7.18 ([**27**]). *For $k > 1$ a positive integer, the class of vertex-transitive k-cop-win graphs is large.*

Proof. Let $H(k)$ be the Cartesian product of $k - 1$ cycles of length 4. Then $H(k)$ is a vertex-transitive graph, as C_4 is. By Theorem 4.3, we have that $c(H(k)) = k$. Now let

$$J(T) = H(k) \boxtimes \hat{T}^{\hat{T}},$$

where $\hat{T}^{\hat{T}}$ are the cop-win graphs from the proof of Theorem 7.15. Then the infinite graph $J(T)$ has order κ. As both factors $H(k)$ and are $\hat{T}^{\hat{T}}$ vertex-transitive, so is $J(T)$. By Theorem 2.8 (3) we have that $c(J(T)) = k$. By Theorem 7.17 if $\hat{T}^{\hat{T}} \not\cong \hat{T'}^{\hat{T'}}$, then $J(T) \not\cong J(T')$. The proof now follows by Theorem 7.15. □

What about classes with bounded chromatic number? The large classes described in Theorem 7.15 and Corollary 7.18 have infinite clique number and hence, infinite chromatic number. An open problem is to find large classes of cop-win graphs whose members are k-chromatic, where $k \geq 2$ is an integer.

Exercises

As a reminder, all graphs throughout are connected.

1. Give examples of countably infinite graphs whose cop number is any fixed finite $k > 0$.
2. Give an example of an infinite graph with girth at least 5 whose cop number is strictly smaller than its minimum degree.
3. For each infinite cardinal κ, give an example of a graph whose cop number is κ.
4. Prove that if each graph in $\{G_i : i \in I\}$ is vertex-transitive, then so is any weak strong product $\boxtimes_f^I G_i$.
5. Let the vertices of G be the set of primes $\mathbb{P}_1$ congruent to 1 (mod 4). The set $\mathbb{P}_1$ is infinite by Dirichlet's Theorem on primes in arithmetic progressions. Two distinct primes p and q in $\mathbb{P}_1$ are joined if p is a square (mod q) or q is a square (mod p). Show that $G \cong \mathcal{R}$. (*Hint*: Use the Chinese Remainder Theorem and Dirichlet's theorem on the arithmetic progression of primes.)

6. Define a graph G with vertices $\mathbb{N}$. A vertex m is joined to n if 2^m occurs in the (unique) base 2 expansion of n, or if 2^n is in the base 2 expansion of m. Show that $G \cong \mathcal{R}$.

7. A graph G satisfies the *pigeonhole property* $(\mathcal{P})$ if whenever the vertices of G are colored red and blue (with each vertex receiving exactly one color), then the subgraph induced by some one color is isomorphic to G.
   (a) Prove that $\mathcal{R}$ satisfies $(\mathcal{P})$.
   (b) [**40**] Prove that the countable graphs satisfying $(\mathcal{P})$ are K_1, $K_{\aleph_0}$, $\overline{K_{\aleph_0}}$, and $\mathcal{R}$. Hence, $\mathcal{R}$ is the unique isotype of 1-e.c. graph with $(\mathcal{P})$.

8. (a) Show that a strongly 1-e.c. graph has no vertex of finite degree.
   (b) Show that a strongly 2-e.c. graph is of diameter 2.

9. By adapting a back-and-forth argument, show that a graph is strongly 0-e.c. if and only if G is a spanning subgraph of $\mathcal{R}$.

10. (a) Prove that $\mathcal{R}$ is homogeneous.
    (b) Prove that $\mathcal{R}$ is the unique isomorphism type of homogeneous and universal graph.
    (c) Determine the cop number for all countable homogeneous graphs.

11. (a) Prove that the Henson graphs H_n are homogeneous and universal for all $n \geq 3$.
    (b) Find the cop number and upper density of the graphs H_n.
    (c) Determine the cop number for all countable homogeneous graphs listed in Theorem 7.7.
    (d) Find the possible cop densities of each countable homogeneous graph.

12. [**28**] Show that the converse of Theorem 7.12 is false by showing that for each real number $r \in [0, 1]$, there is a graph $G(r)$ with $c(G(r)) = 1$, so that for some full chain $\mathcal{C}$ in $G(r)$, $D(G(r), \mathcal{C}) = r$.

13. Find an infinite family of infinite-cop-win graphs that are not strongly 1-e.c.

14. Find an infinite family of strongly 0-e.c. graphs that are cop-win.

15. Prove Lemma 7.14: if G is chordal, robber-win, and diameter 2, then so is the lexicographic product $G \bullet K$, where K is a clique.

16. [**104**] Using notation from Section 7.4, for each $x \in B^*$, define $f_x : V(G^*) \to V(G^*)$ by
$$f_x((u, v, w)) = (xu, v, w).$$
Prove that f_x is an isomorphism of G^* with the subgraph induced by $\bigcup_{x \leq u} B_u$.

17. Show that if G contains an isometric ray, then G is robber-win.

18. Characterize the cop-win graphs which do not contain a triangle. (*Hint*: Show they are trees.)

19. (a) Show that the graph $\boxtimes_f^I G_i$ is connected if each factor is.
    (b) If $|I| \leq \kappa$ and $|V(G_i)| \leq \kappa$ for each $i \in I$, then prove that
$$|V(\boxtimes_f^I G_i)| \leq \kappa.$$

20. (a) Prove that for each infinite cardinal κ there are 2^κ-many non-isomorphic trees of order κ.
    (b) Repeat part (a), but for rayless (and hence, cop-win) trees.

21. [**44**] A graph G is *constructible* if there is a well-order $\leq$ on $V(G)$ such that every vertex x which is not the smallest element of $(V(G), \leq)$ is dominated by some vertex $y \neq x$ in the subgraph induced by $\{z \in V(G) : z \leq x\}$.
    (a) Show that if G is finite, then a graph is dismantlable if and only if it is constructible.
    (b) Show that the *double ray* (that is, the graph with vertices $\mathbb{Z}$ and edges $i(i+1)$, where $i \in \mathbb{Z}$) is constructible, but not dismantlable.
    (c) Show that the ray is constructible but not dismantlable.

22. [**44**] A graph is *weakly-cop-win* if the cop wins either if he really catches the robber or if he forces him to run straight ahead, that is, move endlessly by visiting each vertex at most once, except possibly finitely many of them at the beginning of the game.
    (a) Prove that a finite weakly-cop-win graph is cop-win.
    (b) Explain why a tree (of any cardinality) is weakly-cop-win. In particular, not all weakly-cop-win graphs are cop-win.
    (c) Show that a chordal graph is weakly-cop-win.

Chapter 8

Variants of Cops and Robbers

Having surveyed the major results on the classical Cops and Robbers game, we now investigate what happens when the rules or mode of game play are modified. Cops and Robbers with the usual rules is by far the best understood of the variants that have been studied. The loss of some or all loops has been dealt with in [**34**]; see also Exercises 13 and 14. Even then, as we have seen, many problems (such as Meyniel's conjecture) surround the cop number, and graphs with cop number greater than 1 are not well understood. Nevertheless, Cops and Robbers is less realistic than a game where say the robber is partially invisible, or when the cops can catch the robber at some distance away (say using a gun or taser). In this chapter, we consider what happens when we relax or strengthen the rules of Cops and Robbers. One of the main assumptions in the original game is that both the cops and robber can see each other at all times. A natural variant is therefore, when there is imperfect information about the position of the robber. In Section 8.1 below, we suppose the robber is partially invisible, but he in turn can see the positions of the cops. The cops may then employ a variety of tools which give some partial information about the robber's position or movements. For example, the cop could lay traps which impede or capture the robber; see Section 8.2. Other possibilities would be photo-radar or alarms which sound when the robber moves on a vertex or edge. For another twist, in Section 8.3 we play with pairs of cops who must stay sufficiently close to each other to make a move (in particular, they must be adjacent to each other in all rounds). We introduce the tandem-win graphs, where such a movement-restricted pair of cops have a winning strategy. We consider the situation where the cops and robber play on different sets of edges in Section 8.4.

Another class of variants allows the cops greater power. In distance k Cops and Robbers, the cops can shoot the robber at some prescribed distance k away; see Section 8.5. As a final variant, we measure not the number of cops, but the actual time it takes for cops to capture the robber. Imagine that the cops have an important schedule (such as a tip-off to an imminent theft or attack) and must catch the robber in some prescribed time. This gives rise to the notion of capture time, which measures the minimum number of rounds needed for the cops to win, assuming optimal play (that is, the robber is avoiding capture as long as is possible); see Section 8.6.

As Theorem 2.3 demonstrated, the structure of cop-win graphs is well understood. Most of the variants have one aspect in common: characterizing graphs where one cop can win in the variant is decidedly harder, if indeed it has been accomplished, than in the original game.

8.1. Imperfect Information

In real life it would be unusual for the cops to have perfect information about the robber's whereabouts at all times. On television shows, in movies and in real life, technology in the way of door alarms, motion detectors and video cameras give partial information about the robber's whereabouts. But when the robber moves into the "blind" spots of whatever security system is in place, the cops have to infer the robber's possible movements. A successful game, Scotland Yard©, is based on the premise that cops can only see the robber's position every several moves, while in the meantime he can skulk in the shadows unseen. Clarke and several coauthors [**54, 58, 59, 161**] have looked at the use of "technology" and how it affects the cop number when the underlying graph is cop-win. (See also [**50, 53**]). As one can imagine, a door alarm going off only tells the cops where the robber is. A camera at an intersection, backed up by face recognition software, would not only give the position but also the direction the robber went. The underlying structure in all of the proofs, so far, has been the use of cop-win spanning trees. We recall a definition from Chapter 2, Exercise 28 (also [**51**]). Let G be a cop-win graph with vertices $[n]$, and consider a sequence S of retractions $i \to j$, for $i \in [n]$ and some $j < i$. We need to change this to a directed spanning tree. Define the *directed cop-win spanning tree (relative to S)*, written T, so that $V(T) = V(G)$ and there is a directed edge from i to j if and only if $i \to j$ in S. Since the graph is labeled, the number of cop-win orderings can be large. There are $n!$ possible ordering of the vertices in a given cop-win ordering; when a vertex, say v, is chosen to be retracted there could be as many as $\deg(v)$ choices for the target. A (not too useful) upper bound to the number of directed cop-win spanning trees is then

$$n! \left(\prod_{v \in V(G)} \deg(v) \right).$$

Unless G is a complete graph, not all $n!$ vertex sequences will be cop-win orderings, and when a vertex is retracted, not all of its neighbors will be available as the target of the 1-point retraction.

Suppose the cop starts at t in Figure 8.1 and that she has no information about the robber's position. If he moves to vertex a, then he still has to move to b to ensure that the robber is not there. When he moves back to t he knows that the robber is not in the ab arm; but when he moves down the cd arm and reaches d, the robber

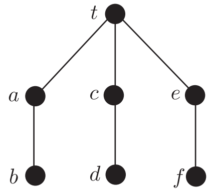


Figure 8.1. The cop has no information on the position of the robber.

could have moved to t. By the time the cop reaches t he no longer knows that the robber is not in the ab arm. Technology can give information that helps the cop. An *alarm* indicates when a vertex has been entered. Even if there were an alarm on t, it would not give the cop any information as to which arm currently hosts the robber. However, if the alarm were on vertex a, the situation is different. The cop checks out the ab arm then moves to c, d, c, and then t. If the alarm had sounded and the robber did not have enough time to move past t, then he would be caught on t; therefore, if he is still at large he must be on the ab arm. If the alarm did not sound, then the robber must be on the ef arm.

In Figure 8.2, one alarm at a no longer helps the cop. If the alarm sounds when the cop is at y, by the time the cop reaches t the robber could be on either of the other two arms. Two alarms, one on a and one on e, would suffice although a better placement would be on a and b. This would give information about the direction the robber is traveling.

Apart from alarms which can be on edges or on vertices, other devices that are used include closed-circuit television or *CCTv* placed on a vertex, which indicates not only that the robber has entered a vertex but also when he leaves and in which direction; *photo-radar,* which is a camera on an edge that indicates when the robber traverses the edge and in which direction; and a *road-block*, sometimes called a *trap*, which could be on an edge or at a vertex—the robber is not

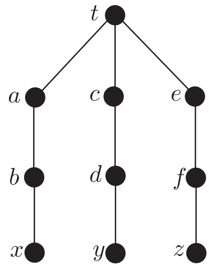


Figure 8.2. Two alarms at a and b.

allowed to pass through a road-block but the cops can. Other possibilities can be imagined: for example, certain edges are doors that are locked and the cop has the key but the robber has to spend one move "picking" the lock.

In both Figures 8.1 and 8.2, a photo-radar unit on ab acts the same as the two alarms on a and b. A road-block at t would trap the robber in one arm. A CCTv at t would also suffice. The cop checks each arm in turn and knows immediately if the robber switches arms, at this point the cop abandons the systematic search and goes down the arm the robber is on.

There is clearly a hierarchy of these tools: for example, equivalent pieces of technology are often better on a vertex than on an edge, and CCTv is better than photo-radar. More information is always better than less, and photo-radar is better than an alarm on the same edge. It is not clear, however, where the traps fit in. For instance, is it more effective to have a camera than a trap?

A typical proof with these additional devices comes in two parts. First, the cops employ a strategy that either captures the robber or causes him to move onto a vertex or an edge which has the appropriate technology. Second, once the robber's position is identified, a "cop-win"-like strategy is used in which the cop moves toward the robber limiting his manoeuvring room.

We consider photo-radar in some detail. We will survey the results for the other technologies but omit the details. Let G be a cop-win graph and let $pr(G)$ be the least number of photo-radar units required so that there is some arrangement which allows a single cop to capture the robber.

Theorem 8.1 ([**58**]). *For all positive integers $n > 0$, there exists a tree T such that $pr(T) > n$.*

Proof. A tree (as in Figure 8.1) rooted at a vertex with $n + 3$ arms each of length 2 suffices. □

An edge is *free* if it has no photo-radar. A path P is said to be *free* if every edge of P is free. Let T be a tree. Let T_a be the tree T rooted at vertex a. An *a-branch* of T_a is a path of T with a as one

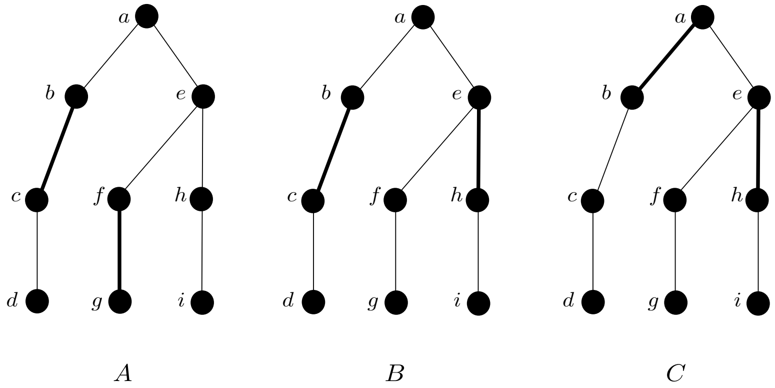


Figure 8.3. A tree with some placements of two photo-radar units.

end-vertex. Define $k(T_a)$ as the minimum number of edges having photo-radar such that there is an arrangement of the units in which the free edges form free paths and each maximal free path is on an a-branch.

The arrangement of photo-radar units (represented by the thick edges) in Figure 8.3 A has a maximal free-path $fehi$ which is not contained within an a-branch. The arrangement in Figure 8.3 B has the maximal free path $baefg$ which is not contained within an a-branch either. Each of the free paths in Figure 8.3 C are all contained in an a-branch.

Define $T' = T \setminus \{v \in V(T) : v \text{ is an end-vertex}\}$, and set

$$k_T = \min\{k(T'_a) : a \in V(T)\}.$$

See Figure 8.4.

Theorem 8.2 ([**58**]). *If T is a tree, then $pr(T) \leq k_T$.*

Proof. We only sketch the proof. A free path together with adjacent leaves will be called a *free area.* The algorithm to capture the robber has two parts. Root the tree T at the vertex v attaining the minimum in the definition of k_T.

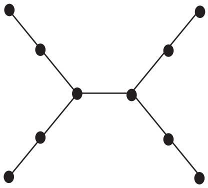


Figure 8.4. A tree T with $pr(T) = k_T = 2$.

Part 1 of the Algorithm: Forcing the robber to reveal his position. The cop carries out a depth first search of T'_a starting at v with two provisos: (1) when he comes to a vertex x joined to a end-vertex, he visits any leaves adjacent to x before moving on; (2) when he has a choice of which edge to move down (that is, away from the root) he chooses a free edge before choosing an edge with a unit. The cop always enters a free path at the end closest to the root (never in the middle by the placement of the units) and exits at the other without leaving the free path, except possibly, for leaves. When on a free path, the still undetected robber can never move past the cop even if the cop is inspecting leaves. Thus from the moment a free area, say F, has been searched, the cop knows that the only way a robber could be on F is if he has used an edge with a photo-radar unit. If the robber stays on a free area, then he will be caught since every free area is searched. If he does move off, then he will be detected by a photo-radar unit and the cop will always know the free area in which the robber is located.

Part 2 of the Algorithm: The robber has been detected. Once the location (that is, the free area) of the robber is known, the cop moves up the tree until he is on a vertex which lies above the free area on which the robber is currently located. This free area could change as the cop moves, but the cop will know which free area the robber occupies and the cop can always move back to v if necessary. Assuming that the robber is not caught in this repositioning, the cop then starts down the v-branch that contains the robber until she enters the same free area as the robber. The robber can move to a different free area but this move will be detected by the photo-radar units and the cop will always move so as to be above the robber.

Again, the robber cannot move up past the cop. By moving down the free path and visiting adjacent leaves, the cop now eliminates the free areas reducing the area that the robber can occupy until the robber is eventually captured. □

How do we translate this algorithm to a general cop-win graph G? Fix a cop-win spanning tree, say T_v, of G, and put a photo-radar unit on every edge of $E(G) \setminus E(T_v)$. Consider T_v and put $k(T'_v)$ units on T_v (so that the free edges form free paths and each maximal free path is on an v-branch). The argument for Theorem 8.2 would seem to generalize except for one problem: the robber can change arms on the T_v tree. The cop will know this because all non-tree edges have units, but even if the cop is above the robber on one arm, can the robber move so as to leave the cop with no move that puts him above the robber? And even if there is such a move, can the robber move around in a cycle forcing the cop to move on a "higher" cycle, thereby avoiding capture forever? The answer to both is no.

Define the parameter

$$K_G = \min_{T'_v}\{k(T_v) : T_v \text{ is a cop-win spanning tree with root } v\}.$$

Theorem 8.3 ([**58**]). *If G is a cop-win graph with $|V(G)| = n$, then*

$$pr(G) \leq |E(G)| - (n-1) + K_G.$$

This can be proven by induction by taking a cop-win spanning tree and removing an end-vertex (see Exercise 3). However, [**58**] takes a more direct approach. Even though the cop-win spanning tree concept had been around for a while, nowhere in the literature, before [**58**], had anyone considered the other edges of the graph. More specifically, if $x \sim y$ but are on different branches of the cop-win spanning tree, then we consider what other edges are forced. A cop-win spanning tree from a specific cop-win ordering has another partial order on the vertices: $x \preceq y$ if y is eventually retracted onto x, and $x \prec y$ if $x \neq y$.

Lemma 8.4 ([**58**]). *Let G be a cop-win graph with cop-win spanning tree T_v, and let B and C be two v-branches of T_v. If there exist vertices $x \in B$ and $y \in C$, $x \simeq y$, then for all $p \succeq x$ there exists $q \succeq y$ such that $p \simeq q$.*

Let B and C be two v-branches of a cop-win spanning tree T_v. Suppose $b \in B$ and b is adjacent to some vertices of C. Let $c \in C$ to be the lowest (furthest away from v) vertex in C that is adjacent to b and write $b \rightarrow c$. In the strategy given in [**58**], the cop will usually take the edge bc if the cop is on b and the robber, below him on B, moves to the C. The main lemma used to prove that the robber cannot force a cycle around the arms of a cop-win spanning tree is the following.

Lemma 8.5 ([**58**]). *Let G be a cop-win graph with cop-win spanning tree T_v, and let B and C be two v-branches of T_v. If $p, x \in B$ and $q, y \in C$ with $x \prec p$, $x \sim y$ and $p \rightarrow q$, then either $y \prec q$ or $y \sim p$.*

In other words, suppose that each of the following items hold.

(1) Vertices x and p are on the branch B with x higher than p.

(2) Vertices y and q are on the branch C. where q is the lowest vertex on C that is adjacent to p.

(3) Vertex x is adjacent to y.

Then Lemma 8.5 tells us that either q is lower than y or p is adjacent to y.

8.2. Traps

Traps can either capture or impede the robber's progress. Either way, the authors improve the "power" of the cops. With sufficient traps, of any description, a graph is cop-win (see Exercise 5). Surprisingly, little is known about using traps. Only two papers [**59, 161**] (the latter one unpublished) on traps exist at the time of writing.

We mention in passing that in [**56**], the authors consider graphs in which deleting any edge changes the cop number from two to one or vice-versa. That could be considered as placing a trap that captures both cops and robber so they both avoid it. There are graphs where the cop number can increase after deleting an edge (any tree, for example).

In the unpublished paper [**161**] by Musson and Tang, their constraints are that the cop has visibility 1; that is, he can see if the

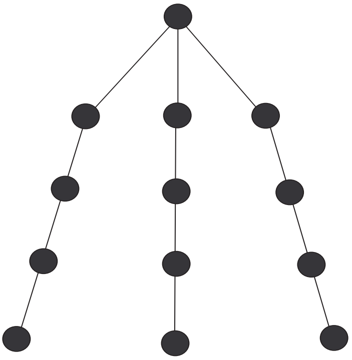


Figure 8.5. The tree M.

robber is on any adjacent vertex. Once a road block is placed, the robber cannot use that edge, but the cop can. The main question is of course, *given a graph G what is the minimum number of road blocks*? This number they call $\mathrm{rb}(G)$. Their main results are for trees. Let M be the tree in Figure 8.5.

Theorem 8.6 ([**161**]). *If T is a tree which has no subtree isomorphic to M, then $\mathrm{rb}(T) = 0$. In general, if T contains k disjoint copies of M, then $\mathrm{rb}(T) \geq k$. Moreover, $\mathrm{rb}(T) = 1$ if and only if all subgraphs of T isomorphic to M have at least one edge in common.*

Like the photo-radar problem, they conjecture that for an arbitrary graph G, a good and frequently best approach is to find the spanning tree T with fewest copies of M, put road blocks on all the edges of $G - T$, plus the necessary ones for T. As evidence they give the following result.

Theorem 8.7 ([**161**]). *If G is a triangle-free graph with n vertices and m edges, then*

$$\mathrm{rb}(G) = m - n + 1.$$

Further, if G contains a spanning tree which is does not contain M as an induced subgraph, then

$$\mathrm{rb}(G) = m - n + 1.$$

They also show the following.

Theorem 8.8 ([**161**]). *If H is a connected induced isometric subgraph of G, then*

$$\mathrm{rb}(H) \leq \mathrm{rb}(G).$$

In [**59**], there is perfect information and the traps are placed on vertices but are moveable (something akin to tire deflation devices, informally known as spike strips). To place a trap, the cop has to visit the vertex to deploy it, and to move it he has to pick it up and then place it.

If the cops have a winning strategy on a graph G with n cops and m traps, then G is referred to as (n, m)*-win*. Note that a cop-win graph is $(1, 0)$-win. A cycle of length at least 4 is $(2, 0)$-win, but it is also $(1, 1)$-win. We observe that the complete bipartite graph $K_{n+2,n+2}$ is $(2, 0)$-win but not $(1, n)$-win (see Exercise 6).

Surprisingly, retracts are important.

Theorem 8.9 ([**59**]). *If H is a retract of G and G is (m, n)-win, then H is (p, q)-win, for some $p \leq m$ and $q \leq n$.*

There is a result corresponding to that of Theorem 1.11 from [**16**].

Theorem 8.10 ([**59**]). *Let H be a retract of G with H $(n_0, 0)$-win, and let $G - H$ be (n_2, m_2)-win. Then G is (n_1, m_1)-win, where $m_1 \leq m_2$ and $n_1 \leq \max\{n_0, n_2 + 1\}$.*

A *linear layout* of a graph G is simply an ordering of the vertices. (Usually when a concept is defined using a linear layout, one thinks of placing them in a row on a page, and then the concept involves a property of the drawing of the graph. Although a cop-win ordering is an ordering of the vertices, it is not referred to as a linear layout since no property of the actual drawing is used.) Let $L = \{v_1, v_2, \ldots, v_n\}$ be a linear layout of G, and set

$$L_i = |\{j : j \leq i \text{ and for some } k > i,\ v_j \sim v_k\}|.$$

Intuitively, having placed the vertices in a horizontal line from left to right and drawn the edges, L_i is the number of vertices to the left of

v_i which have neighbors to the right of v_i. Let $\widehat{L} = \max_i\{L_i\}$. The *vertex separation number* of G is

$$\text{vs}(G) = \min\{\widehat{L} : L \text{ is a linear layout of } G\}.$$

This is also known as the *pathwidth* of G. The vertex separation number is a (usually bad) upper bound for the cop number and has more relevance for searching a graph with hidden intruders; see Section 9.4. However, it appears to be relevant in this context.

Theorem 8.11 ([**59**]). *A graph G is $(1, \text{vs}(G) + 1)$-win. Moreover, if G is (n, m)-win but not $(n - 1, m)$-win or $(n, m - 1)$-win, then*

$$n + m \leq \text{vs}(G) + 2.$$

In an attempt to characterize $(1, 1)$-win graphs, we mention a set of vertices called *handles*. A handle is something like a corner but the definition is more technical, and not easy to recognize since it involves many cop-win orderings. We refer the interested reader to the paper [**59**] for the actual definition. A result of the following form is false, but may be close to the truth: "*A graph G can be reduced to a cop-win graph by corner and handle retractions if and only if G is $(1, 1)$-win.*" Here are the corresponding results for handles.

Theorem 8.12 ([**59**]).
(1) *Let G be a graph, and let H be a handle in G. If the robber is forced to move onto H, then the cop will win.*
(2) *Let G be a graph which can be reduced to a single vertex by corner and handle retractions. If there is such a sequence of retractions in which no vertex is retracted to any vertex in a handle, then G is $(1, 1)$-win.*

Note that $(1, 1)$-win graphs are a subset of the 2-cop-win graphs, since the second cop can be left in lieu of the trap. They are also a superset of the graphs that require one road-block. In [**56**], the authors consider graphs that are 2-cop-win, edge-critical graphs (that is remove any edge and the remaining graph is cop-win). The $(1, 1)$-win graphs would also contain such graphs. We finish with two interesting open problems on (n, m)-win graphs.

(1) Is there a structural characterization of $(1,1)$-win graphs?

(2) Given a graph G, what is the minimum p such that G is $(n, p-n)$-win?

8.3. Tandem-win

As a small step toward finding a characterization of graphs with cop number 2, Clarke and Nowakowski [**61**] introduced the concept of *tandem-cops*. The intuitive idea is that two cops "patrol" together—they stay within close proximity of each other. Graph theoretically, the two cops must always be on the same or adjacent vertices at the end of each move. As an example, if a 4-cycle with vertices $1, 2, 3, 4$ has one cop on 1 and the other on 2, then they can move to 4 and 3, respectively. A graph is *tandem-win* if one pair of tandem-cops suffices to capture the robber. A cop-win graph is also a tandem-win graph, but since C_4 is also tandem-win, the class of tandem-win graphs strictly contains cop-win graphs; see Figure 8.6.

For cop-win graphs, corners are all-important. A vertex v is *o-dominated* (or *nearly irreducible*; see Exercise 10 in Chapter 2) if there exists a vertex $y \neq v$ such that $N(v) \subseteq N[y]$. An o-dominated vertex looks like a corner but without the requirement that $v \sim y$. In characterizing cop-win graphs, instead of Theorem 2.3, one could write the following.

Theorem 8.13. *Let v be a corner of G. Then G is cop-win if and only if $G - v$ is cop-win.*

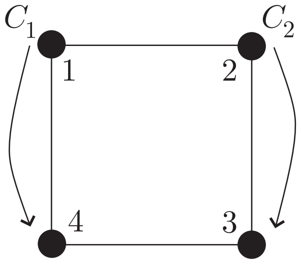


Figure 8.6. A legal move of tandem-cops.

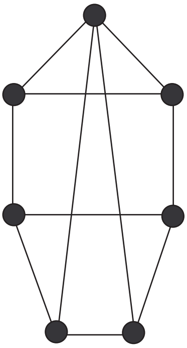


Figure 8.7. A tandem-win graph with no o-dominated vertex.

In [**61**], there is a theorem very similar to Theorem 8.13.

Theorem 8.14 ([**61**])**.** *Let v be o-dominated in a graph G. Then G is tandem-win if and only if $G - v$ is tandem-win.*

The characterization for cop-win graphs is completed by applying Lemma 2.1, which states that a cop-win graph has a corner. Unfortunately, there is no equivalent statement for tandem-win graphs. Figure 8.7 is a tandem-win graph but has no o-dominated vertex (see Exercise 9).

The results for products are not dissimilar from those for the usual Cops and Robbers game (see Chapter 4), and can be proved relatively quickly given those results. For a graph G, let $t(G)$ be the least number of tandem-cops needed to capture the robber. For the Cartesian product there is nothing surprising.

Theorem 8.15 ([**61**])**.** *Let G and H be graphs.*

(1) *If G is cop-win and H is a tree, then $t(G\square H) = 1$.*

(2) $t(G\square H) \leq t(G) + t(H)$.

For the strong product the upper and lower bounds are quite different.

Theorem 8.16 ([**61**]). *Let G and H be graphs.*

(1) *If G is cop-win and H is tandem-win, then $t(G \boxtimes H) = 1$.*

(2) *If for a family of graphs $(G_i : 1 \leq i \leq n)$ we have that $t(G_i) = 1$, then $t(\boxtimes_{i=1}^{n} G_i) \leq 2^{n-1}$. Moreover,*
$$t((C_4)_{\boxtimes}^{2n}) > n.$$

We leave the proofs as exercises, but sketch the proof of the lower bound in Theorem 8.16 (2). Suppose n tandems of cops choose their vertices. The robber then chooses a vertex such that on the ith factor of the product, the robber's projection is two away from the projection of the ith cop. In one move, no cop can capture the robber on all the projections, and thus, not on $(C_4)_{\boxtimes}^{2n}$. Thereafter, the robber moves to maintain these distances. We mention the following open problem: What are the proper bounds for Theorem 8.16 (2)?

For the categorical product even less is known. We showed in Chapter 4 that the cop number of the categorical product of two cop-win graphs is at most three, and also that if G and H are non-bipartite graphs with $c(H) \geq c(G)$ and $c(H) \geq 2$, then
$$c(G \times H) \leq 2c(G) + c(H) - 1.$$
In [**60**], it is shown that two tandem-cops suffice and are necessary on the categorical product of certain tandem-win graphs.

A graph G has a *special tandem-win decomposition* by o-dominated vertices if there is a set of 1-point retractions that reduce the graph to a single vertex in which all leaves are retracted (as o-dominated vertices) before any other vertices, and then the o-dominated vertices are retracted.

Theorem 8.17 ([**61**]). *Let G and H be triangle-free tandem-win graphs, each having at least one cycle. If G and H have special tandem-win decompositions, then $t(G \times H) = 2$.*

8.4. Playing on Different Edge Sets

On a whimsical note, the nature of the cops and the robber is diametrically opposite, so why not have them play on different sets of edges? The natural situation is for the cops to have one set of edges and the robber to have the complementary set. Precisely, for a graph G the cops move on $E(G)$ while the robber moves on $E(\overline{G})$. Alan Hill [**113**] called this variant *complementary Cops and Robbers*, and for a graph G the least number of cops required to catch the robber is denoted by $CC(G)$. An intriguing result for this parameter is that it is within one of the domination number.

Theorem 8.18 ([**113**]). *If G is a graph, then*

$$\gamma(G) - 1 \leq CC(G) \leq \gamma(G).$$

The upper bound is trivial, and the lower bound is also straightforward; see Exercise 12. It is **NP**-complete to determine the minimum domination number of a graph, and we observe that the same can be said for the parameter $CC(G)$.

Corollary 8.19. *For an arbitrary graph G, it is* ***NP****-complete to determine $CC(G)$.*

Proof. If $CC(G)$ could be determined in polynomial time, then one only has to test all subsets of $V(G)$ of size $CC(G)$ and $CC(G) + 1$ to determine $\gamma(G)$. □

The graph G, which consists of two copies of with $K_{1,n}$ with one end-vertex in each copy identified, has $\gamma(G) = 2$ and $CC(G) = 2$. Further, $\gamma(K_{m,n}) = 2$, but $CC(K_{m,n}) = 1$; see Exercise 13. As a special case, it is straightforward to show that $CC(C_n) = k$ for $n = 3k - 1, 3k$ or $3k + 1$.

Hill has one further result on the structure of graphs needing k cops.

Lemma 8.20 ([**113**]). *If $CC(G) = k$, then G has a set of $k + 1$ vertices at least two of which are adjacent, that dominate the graph.*

Proof. Consider the final round of play. The robber must be on a vertex which is adjacent to a cop; moreover, all vertices not adjacent to the robber must be adjacent to a vertex occupied by a cop. Thus, the robber's and the k cops' positions form a dominating set of size $k + 1$, with two of the vertices being adjacent. □

In many situations, the structure of the lexicographic product of two graphs G and H allows for more efficient use of "resources". In particular, the fractional (that is, LP relaxation) version of a graph problem can be phrased in terms of the lexicographic product of G with itself. Hill asked whether there exist graphs G and H such that $CC(G \bullet H) < CC(G)$. As evidence, he proves the following.

Theorem 8.21 ([**113**]). *If G and H are connected graphs with $CC(G)$ at least two, then $CC(G \bullet H) \leq 2CC(G)$.*

A different approach to playing on different edge sets was taken by Neufeld and Nowakowski [**165**]. They considered the case where the vertex set of the graph was the Cartesian product of the vertex sets of G and H, and the available edges for the cops came from one product whereas the robber edges came from a different product. To this end, let $XY(G \otimes H)$ denote the number of cops required to capture a robber when the cops are restricted to the edges of the product graph X and the robber to those in the product graph Y. For a sequence of graphs $(G_i : 1 \leq i \leq n)$ they defined $XY(\bigotimes_{i=1}^{n} G_i)$ in the analogous manner. They focused on the edges from the categorical, Cartesian and strong products. Let S be the edges of the Cartesian product (**S**traight edges), $\overline{S}$ be the edges of the strong product minus the edges of the Cartesian product, O be the edges of the categorical product (**O**blique edges), and $\overline{O}$ be the edges of the strong product minus the edges of the categorical product. Note that in the product of two graphs, the edges of the strong product is the union of the edges in the Cartesian and categorical products, but that is not true for products of three or more graphs. In general, $S \subseteq \overline{O}$ and $O \subseteq \overline{S}$ so that we have the following inequalities:

$$S\overline{S}(\bigotimes_{i=1}^{n} G_i) \geq SO(\bigotimes_{i=1}^{n} G_i) \geq \overline{O}O(\bigotimes_{i=1}^{n} G_i)$$

and

$$\overline{S}S(\bigotimes_{i=1}^{n} G_i) \leq OS(\bigotimes_{i=1}^{n} G_i) \leq O\overline{O}(\bigotimes_{i=1}^{n} G_i).$$

The growth of the number of edges is exponential in the case of the strong and the categorical but only linear for the Cartesian. For example, if G is r-regular, then in the kth power of G, the degree of each vertex is rk in the Cartesian product, $(r+1)^k - 1$ in the strong product, and is r^k in the categorical.

Often the strategy given is a combination of catching the projection in one graph and using a strategy borrowed from graph searching (see Section 9.4). Take a linear layout that realizes the vertex separation number, the $\mathrm{vs}(G)$ cops occupy a consecutive string of vertices in the layout, and the free cop moves so as to occupy the next vertex, thereby freeing up the cop occupying the first vertex in the string. (This is akin to "rolling a prime" in Backgammon.) In this context, if the vertex separation appears in the upper bound, then it would appear that the robber has a great advantage using this set of edges.

Define $\overline{\beta}(G) = |V(G)| - \beta(G)$, where $\beta(G)$ is the *independence number* of G (that is, the cardinality of a largest independent set in G). For the parameter $SO(G \otimes H)$, the cop has a linear increase in the available edges but the robber has the quadratic increase so one would expect that the robber has the advantage.

Theorem 8.22 ([**165**]). *If G and H are connected graphs, then $SO(G \otimes H)$ is bounded above by*

$$\min\{c(H) + vs(G), c(G) + vs(H), c(H) + \overline{\beta}(G) - 1, c(G) + \overline{\beta}(H) - 1\}.$$

Further,

$$SO(G \otimes H) \geq \max\{c(G), c(H)\}.$$

Note that $\overline{\beta}(G) - 1$ is usually greater than $\mathrm{vs}(G)$ but not for complete graphs. In fact,

$$SO(K_m \otimes K_n) = \min\{m, n\} - 1$$

(see Exercise 14).

When the edge sets are reversed, the cops should have the advantage. Indeed only the graph with the larger cop number need be considered.

Theorem 8.23 ([**165**]). *Let G and H be finite connected graphs both with at least two vertices.*

(1) *If $c(G) = c(H) = 1$, then $OS(G \otimes H) = 2$.*

(2) *If $c(G) \geq c(H) \geq 1$, then $c(G) \leq OS(G \otimes H) \leq c(G) + 1$.*

In the product of at least three graphs, if the robber is restricted to just the Cartesian edges, then the situation is better for the cops.

Theorem 8.24 ([**165**]). *If $c(G_i) \geq c(G_{i-1})$ for $i = 2, 3, \ldots, n$, then the following items hold.*

(1) *If $c(G_n) = 1$, then $\overline{S}S(\bigotimes_{i=1}^{n} G_i) \leq 2$ for $n = 2, 3$ and $\overline{S}S(\bigotimes_{i=1}^{n} G_i) = 1$, otherwise.*

(2) *If $c(G_n) > 1$, then $\overline{S}S(\bigotimes_{i=1}^{n} G_i) = c(G_n)$ for $n \geq 3$.*

When the situation is reversed the cops have a harder time. We use the notation $XY(G^n)$ for $XY(\bigotimes_{i=1}^{n} G_i)$ where $G_i = G$ for all i.

Theorem 8.25 ([**165**]). *Let $\delta = \delta(G)$ and $v = |V(G)|$. Then*

$$\frac{(\delta+1)^n - n\delta}{n\delta + 1} \leq S\overline{S}(G^n) \leq (c(G) + \mathrm{vs}(G))v^{n-2}.$$

The edges sets O and $\overline{O}$ both grow exponentially with the numerical advantage to the player on the $\overline{O}$ set. It is surprising that the cops do not have a better advantage in the $\overline{O}O$ case.

Theorem 8.26 ([**165**]). *If G is a graph, then for $n > 1$ we have the following.*

(1) $\overline{O}O(G^n) \leq (c(G) - 1)(n - 1) + \min\{\mathrm{vs}(G) + 1, \overline{\beta}(G)\}$.

(2) $\overline{O}O(G^n) \geq (c(G) - 2)(n - 1)$.

With the edge sets reversed, the robber again has a definite advantage.

Theorem 8.27 ([**165**]). *Let G be a finite graph, $\delta = \delta(G) \geq 2$ and $c(G) \geq 2$. Then for $n > 2$ we have the following.*

(1) $\left(1 + \frac{1}{\delta}\right)^n - 1 \leq O\overline{O}(G^n) \leq 2^{n-1}c(G)$.

(2) *If the girth of G is at least 5, then $O\overline{O}(G^n) \geq 2^{n-1}$.*

8.5. Distance k Cops and Robbers

We consider a variation of Cops and Robbers where a cop need not occupy the vertex of the robber to capture him, but must only “see” or “shoot” the robber from some prescribed distance away. For analogies from computer gaming, consider first-person shooter games where weapons hit targets at some prescribed distance (so-called "hitscan"). More precisely, fix a non-negative integer parameter k. The game of *distance k Cops and Robbers* is played in a way analogous to Cops and Robbers, except that the cops win if a cop is within distance at most k from the robber. If $k = 0$, then distance k Cops and Robbers reduces to the classical Cops and Robbers game studied in the first seven chapters of this book. The minimum number of cops possessing a winning strategy in G playing distance k Cops and Robbers is denoted by $c_k(G)$.

Observe that $c_0(G)$ is just the usual cop number $c(G)$, and for all k and j with $k < j$, $c_j(G) \leq c_k(G)$. For a basic example illustrating that something new comes about by considering $k > 0$, note that $c_0(C_4) = 2$, while $c_k(C_4) = 1$ for all $k \geq 1$.

Distance k Cops and Robbers and the parameters c_k were introduced in [**23**] as a generalization of the classic game. We highlight some of the main findings from that work and finish with some new work characterizing graphs satisfying $c_1(G) = 1$ in [**43**].

First, we consider bounds on c_k. As an extension of Theorem 3.1, there is an analogous upper bound on $c_k(n)$. Let $c_k(n)$ be the maximum value of $c_k(G)$, where G is a graph of order n.

Theorem 8.28 ([**23**]). *For integers $n > 0$ and $k \geq 0$ (where k can be a function of n)*

$$c_k(n) \quad = \quad O\left(\frac{n\log(k+2)}{(k+1)\log\left(\frac{2n}{k+1}\right)}\right).$$

As a generalization of Meyniel's conjecture (discussed in Chapter 3, at length), it is conjectured that for all $k \geq 1$

$$c_k(n) = \Theta\left(\left(\frac{n}{k}\right)^{1/2}\right). \tag{8.1}$$

Given a graph G and a positive integer ℓ, form $G^{(\ell)}$ by replacing each edge of G by a path with ℓ edges. For example, $K_4^{(2)}$ is illustrated in Figure 8.8. For simplicity, we identify vertices of G

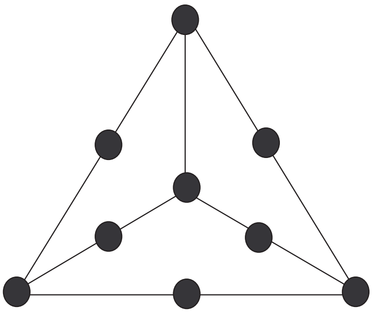


Figure 8.8. The graph K_4^2.

with corresponding vertices in $G^{(\ell)}$; in particular, $V(G) \subseteq V(G^{(\ell)})$. Vertices of $G^{(\ell)}$ that are not in G are called *internal vertices.*

The usual cop number and the parameters c_k are related as per the following lemma.

Lemma 8.29 ([**23**])**.** *For any graph G and any integer $k \geq 0$,*

$$c(G) \leq c_k(G^{(2k+1)}) \leq c(G) + 1.$$

Lemma 8.29 sets up a relationship between $c(G)$ and $c_k(G)$. We note that either of the two values bounding $c_k(G^{(2k+1)})$ in the lemma may be realized. For example, $c_1(K_3^{(3)}) = 2$ with $c(K_3) = 1$, while $c(G) = c_k(G^{(2k+1)})$ if G is a tree.

Proof of Lemma 8.29. Joret et al. [**125**] proved that

$$c(G^{(2k+1)}) \leq c(G) + 1.$$

Since $c_k(G^{(2k+1)}) \leq c(G^{(2k+1)})$, it remains to prove that $c(G) \leq c_k(G^{(2k+1)})$.

Let $c = c(G) - 1$. The robber R has a winning strategy in Cops and Robbers played on G if there are only c cops. We will show that R has a winning strategy in distance k Cops and Robbers played on $G^{(2k+1)}$ if there are only c cops.

For each internal vertex $x \in V(G^{(2k+1)})$, there is exactly one vertex in $V(G)$ whose distance from x is at most k; name this vertex

x_k. Define a function f from the vertices of $G^{(2k+1)}$ to vertices of G that is the identity on $V(G)$, so that if x is internal vertex, then $f(x) = x_k$. The robber R simulates the winning strategy for Cops and Robbers played on G in distance k Cops and Robbers played on $G^{(2k+1)}$ by using the function f, and will play in a way that the robber will always be in $V(G)$ in rounds

$$2k, 4k+1, \dots, 2ik+i-1, \dots$$

for all $i \geq 1$.

In round 0, C puts c cops in $v_1, v_2, \dots, v_c$. In round 0, R assumes that the cops are at $f(v_1), f(v_2), \dots, f(v_c)$ and puts the robber in a vertex $r \in V(G)$ pretending that the game is being played in G. Since the robber would not be captured in G, neither of $f(v_i)$'s are adjacent to r in G, and hence, v_i's are of distance at least $3k+2$ from r in $G^{(2k+1)}$. Therefore, the cops cannot capture the robber in rounds $0 \leq t \leq 2k+1$, if the robber stays at r in rounds $0 \leq t \leq 2k$.

Let $v'_1, v'_2, \dots, v'_c$ be the positions of cops in round $2k+1$. In $2k+1$ rounds, for each $1 \leq i \leq c$, we will have either $f(v_i) = f(v'_i)$ or $f(v_i)$ is adjacent to $f(v'_i)$ in G. Thus, R can assume that the cops have moved from $f(v_1), f(v_2), \dots, f(v_c)$ to $f(v'_1), f(v'_2), \dots, f(v'_c)$ in G in one round. Let r' be the vertex to which the robber would move to if the game was being played in G. The strategy of R in $G^{(2k+1)}$ is to move the robber from r to r' in the next $2k+1$ rounds. The cops cannot capture the robber in the next $2k+1$ rounds and, in round $4k+2$, the robber can decide the next $2k+1$ rounds. The rest follows by induction. □

Lemma 8.29 gives us a tool for transfering lower bounds on $c(n)$ to lower bounds on $c_k(n)$.

Theorem 8.30 ([**23**]). *For all $k \geq 1$ and $n \geq 1$ integers, we have that*

$$c_k(n) \geq \left(\frac{n}{k}\right)^{1/2+o(1)}.$$

Proof. Consider a random graph $G = G(n,p)$ with average degree $np = 3\log n$. Then a.a.s. G is connected, and by Theorem 6.11,

$c(G) = n^{1/2+o(1)}$ a.a.s. Now by Lemma 8.29 a.a.s. we have that

$$c_k(G^{(2k+1)}) \geq c(G) = n^{1/2+o(1)}.$$

Since a.a.s. $N = |V(G^{(2k+1)})| = \Theta(k|E(G)|) = kn^{1+o(1)}$, the proof follows since a.a.s.

$$c_k(G^{(2k+1)}) \geq \left(\frac{N}{k}\right)^{1/2+o(1)}. \qquad \square$$

Giving the cops a longer reach, as in distance k Cops and Robbers, does not change the complexity of the underlying game. Using strong products, it was shown in [**23**] that computing if $c_k(G) \leq s$ for a fixed s is in **P**.

Theorem 8.31 ([**23**])**.** *The problem of computing if $c_k(G) \leq s$ for a fixed s has complexity $O(n^{2s+3})$.*

Note that the bound in Theorem 8.31 is independent of k. A result of Fomin et al. [**85**] states that there is a constant $c > 0$ such that there is no polynomial-time algorithm to approximate $c(G)$ within ratio $c \log n$, unless **P**=**NP**. Combining this fact with Lemma 8.29 gives the following corollary.

Corollary 8.32 ([**23**])**.** *For any integer $k \geq 0$, computing $c_k(G)$ is **NP**-hard.*

Proof. Assume that there is an integer k and a polynomial-time algorithm A such that $A(G) = c_k(G)$, for all graphs G. Let B be a polynomial-time algorithm such that $B(G) = G^{(2k+1)}$, for all graphs G. By Lemma 8.29, it follows that the composition of the algorithms A and B is a polynomial-time 2-approximation algorithm for computing $c(G)$. $\square$

We may consider the parameters c_k in random graphs. For a fixed integer $k \geq 0$, function $f_k : (0,1) \to \mathbb{R}$ defined as

$$f_k(x) = \frac{\log \overline{c_k}(G(n, n^{x-1}))}{\log n},$$

where $\overline{c_k}(G(n,p))$ denotes the median value of the distance k cop number for $G(n,p)$. Recall the Zig-Zag Theorem (Theorem 6.16),

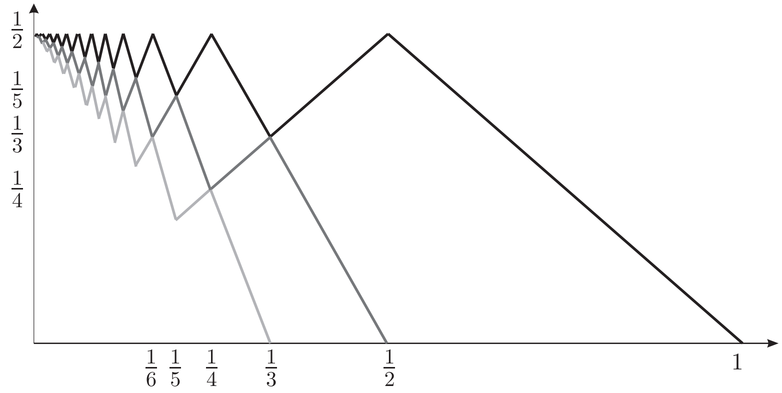


Figure 8.9. The functions f_k, for $k = 0$, 1, and 2, with the darker lines representing smaller values of k.

which described the behaviour of f_0. Generalizing to $k > 0$, we actually find infinitely many zig-zags for all $k \geq 0$, as described in the following theorem; see Figure 8.9 for the functions f_k in the cases $k = 0, 1, 2$.

Theorem 8.33 ([**23**]). *Let $k \geq 0$, $0 < \alpha < 1$, and $d = d(n) = np = n^{\alpha+o(1)}$.*

(1) *If $\frac{1}{2j+1+k} < \alpha < \frac{1}{2j+k}$ for some $j \geq 1$, then a.a.s.*

$$c_k(G(n,p)) = \Theta(d^j)\,.$$

(2) *If $\frac{1}{2j+k} < \alpha < \frac{1}{2j-1+k}$ for some $j \geq 1$, then a.a.s.*

$$\Omega\left(\frac{n}{d^{j+k}}\right) = c_k(G(n,p)) = O\left(\frac{n \log n}{d^{j+k}}\right)\,.$$

No structural characterization of graphs G satisfying $c_k(G) = 1$, where $k \geq 1$ is a fixed integer, is known. This appears to be a difficult problem, where very little is known except in special cases. In the case when G is bipartite and $k = 1$, a characterization was reported in [**43**]. A bipartite graph G is *almost-dismantlable* if its vertices can be ordered $v_1, \ldots, v_n$ so that $v_{n-1}v_n$ is an edge of G, and for each v_i where $i < n$, there exists a vertex $y = v_j$ with $j > i$ (necessarily not adjacent to v_i) such that

$$N[v_i] \cap \{v_{i+1}, \ldots, v_n\}) \subseteq N[y].$$

Theorem 8.34 ([**43**]). *If G is bipartite, then $c_1(G) = 1$ if and only if G is almost-dismantlable.*

8.6. Capture Time

A recent variation on the cop number is to consider not how many cops are needed to capture the robber, but rather how long it takes them to capture the robber. To be more precise, the *length* of a game is the number of rounds it takes (not including the initial or 0th round) to capture the robber. Equivalently, the length of the game equals the number of rounds needed for the cop to capture the robber (the degenerate case is the game played on K_1 which has length 0). We say that a play of the game with $c(G)$ cops is *optimal* if its length is the minimum over all possible games played by the cops, assuming the robber is trying to evade capture for as long as possible. There may be many optimal plays possible (for example, on P_4, the cop may start on either vertex of the center), but the length of an optimal game is an invariant of G. In a graph with $c(G) = k$, we denote this invariant $\mathrm{capt}_k(G)$, which we call the *k-capture time* of G. In the case $k = 1$, which will be our focus, we just write $\mathrm{capt}(G)$. The capture time parameters may be viewed as temporal counterparts to the cop number, and were introduced in [**26**]. Time is a well-measured resource in graph algorithms (see Chapter 5), which inspired this approach to the Cops and Robbers game. We note in passing that in the recent work [**86**], a variant of Cops and Robbers is studied where every cop can make at most a fixed number of steps to capture the robber.

We recall the following upper bound from Chapter 2 (stated with our new notation).

Theorem 8.35. *If G is cop-win of order $n \geq 5$, then* $\mathrm{capt}(G) \leq n-3$.

By considering small order cop-win graphs, the bound in Theorem 8.35 was improved to $\mathrm{capt}(G) \leq n - 4$ for $n \geq 7$ in [**97**]. As we will see below, the bound of $n - 4$ is optimal.

In many cop-win graphs such as trees, the cop can win in much fewer than $n-4$ moves. Two corners a and b in a cop-win graph G are *separate* if neither is dominated only by the other. For example, two distinct end-vertices in a tree are separate corners. We say that a graph G is 2-*dismantlable* if it is cop-win, has two separate corners a and b, and $G - \{a, b\}$ either has two separate corners or has fewer than seven vertices. Observe that deleting two corners from a large enough 2-dismantlable graph leaves an induced subgraph which is a 2-dismantlable, cop-win graph. Each chordal graph is 2-dismantlable as chordal graphs contain at least two simplicial vertices. However, the 4-wheel is 2-dismantlable but not chordal.

As the next theorem demonstrates, the 2-dismantlable graphs have capture time about one half of their order.

Theorem 8.36 ([**26**]). *If G is 2-dismantlable of order n, then*

$$\text{capt}(G) \leq \left\lfloor \frac{n}{2} \right\rfloor .$$

Proof. The proof is by induction on n. We leave the verification of the theorem for $n \leq 6$ as an exercise. Let a and b be two separate corners in a 2-dismantlable graph of order $n \geq 7$, covered by a' and b', respectively. Let H be the induced subgraph formed by deleting a and b. As H is 2-dismantlable, there is an optimal game on H of length at most $\lfloor \frac{n-2}{2} \rfloor$. The cop plays this optimal game in H, and for $x \in \{a, b\}$, whenever R is on x, then C plays as if he were on x'. After at most $\lfloor \frac{n-2}{2} \rfloor$ moves, either the robber is caught on H, or R is on x and C is on x'. But then C can win in one more move, and so this strategy uses at most $\lfloor \frac{n-2}{2} \rfloor + 1 = \lfloor \frac{n}{2} \rfloor$ moves. □

Unfortunately, not every cop-win graph is 2-dismantlable. There are even graphs with a unique corner! For an integer $n \geq 4$, define $G(n)$ by adjoining two vertices x and y joined to each vertex of a path P with n vertices. Add a vertex z that is joined to y and the endpoints of P. Then $G(n)$ is cop-win but z is the unique corner of $G(n)$; see Figure 8.10 for $G(4)$.

Using the graph $G(4)$, we construct an infinite family of graphs of order n with maximum capture time $n-4$. For $n \geq 7$, the graph $H(n)$ has vertices $1, \dots, n$, where $1, 2, 3, 4, 5, 6, 7$ induce $G(4)$ (so that

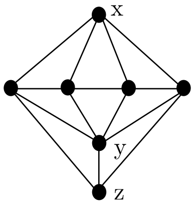


Figure 8.10. The cop-win graph $G(4)$ with a unique corner.

$x = 5, y = 3, z = 7$, and the remaining vertices on the path joined to x and y are (from left to right) $6, 2, 1, 4$). For $i > 7$, the vertex i is joined to $j < i$ if j equals one of $i - 4$, $i - 3$, and $i - 1$. We name the vertices $7, 8, \dots, n$ *special*; see Figure 8.11 for $H(11)$.

The following theorem was proved in [**26**].

Theorem 8.37 ([**26**])**.** *For a fixed integer $n \geq 7$, the graphs $H(n)$ have the following properties.*

(1) *The graph $H(n)$ is planar.*

(2) *The graph $H(n)$ is cop-win and has a unique corner.*

(3) $\text{capt}(H(n)) = n - 4$.

Proof. We leave items (1) and (2) as an exercise (see Exercise 19). For item (3), we present a strategy $\mathcal{S}$ for the cop to win which always

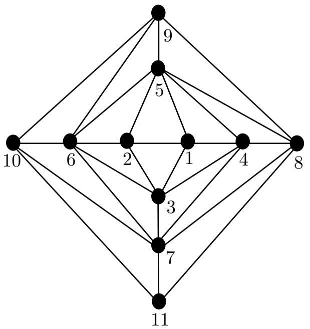


Figure 8.11. The graph $H(11)$.

results in a game of length at most $n-4$. First note that each vertex $5 \leq x \leq n-4$ has neighbors $\{x-4, x-3, x-1, x+1, x+3, x+4\}$. Thus, the cop and robber may move to vertices with index $1, 3$ or 4 more or less than their current index.

The strategy $\mathcal{S}$ has three parts, with the third part repeated until the robber is captured (which we will demonstrate eventually happens).

($\mathcal{S}1$) In the 0th round, place the cop on vertex 1.

($\mathcal{S}2$) After the robber places himself on $i > 1$ in the 0th round, in the first round move the cop to $j \in V(G(4))$ with $j \in \{2, 3, 4, 5\}$ so that $i \equiv j \pmod 4)$.

($\mathcal{S}3$) Repeat the following steps until the robber is eventually caught.

  (a) If robber moves from i to $i+k$, then the cop moves from j to $j+k$, where $k = 1, 3$, or 4.
  (b) If robber moves from i to $i-1$, then the cop moves from j to $j+3$.
  (c) If robber moves from i to $i-3$, then the cop moves from j to $j+1$.
  (d) If robber moves from i to $i-4$, then the cop moves from j to $j+4$.

Let the cop play with $\mathcal{S}$, and let $\text{cop}(t)$ and $\text{robber}(t)$ be the positions of the cop and robber at round t in this game. Note that for all $t \geq 0$, $\text{cop}(t+1) > \text{cop}(t)$. We prove by induction that for all $t \geq 1$,

$$\text{cop}(t) \equiv \text{robber}(t-1) \pmod 4). \tag{8.2}$$

The base case of (8.2) follows by ($\mathcal{S}1$) and ($\mathcal{S}2$). Suppose (8.2) holds for a fixed $t \geq 1$. Suppose that $\text{cop}(t) = j$, with $\text{robber}(t-1) = i$. At time t, the robber moves to $i+m$, where $m \in \{-4, -3, -1, 1, 3, 4\}$.

Then the cop moves at round $t+1$ to $j+m'$ for some m' as instructed by ($\mathcal{S}3$). It is straightforward to check that $i+m \equiv j+m' \pmod 4$ holds for all possible moves of the robber. Hence, the induction step follows.

It follows that the difference of the indices of the cop and robber is kept 0 (mod 4), and when the robber goes to a higher or lower index, the difference is monotonically decreasing. Over time the cop gets closer to the robber. For all rounds except for the last one where the robber is captured, $\mathrm{cop}(t) < \mathrm{robber}(t)$. To complete the proof, we note that the robber can survive $n-4$ moves in $H(n)$ no matter what the cop does (see [**26**]). □

There are many more examples of graphs non-isomorphic to $H(n)$ with capture time $n-4$. An exponential family of such graphs was found in [**98**], and cop-win graphs with maximum capture time $n-4$ were characterized. See also [**97**].

Computing the capture time is a tractable problem. By the results of [**105**], if m is a fixed positive integer, then the problem of determining whether $\mathrm{capt}_k(G) \leq m$ is in **P**. For a non-negative integer t, define c_t to be the minimum number of cops needed to capture the robber in at most t rounds. Note that $c_0(G) = n$ and $c_1(G) = \gamma(G)$. In [**26**], it was shown that the problem of determining if $c_t(G) \leq k$ for a non-negative integer k is **NP**-complete.

The parameters $\mathrm{capt}_k(G)$ for $k > 2$ are not well understood, and a number of questions remain. For example, what are good bounds for these parameters if $k > 1$? How do they behave on graph products? Can we classify the k-cop-win graphs whose capture time is maximum?

Exercises

1. Find all directed cop-win spanning trees in C_4 with a chord.
2. Show that the number of directed cop-win spanning trees of K_n is $n!(n-1)!$
3. Prove Theorem 8.3 by induction.
4. Show how Lemma 8.5 prevents the robber from going indefinitely around in a cycle.

5. Suppose the cop has traps which cause the robber to lose a turn whenever he enters one.
   (a) Show that placing a trap on each vertex suffices to capture the robber if the cops have perfect information.
   (b) Prove that placing a trap on each vertex suffices to capture the robber even if the cops have no information about the robber's whereabouts.
6. Prove that $K_{n+2,n+2}$ is $(2,0)$-win but not $(1,n)$-win.
7. Prove the first part of Theorem 8.6: if T is a tree and it has no subtree isomorphic to M, then $\mathrm{rb}(T) = 0$.
8. (a) Show that for a graph G, $c(G) \leq \mathrm{vs}(G)$.
   (b) Find infinitely many graphs such that $c(G) = \mathrm{vs}(G)$.
9. Show that the graph in Figure 8.7 is tandem-win but has no o-dominated vertices.
10. Prove Theorem 8.15.
11. Prove Theorem 8.16 (1).
12. Prove the lower bound in 8.18.
13. Let G be the graph formed from two copies of with $K_{1,n}$ one end-vertex in each copy being identified. Show $CC(G) = 2$. Further, prove that $CC(K_{m,n}) = 1$
14. Prove that $SO(K_m \otimes K_n) = \min\{m,n\} - 1$.
15. Derive that if $k \geq \mathrm{diam}(G) - 1$, then $c_k(G) = 1$.
16. Prove that for all $k \geq 1$, $c_k(G) \leq c_{k-1}(G)$.
17. Show that all graphs of order at most 6 have capture time at most three.
18. [**97**] Prove that the maximum capture time of a graph of order 7 is three.
19. Prove items (1) and (2) of Theorem 8.37.
20. [**26**] Prove that the robber can survive $n - 4$ moves in $H(n)$.
21. [**26**] Derive a formula for $\mathrm{capt}_2(C_n)$, where $n \geq 4$. (*Hint*: Consider cases modulo 4.)

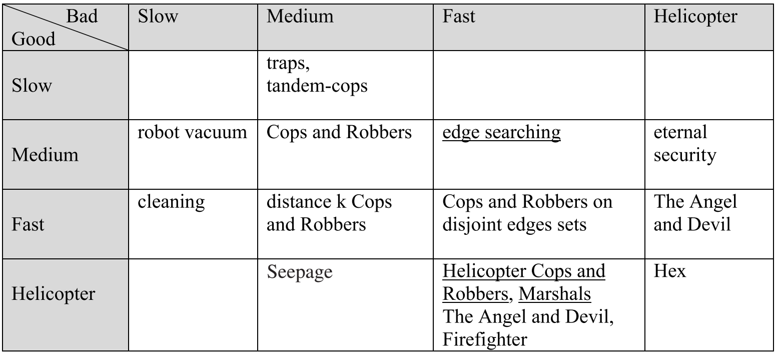
Chapter 9

Good Guys Versus Bad Guys

9.1. Introduction

From westerns to police dramas to comic book capers, there are good guys and bad guys. At the heart of all the games we consider in this book, there is a notion of a set of *good guys* (agents, cops, Greens, etc.) trying to stop, contain, or capture, a *bad guy* (intruder, robber, Sludge, etc.). We note that unlike the variants studied in Chapter 8, the players in most of the games in this chapter all have perfect information (except for edge searching, Helicopter Cops and Robbers, and Marshals).

The Cops and Robbers game uses the metaphor of catching a bad guy by some set of good guys, where both players have the same, finite speed. The metaphor can be modified in many ways, some of which were explored in Chapter 8. One way is to vary the *speeds* or the *methods of moving* of one or both players. The games Seepage (discussed in Section 9.3) and Helicopter Cops and Robbers (discussed in Section 9.5) fall in this category. For example, in Seepage, an intruder begins at the source of a directed acyclic graph and tries to reach a sink without being blocked by his opponent. To help summarize the games we consider (and some we do not!) we include a table with reference to their speed and information. The vertical labels refer to the speed of the good guys, while the horizontal ones refer to the speed of the bad guys. There are four kinds of movement we consider. There is *slow* or *restricted movement*, and *medium* or *average* movement. For example, in Cops and Robbers, the cops and robber move at medium speed. For *fast* movement, the player can move more quickly than in the usual Cops and Robbers game. In *helicopter* or *teleporting* movement, the player can move from a given vertex to any other in the graph (regardless of whether they are connected by a path). A game that is underlined gives one or both players imperfect information. For more on *eternal security* (also called *defending the Roman empire*), see [**182**].



Bad Good	Slow	Medium	Fast	Helicopter
Slow		traps, tandem-cops		
Medium	robot vacuum	Cops and Robbers	edge searching	eternal security
Fast	cleaning	distance k Cops and Robbers	Cops and Robbers on disjoint edges sets	The Angel and Devil
Helicopter		Seepage	Helicopter Cops and Robbers, Marshals The Angel and Devil, Firefighter	Hex

The reader will note that some cells of the table are blank. We invite you to consider new good guys versus bad guys games which could fit into these blank cells. What would their rules be? What parameters would you associate to such games, and what bounds on these can you find?

Another variant is to change the nature of the bad guy from intelligent to non-sentient. A good example of this is Firefighter, where the fire spreads aimlessly to all neighboring vertices; see Section 9.2. We could also change the nature of how the good guys win. For instance, instead of catching an intruder, we must instead clean a contaminated network as in cleaning in Section 9.6, robot vacuum in Section 9.6.3, or edge searching in Section 9.4. As we will see, these are not all the studied variations, nor are they mutually exclusive.

We close the chapter in Section 9.7 with a brief discussion of combinatorial games, and discuss the well-studied Angel and Devil game introduced by Conway [**63**].

9.2. Firefighter

The *Firefighter Problem*—introduced by Hartnell in 1995 at the 25th Manitoba Conference on Combinatorial Mathematics and Computing [**108**]—presents a simplified deterministic model of the spread of fire, diseases, and computer viruses. In *Firefighter*, vertices are either *burning* or not. There is one *firefighter* who is attempting to control the fire. Once a vertex is occupied by the firefighter, it can never burn in any subsequent round and is called *saved* or *protected*. The fire begins at some vertex in the first round, and the firefighter chooses some vertex to save. The firefighter can visit any non-burning vertex in a given round (for example, he can jump between two non-joined vertices from one round to the next), but he cannot protect a vertex on fire. The fire acts without intelligence and spreads to all non-protected neighbors. Once a vertex has been protected, its state cannot change; that is, it can never be on fire. The process stops when the fire can no longer spread.

Firefighter may be viewed as another variant of Cops and Robbers, although it is only a one-person game, with only the firefighter using any strategy. See Figure 9.1 for an example of Firefighter played on a tree (see also [**109**]). With the firefighter's choice of moves, the fire burns only two vertices (other moves will lead to more burned vertices).

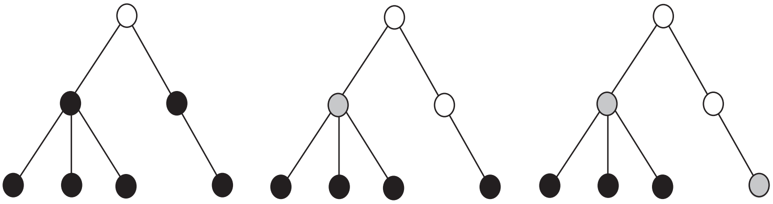


Figure 9.1. Firefighter played on a tree over two rounds, with white vertices burning and grey vertices protected.

Consider two extreme examples: a clique and a path. In a clique, the firefighter can save at most one vertex. In a path, suppose the fire breaks out at an end-vertex x. The firefighter saves the unique neighbor of x, and so saves the rest of the vertices. If a fire breaks out at a vertex of degree 2 y, then the firefighter saves one neighbor of y. The fire spreads to its unsaved neighbor where it is contained by the firefighter in the next round.

One goal of Firefighter may be to save the maximum possible number of vertices; see the survey [**76**] for a discussion of various desired outcomes of the game. For a graph G with vertex v, define $\mathrm{sn}(G, v)$ to be the maximum number of vertices in G the firefighter can save if the fire breaks out at v. For example, $\mathrm{sn}(K_n, v) = 1$, $\mathrm{sn}(C_n, v) = n - 2$, while

$$\mathrm{sn}(P_n, x) = \begin{cases} n-1 & \text{if } x \text{ is an end-vertex,} \\ n-2 & \text{else.} \end{cases} \tag{9.1}$$

If a fire breaks out in a hypercube, then the following theorem shows that the maximum number of saved vertices equals its dimension.

Theorem 9.1 ([**145**]). *For $n \geq 1$ and all vertices v, $\mathrm{sn}(Q_n, v) = n$.*

Proof. As Q_n is vertex-transitive, without loss of generality, we let the fire break out at v equaling the constant sequence of all zeros. The proof is by induction on the number of rounds t, with the inductive hypothesis that all vertices with at most t ones are either saved or burned. The proof of the theorem follows, since the firefighter can save at most t vertices in t rounds. The base case is immediate, and we consider the $(t+1)$th round. A vertex with $(t+1)$-many ones is joined to exactly $t+1$ vertices with t ones, as there $(t+1)$-many coordinates that can change from a one to a zero. By the induction hypothesis, at most t vertices with t ones are saved, so at least one such vertex x is burning. Hence, a non-saved vertex in round $t+1$ with $(t+1)$-many ones will burn if it is not saved in that round. □

Before continuing, it is important to point out that determining the number of vertices saved in Firefighter is a difficult computational problem. Consider the following graph decision problem.

FIREFIGHTER: Given a graph G and vertex v, is $\operatorname{sn}(G,v) \geq k$? That is, is there a finite sequence $u_1, u_2, \ldots, u_t$ of vertices of G such that if the fire breaks out at v, then each of the following items hold?

(1) Vertex u_i is neither burning nor saved in round i.

(2) At round t no non-saved vertex is adjacent to a burning vertex.

(3) At least k vertices are saved at the end of round t.

The problem FIREFIGHTER is in **NP**, since given a sequence

$$u_1, u_2, \ldots, u_t$$

of vertices, we can quickly check whether it meets items 1, 2, and 3 above. Finbow et al. [**77**] proved the following.

Theorem 9.2 ([**77**]). *FIREFIGHTER is **NP**-complete in trees with maximum degree* 3 *with the fire starting at a vertex of degree at most* 2.

Our main focus is on the expected percentage of vertices saved if the fire breaks out at a random vertex (where "random" means relative to the uniform distribution). For a graph G of order n, define the *surviving rate* of G, written $\rho(G)$, by

$$\rho(G) = \frac{1}{n^2} \sum_{v \in V(G)} \operatorname{sn}(G,v).$$

For example, since $\operatorname{sn}(K_n, v) = 1$ for all vertices v, we have that $\rho(K_n) = \frac{1}{n}$. The surviving rate was introduced by Cai and Wang, and studied in [**37**]. The following lemma is left as an exercise.

Lemma 9.3. (1) *For all graphs G of order n,*

$$\frac{1}{n} \leq \rho(G) < 1.$$

(2) *For paths, we have that*

$$\rho(P_n) = 1 - \frac{2}{n} + \frac{2}{n^2}.$$

We do not know the exact surviving rate for many graph families, even for Cartesian grids. It was proved in [**36**] that

$$\frac{5}{8} - O\left(\frac{1}{n^2}\right) \leq \rho(P_n \square P_n) \leq \frac{37}{48} - O\left(\frac{1}{n}\right).$$

It is even an open problem to determine the maximum number of vertices saved for a fire starting in the centre of the grid; for example, if the vertices on the grid are labeled (i, j) with $1 \leq i, j \leq n$, it is conjectured that

$$\frac{\operatorname{sn}\left(P_n \square P_n, \left\lceil \frac{n}{2} \right\rceil, \left(\left\lceil \frac{n}{2} \right\rceil\right)\right)}{n^2} = \frac{1}{4} + o(1).$$

It is conjectured that the actual value of $\rho(P_n \square P_n)$ is $\frac{5}{8} - O\left(\frac{1}{n^2}\right)$, although this remains open at the time of writing. Consider the following strategy, first given in [**157**], assuming the vertices of the grid are labeled (r, c), where $1 \leq r, c, \leq n$. When the fire breaks out at (r, c), $1 \leq r \leq c \leq n$, the firefighter saves the following vertices in order:

$$\begin{aligned}
&(r+1, c), (r+1, c+1), (r+2, c-1), (r+2, c+2),\\
&(r+3, c-2), (r+3, c+3), \ldots, (r+c, 1),\\
&(r+c, 2c), (r+c, 2c+1), \ldots, (r+c, n).
\end{aligned}$$

It is an exercise to show that using this strategy,

$$n(n-r) - (c-1)(n-c)$$

vertices are saved. It is conjectured that this is the optimal strategy for grids. This has been shown if $r = 1, 2$ (see [**145**]). We also note that Moeller and Wang [**157**] conjectured that for all vertices v

$$\lim_{n \to \infty} \frac{\operatorname{sn}(P_n \square P_n \square P_n, v)}{n^3} = 0,$$

and this was proven by Develin and Hartke [**67**].

For trees we have the following result.

Lemma 9.4 ([**145**]). *In an optimum strategy for the firefighter on a tree, the firefighter saves a vertex adjacent to the burning vertices in each round.*

Proof. Suppose that the firefighter saves a vertex u not adjacent to the burning vertices. As trees have unique paths connecting vertices, the strategy which protects a closer vertex of the path connecting u to a burning vertex saves more vertices. $\square$

Recent work of Cai et al. [**38**] establishes an asymptotically tight bound on the surviving rate of trees.

Theorem 9.5 ([**38**]). *For a tree T, $\rho(T) \geq 1 - \Theta\left(\frac{\log n}{n}\right)$. For the complete ternary tree (that is, every vertex is degree 1 or 3), we have equality.*

We note that the lower bound in Theorem 9.5 was generalized in Cai et al. [**38**] to outerplanar graphs. For sparse graphs, it is apparent that survival rates should be relatively large. Finbow, Wang, and Wang [**78**] showed that any graph G with $n \geq 2$ vertices and size at most $\left(\frac{4}{3} - \varepsilon\right) n$ has the property that $\rho(G) \geq \frac{6}{5}\varepsilon$, where $0 < \varepsilon < \frac{5}{24}$ is fixed. In [**178**], this was improved to show that graphs with size at most $\left(\frac{15}{11} - \varepsilon\right) n$ have surviving rate $\rho(G) \geq \frac{1}{60}\varepsilon$, where $0 < \varepsilon < \frac{1}{2}$ is fixed. Moreover, a construction of a random graph has been proposed to show that no further improvement is possible; that is, $\frac{15}{11}$ is the threshold.

We mention some results on firefighting in infinite grids. The firefighters can protect vertices to create *fire walls*: barriers through which the fire cannot escape. A fire is said to be *contained* if it is surrounded by fire walls and can no longer spread. See Figure 9.2.

For grids, it is interesting how many firefighters are needed to contain the fire; label this f_G. For Cartesian grids, $f_G = 2$ (see [**82, 157**]), while in strong grids (that is, the strong product of two infinite paths) $f_G = 4$; see [**150**]. The triangular grid (also called the isometric grid) is formed by tiling the plane regularly with equilateral triangles. It was shown in [**82, 149**] that $f_G = 3$. An open problem from [**148**] is whether one firefighter can contain a fire on the infinite hexagonal grid (that is, a tiling by equilateral hexagons).

9.2.1. Fighting intelligent fires. We consider a variant of firefighting, called *k-Firefighter* for a fixed positive integer k, where at each time-step, the fire chooses at most k edges to burn. This game is

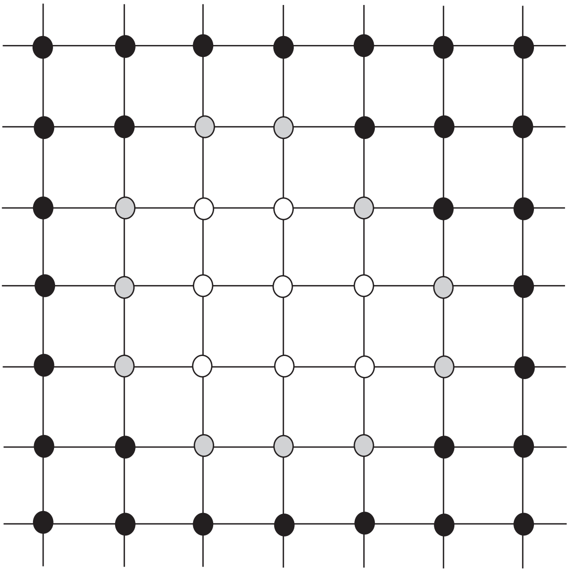


Figure 9.2. A fire wall in the infinite Cartesian grid. The grey vertices are protected, while white are burning.

referred to as the *k-Firefighter*, and it was first studied in [**30**]. Note that the fire now acts intelligently, unlike in Firefighter. Hence, we have a two-player game which is even more akin to Cops and Robbers. We note that the game of k-Firefighter was first suggested as a direction for future investigation in [**67**]. For the rest of the section, k is a fixed positive integer.

Assuming optimal play, for a vertex v in G, define $\mathrm{sn_k}(G, v)$ to be the number of vertices that can be saved if a fire breaks out at v in the game of k-Firefighter. For a finite graph G, define its *k-surviving rate* to be

$$\rho_k(G) = \frac{1}{n^2} \sum_{u \in V(G)} \mathrm{sn_k}(G, u).$$

For example, for a clique K_n,

$$\rho_k(K_n) = \frac{\lceil (n-1)/(k+1) \rceil}{n} \geq \frac{1}{k+1}\left(1 - \frac{1}{n}\right),$$

which is approximately $\frac{1}{k+1}$ for n large. For a path,

$$\rho_k(P_n) = \rho(P_n) = 1 - \frac{2}{n} + \frac{2}{n^2}.$$

As adding edges does not increase the k-surviving rate, it follows that cliques have the smallest surviving rates. Hence, for a graph with n vertices we have that

$$\rho_k(G) \geq \frac{\lceil (n-1)/(k+1) \rceil}{n} \geq \frac{1}{k+1}\left(1 - \frac{1}{n}\right). \tag{9.2}$$

We note the following upper bound for the k-surviving rate of a connected graph as a function of k and its order, whose proof is left as an exercise.

Theorem 9.6 ([**30**]). *For a connected graph G on n vertices,*

$$\begin{aligned} \rho_k(G) &\leq 1 - \frac{2}{n} + \frac{1}{n^2} + \frac{1}{n^2}\left\lceil \frac{n-1}{k+1} \right\rceil \\ &\leq 1 - \frac{1}{n}\left(2 - \frac{1}{k+1}\right) + O\Big(\frac{1}{n^2}\Big). \end{aligned} \tag{9.3}$$

Note that the bound in (9.3) is sharp as equality holds for a star on n vertices.

We now consider k-Firefighter played on random d-regular graphs with the uniform probability distribution. The probability space on d-regular graphs is denoted by $\mathcal{G}_{n,d}$. Recall from Chapter 6 that an event holds *asymptotically almost surely* or *a.a.s.* in $\mathcal{G}_{n,d}$ if it holds with probability tending to one for $n \to \infty$ with $d \geq 2$ fixed, with the proviso that n is even if d is odd. For more on random regular graphs, see the survey [**199**].

One might expect, for example, that a typical cubic graph can be well protected, especially in the case where k is small. However, as we will see in Theorem 9.7, random regular graphs are *flammable*, in the sense that the fire can a.a.s. burn a sizeable proportion of the graph. We now present an asymptotic upper bound for the k-surviving rate of random d-regular graphs for all values of d and k.

Theorem 9.7 ([**30**]). *Let $d \geq 3$, $k \geq 1$, and fix $\varepsilon > 0$. Let*

$$\lambda = 2\sqrt{d-1} + \varepsilon.$$

Then, for $G \in \mathcal{G}_{n,d}$ we obtain that a.a.s.

(9.4)

$$\rho_k(G) \leq \frac{(1+o(1))}{k+1}\left(1+\frac{\lambda}{d}\left(\sqrt{k}+\frac{d}{d-\lambda}\right)\right) = \frac{(1+O(d^{-1/2}))}{k+1}.$$

By (9.2) and (9.4), we have that $\rho(G,k) \to \frac{1}{k+1}$ as $d \to \infty$. Hence, for large values of d, a.a.s. random d-regular graphs have, in a certain sense, the smallest possible k-surviving rate.

9.3. Seepage

Like Intelligent Firefighting and Cops and Robbers, Seepage has intelligent opponents. Unlike Firefighting but like Cops and Robbers, the game of Seepage has a definite goal rather than a score. This good guy versus bad guy game was introduced in [**55**]. The motivating example was the 1973 eruption of the Eldfell volcano in Iceland. In order to protect the harbour, the inhabitants poured water on the lava in order to solidify it and thus, halt its progress. The mathematical model has two opponents, *Sludge* and *Greens*, a directed acyclic graph (or *dag*) with one source (that is, the top of the volcano) and many sinks (representing the lake). To simplify the diagrams, we omit orientations of directed edges, and assume all edges point from higher vertices to lower ones. See Figure 9.3. The players take turns, with the Sludge going first by contaminating the top vertex (source). On subsequent moves the Sludge contaminates a non-protected vertex that is adjacent (that is, downhill) to a contaminated vertex. The Greens, on their turn, choose some non-protected, non-contaminated vertex to protect. Once protected or contaminated, a vertex stays in that state to the end of the game. The Sludge wins if some sink is contaminated; the Greens win if they erect a cutset of vertices which separates the contaminated vertices from the sinks. The name "seepage" is used because unlike firefighting, the rate of contamination is slow.

Unless otherwise specified, we only consider games in which there is one Green; that is, only one vertex can be protected on each turn of the Greens. A *green-win* dag is one in which the Greens can win;

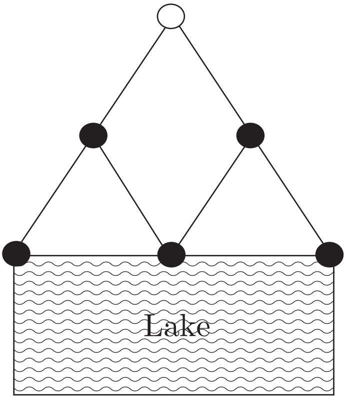


Figure 9.3. Seepage on a volcano overlooking a lake. The white vertex is the source where the contamination begins.

otherwise, it is a *sludge-win* dag. The dag in Figure 9.4 is sludge-win; see Exercise 8.

In a sludge-win graph, an *efficient* Sludge strategy pollutes only the vertices of a directed path, and we say Sludge wins *efficiently*. The following result is proven in [**55**].

Theorem 9.8 ([**55**]). *If G is a sludge-win dag, then Sludge has an efficient strategy.*

The following basic question deserves our attention.

Question: *Characterize the green-win dags.*

We do not know the full answer to this problem. We note that in [**55**] only green-win trees were characterized. Let T be a directed tree with root x, v a vertex of T, and T_v the subtree rooted at v; that is, the subtree downhill from v. Recall that the *out-degree* of a vertex v, written $d^+(v)$, is the number of edges emanating from v (that is, going downhill).

We are led to a characterization of green-win trees analogous to the cop-win characterization of Theorem 2.3.

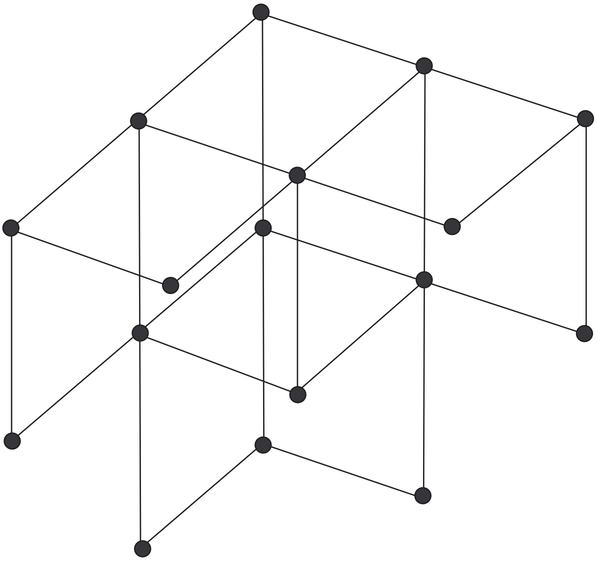


Figure 9.4. Seepage on a truncated Cartesian product of paths.

Lemma 9.9 ([**55**]). *If T is a tree with no vertex v where $d^+(v) = 1$, then T is sludge-win.*

Theorem 9.10 ([**55**]). *If $v \in V(T)$ and $d^+(v) = 1$, then $T - T_v$ is green-win if and only if T is green-win.*

In particular, if T is green-win, then there exists a vertex v with $d^+(v) = 1$; moreover, deleting this v and all the vertices downhill from it leaves a green-win tree. Part of the proof involves showing that, unlike in Figure 9.3, a good move for the greens is to always move just below a contaminated vertex. All the proofs can be extended to characterizing trees which are green-win when the greens can protect k vertices on each move.

The rest of [**55**] considers dags which are truncated Cartesian products, as in Figure 9.3. Let P_n be the directed path $(0, 1, \ldots, n)$. We consider the Cartesian product of copies of P_n rooted at the vertex $(0, 0, \ldots, 0)$. (The Cartesian product of dags is defined in an analogous way to the undirected case.) For a fixed $d > 0$, let (G, d) be the Cartesian product of the paths P_n which only includes the vertices at distance d or less from the root. For a dag G, let $gr(G)$ be the least positive integer r, so that if Green may protect r vertices of G on each move, the Green has a winning strategy in the Seepage game. One surprising result is the following.

Theorem 9.11 ([**55**]). *For $d \geq 3$, we have that*

$$gr(P_n^3, d) \leq 2.$$

The following result gives an upper bound for paths.

Theorem 9.12 ([**55**]). *For positive d,*

$$gr(P_n^k, d) \leq \min_{1 \leq j \leq k} \{\max\{j, gr(P_n^j, d - (k-j)n)\}\}.$$

In particular, we have the following.

Corollary 9.13 ([**55**]). *If $d \geq (k-j)n + 1$, then $gr(P_n^k, d) \leq j$.*

A specific case is considered in the next theorem.

Theorem 9.14 ([**55**]). *If $n > 8$ and $d > 8$, then (P_n^3, d) is green-win.*

These results suggest the following open problem.

Question: *For what d, k, and n is $gr(P_n^k, d) = 1$?*

9.4. Graph Searching

T.D. Parsons was approached by a group of spelunkers who had a problem: How would you search for a person lost, and possibly injured, in a network of caves? He wrote two papers on this topic [**173, 174**], and introduced the field of *graph searching.* The original model had issues of how long it took to search individual edges and that the lost person or intruder could hide along an edge; also, another feature was that the intruder was infinitely fast. Another interpretation that came later, is that a network could be contaminated by a noxious chemical or virulent biological agent. The issue of dawdling or stopping along an edge becomes irrelevant. If a cleaned edge were re-exposed to the agent, then it and all the other cleaned edges to which it was connected would have to be considered as recontaminated, giving credence to the "infinitely fast" aspect. The subject of graph searching has a large literature. In 2004 Alspach [**7**] gave a brief survey, and in 2008 Fomin and Thilikos [**87**] compiled a bibliography with 172 entries.

There are two basic problems: only the vertices must be searched, or only the edges. In the former one, we may think of the edges as doors and the vertices as rooms; in the second the edges are corridors. The basic models may have the searchers moving in one of two ways: moving from vertex-to-vertex along an edge or jumping (or teleporting) immediately from one vertex to any other vertex. Other constraints may be placed on the searchers. A vertex or an edge can be searched in different ways: traversing the edge or visiting the vertex, being adjacent, or being incident to the vertex the searcher occupies. In all cases, once cleaned, an edge stays clean unless there is a searcher-free path from one of its endpoints to a contaminated edge or vertex. In Figure 9.5, the searchers traverse edges. In Figure 9.5 A, the single searcher on a cannot clean the horizontal edge without recontaminating the whole network. In Figure 9.5 B, two searchers can clean the two contaminated edges incident with a but that is as far as they can go without recontaminating the network. Because of the original motivating example and the methods used, graphs are not restricted to being simple but may have loops or multiple edges.

In any model of graph searching, the most fundamental question is: *What is the least number of searchers needed to clean the graph*? If G is a finite graph, then placing a searcher on every vertex, and possibly one more to clean all of the edges, suffices to clean the graph. Hence, it is evident that the number of required searchers is finite.

Suppose the edges of G must be cleaned. In the earlier literature, this was referred to as *sweeping a graph* but the term has been replaced by *edge searching.* Let $s(G)$ denote the fewest number of searchers required if the edges must be traversed to be cleaned, usually called the *edge search number*; $ls(G)$ if an edge can only be cleaned by having searchers at either end of the edge ("l" for *laser* or *line-of-sight*); and $xs(G)$ if both methods are employed. Alspach [**7**] shows the following.

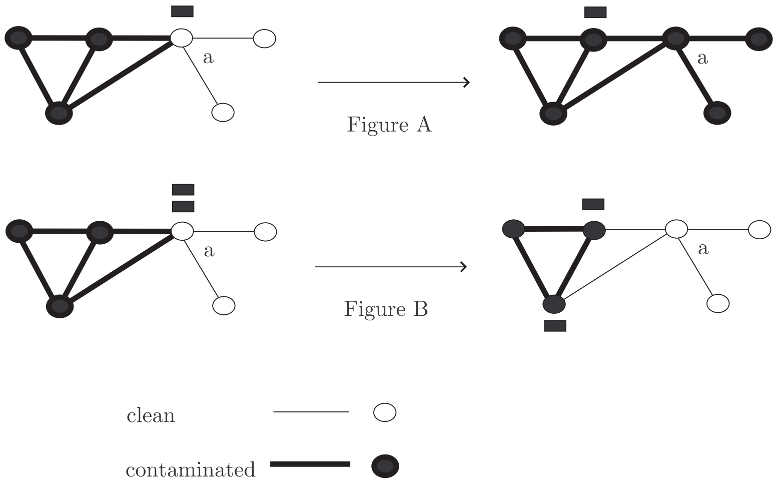


Figure 9.5. Two instances of graph searching.

Theorem 9.15. *If G is a graph, then the following inequalities hold:*

(1) $s(G) - 1 \leq ls(G) \leq s(G) + 1$.

(2) $s(G) - 1 \leq xs(G) \leq s(G)$.

(3) $ls(G) - 1 \leq xs(G) \leq ls(G)$.

It was an early conjecture that a graph can be edge-searched with the minimum number of searchers without allowing any edges to be recontaminated. Lapaugh [**135**] showed this is true and changed the status of the problem from being NP-hard to NP-complete. "No recontamination" also allows the idea of employing strategies that at the end of each move have reduced the contaminated area, often called the *intruder territory*. We introduced this back in Section 4.4 in planar graphs where it was called the *robber territory*.

Recall that the *vertex separation number*, or *pathwidth*, of a graph, $vs(G)$ was introduced in Chapter 8. Ellis et al. [**71**] showed that

$$vs(G) \leq s(G) \leq vs(G) + 2.$$

It follows that for any positive integer k there is a tree which requires at least k searchers to clean it; this is in direct contrast to all finite

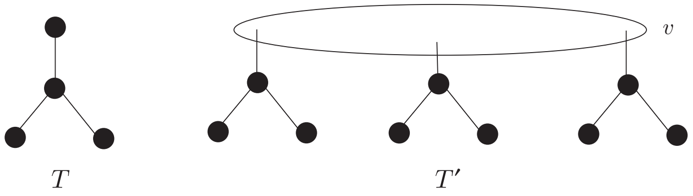


Figure 9.6. The tree T requires two searchers, while T' requires three searchers.

trees being cop-win. There is a nice recursive construction to give such a tree for each k.

Theorem 9.16 ([**173**]). *Let T_1, T_2, and T_3 be vertex-disjoint trees each having at least one edge, and let v_i be a vertex of degree one in T_i, $i = 1, 2, 3$. Let T' be the tree obtained by identifying the vertices v_1, v_2, v_3 as a single vertex v. If $s(T_i) = k$, with $i = 1, 2, 3$, then*

$$s(T) = k + 1.$$

Figure 9.6 shows the construction of a tree requiring three searchers. We remark also that Megiddo et al. [**147**] showed that the edge search number of a tree can be computed in linear time, but that in general, determining whether $s(G) \leq k$ for a given k is **NP**-complete. In the same paper, it is shown that if G has no cut vertices, then $s(G) \leq 3$ if and only if G is a member of a class of outerplanar graphs, and there is a linear time algorithm recognizing such graphs.

Alspach et al. [**8**] give lower bounds for the edge search number. Recall that for a graph G, $\delta(G)$ is the minimum degree of G, and $\omega(G)$ is the size of the largest clique.

Theorem 9.17 ([**8**]). *If G is a connected graph, then the following inequalities hold.*

(1) *If $s(G) \geq \delta(G)$ and $\delta(G) \geq 3$, then $s(G) \geq \delta(G) + 1$.*

(2) *If $\omega(G) \geq 4$, then $s(G) \geq \omega(G)$.*

(3) *If H is a minor of G, then $s(G) \geq s(H)$.*

Other variants have the searchers constrained in some way. For example the cleaned area has to be a connected subgraph. The issue of minimizing the "cost" of the searchers has also been considered.

9.5. Helicopter Cops and Robbers and Marshals

The vertex separation number, introduced in Chapter 8, was shown to be a searching parameter in the last section. By varying the rules of the game, other useful graph parameters can be defined. The main and next concept after vertex separation number or pathwidth to be shown to be equivalent to a searching game was *treewidth* in [**188**].

In a tree decomposition, each vertex of the graph is represented by a subtree, such that vertices are adjacent only when the corresponding subtrees intersect. Formally, given a graph $G = (V, E)$, a *tree decomposition* is a pair (X, T), where $X = \{X_1, \ldots, X_n\}$ is a family of subsets of V, and T is a tree whose vertices are the subsets X_i, satisfying the following three properties:

(1) $V = \bigcup_{i=1}^{n} X_i$. That is, each graph vertex is associated with at least one tree vertex.

(2) For every edge (v, w) in the graph, there is a subset X_i that contains both v and w. That is, vertices are adjacent in G only when the corresponding subtrees have a vertex in common.

(3) If X_i, X_j, and X_k are vertices and X_k is on the path from X_i to X_j, then $X_i \cap X_j \subseteq X_k$.

The *width* of a tree decomposition is the size of its largest set X_i minus one. The *treewidth* of a graph G, written $tw(G)$, is the minimum width among all possible tree decompositions of G. In Figure 9.7 we have that $|X_1| = |X_3| = 3$, $|X_4| = 2$ and $|X_2| = 4$ so the treewidth of G is at most $4 - 1 = 3$.

The rules for *Helicopter Cops and Robbers* are as follows: the cops choose vertices, the robber then chooses a vertex; the positions of both players are known to each other. The moves are almost simultaneous: the cops announce which ones will move and are "transported by helicopter" to the new positions; that is, they are not on the graph

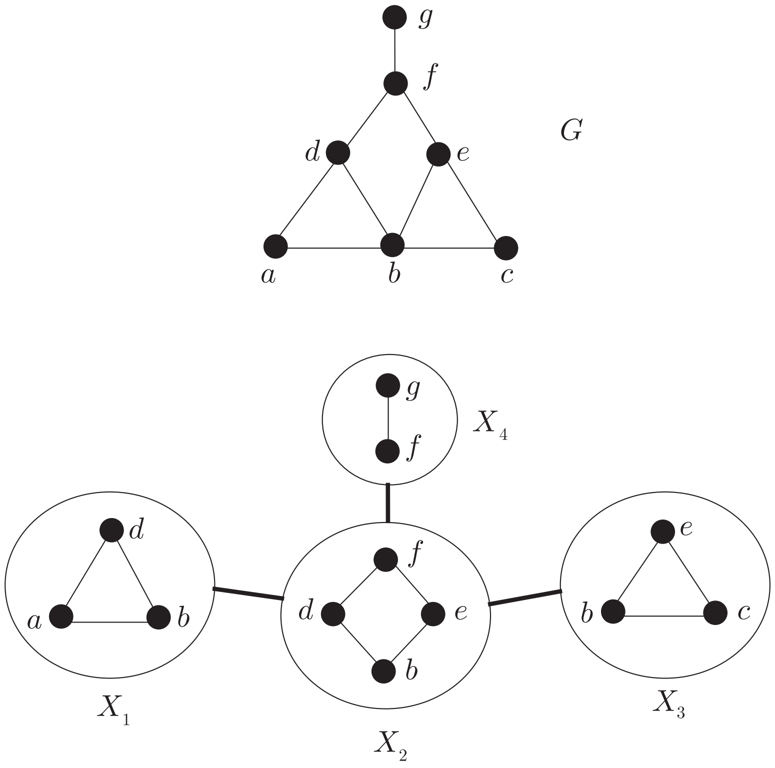


Figure 9.7. A tree decomposition of G.

for a period of time. Note that a cop can move to any vertex, not only its neighbors. During this time, the robber can move from his present position to any vertex that is reachable by a path that does not go through a vertex occupied by a cop still on the graph. In particular, this also means that he could remain on the same vertex. The game is over if a cop occupies the same vertex as the robber. Seymour and Thomas [**188**] gave a characterization of treewidth in terms of this game.

Theorem 9.18 ([**188**])**.** *A graph G has treewidth at most k if $k+1$ cops can capture the robber in Helicopter Cops and Robbers.*

The width and other definitions become technical so we do not go into detail in this abbreviated survey. A more powerful cop has been defined, a *marshal*, who occupies a subset of vertices or hyperedges which are defined as part of a hypergraph. Adler [**1**] considers the game of Marshals and a visible robber; in [**117**], the robber is invisible to the marshals but is *inert*: he only moves if a marshal will land on the same vertex.

9.6. Cleaning

In the *cleaning model*, imagine a network of water pipes that periodically have to be cleaned, by *brushes*, of a contaminant that regenerates, say algae or bio-film, but the contaminant grows in a time comparable with the time needed to clean the network. The regrowth is slowed if during the cleaning process, the cleaned edges are not exposed to water from "dirty" pipes. The cleaning model was suggested by the real-life situation of cleaning zebra mussels from the water pipes in a nuclear power plant on the Great Lakes [**114, 128**], and was first presented at the 2005 East Coast Combinatorics Conference by the second author of this book. Since then several papers have appeared; see [**5, 95, 96, 151, 153, 154, 168**]. A related cleaning model called the *robot vacuum*, will be dealt with in Section 9.6.3. When the cleaning process is completed, the contamination will come back and the network recleaned. This repetition is an integral part of the problem statement unlike other good and bad guy games. Moreover, in a real-life situation it would be efficient if the brushes did not have to reposition themselves, but could start the next cleaning cycle from their final position from the previous cycle, so this is also added to the problem specification.

The cleaning is accomplished by brushes assigned to some vertices. Vertices are either *dirty* or *clean*. An edge is *dirty* if both its endpoints are dirty. Initially, all vertices and edges are dirty. A vertex is *primed* if it has at least as many brushes as incident dirty edges. When a primed vertex is fired, it sends a brush down each dirty edge which is then added to the brushes at each of the adjacent vertices. Once a brush has traversed an edge, that edge has been *cleaned*. A graph G has been *cleaned* once every edge of G has been cleaned.

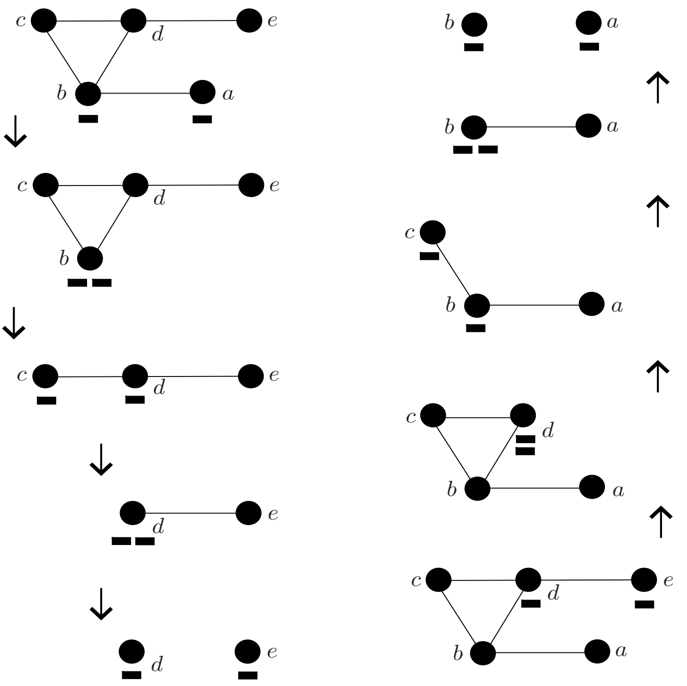


Figure 9.8. Cleaning a graph twice in succession.

Since a clean edge never takes part in the cleaning process again, we may as well delete it. Figure 9.8 illustrates the cleaning of a graph with two brushes, not once but twice.

At any stage during the cleaning, there may be more than one primed vertex, and here the model splits into two cases. In *sequential cleaning*, any primed vertex is chosen to fire, and if after firing there are still other primed vertices, one of them is chosen to fire, and so on. In *parallel cleaning* all primed vertices fire at the same time. Let a *good configuration* be a placement of brushes on a graph G that, after successively firing primed vertices, will clean G.

The main questions for both sequential and parallel are the following:

(1) What good configurations will result in a final configuration that will also be a good configuration for the original graph?

(2) Among such configurations, which has the fewest number of brushes?

We will always assume the that the original graph is connected. It then follows that if a vertex becomes isolated, then it will have at least one brush.

9.6.1. Sequential Cleaning. The element of choice in the definition of Sequential Cleaning might be worrying, but fortunately, the choice is irrelevant. Before getting into the details, we need to be able to describe the inner workings of the cleaning process. Given a graph G, let $b_i(x)$ be the number of brushes on vertex x at time i, where $b_0(x)$ is the number of brushes initially assigned to x. Let $d_i(x)$ be the number of dirty edges incident with x at time i. A sequence of vertices, $a_1, a_2, \ldots, a_p$ in graph G is a *cleaning sequence* if a_1 through a_p can be cleaned in that order and that, after a_p, no other vertex has enough brushes to be cleaned. A cleaning process is an initial placement of brushes together with a cleaning sequence. More formally, a *cleaning process* starts with a configuration of brushes

$$\omega(G) = \{b_0(x) : x \in V(G)\}$$

and is a sequence of vertices $a_1, a_2, \ldots, a_p$ such that a_i is primed in $V(G) \setminus \{a_1, a_2, \ldots, a_{i-1}\}$ and is the next vertex to be fired. Note that

(1) $b_{i+1}(x) = b_i(x)$ and $d_{i+1}(x) = d_i(x)$ if x has already been fired or $x \notin N[a_i]$;

(2) $b_{i+1}(x) = b_i(x) + 1$ and $d_{i+1}(x) = d_i(x) - 1$ if $x \in N(a_i)$; and

(3) $b_{i+1}(x) = b_i(x) - d_i(x)$ and $d_{i+1}(x) = 0$ if $x = a_i$.

A cleaning process depends upon the graph and initial placement of brushes, but it will cause no confusion if we drop these and denote the process by ω.

When a cleaning sequence, say ω, is being considered and needs to be specified, we will insert it as a superscript as in $b_i^{\omega}(x)$ and $d_i^{\omega}(x)$.

A *good configuration*, is one such that there is a cleaning sequence that returns an empty set of dirty edges.

The *brush number* of G, written $b(G)$, is the minimum number of brushes needed to clean G. Similarly, $b_\omega(G)$ is defined as the minimum number of brushes needed to clean G using the cleaning sequence ω. It is evident that for every cleaning sequence ω, $b_\omega(G) \geq b(G)$ and

$$b(G) = \min_\omega b_\omega(G).$$

(The last relation can be used as an alternative definition of $b(G)$.) In general, it is difficult to compute $b(G)$ (it is in fact **NP**-complete [**96**]), but $b_\omega(G)$ can be easily computed. To see this, choose a sequence $(a_1, a_2, \ldots, a_n)$, of the vertices and regard it as a linear layout, starting with a_1 on the left. After $a_1, a_2, \ldots, a_{i-1}$ have been fired, then in order to fire a_i, $b_0(a_i)$ needs to make up the discrepancy between the number of edges to $a_{i+1}, a_{i+2}, \ldots, a_n$ (the number of dirty edges at a_i at time i) and the number of brushes it has received from $a_1, a_2, \ldots, a_{i-1}$. Let $l(v)$ and $r(v)$ be the number of edges going to the left and to the right, respectively, from v in the linear layout. Then we have that

$$b_0(v) = \max\{r(v) - l(v), 0\}. \tag{9.5}$$

For a given linear layout ω, this gives

$$b_\omega(G) = \sum_{v \in V(G)} \max\{r(v) - l(v), 0\}. \tag{9.6}$$

This is not the way it is defined in [**153**], but it is an equivalent formulation and it makes the connection clear to the "imbalance problem" which is about to be introduced.

In [**153**], the first paper on the subject, the authors first eliminate the "choices" in the cleaning sequence.

Theorem 9.19 ([**153**]). *Given a graph G and the initial configuration of brushes b_0, the cleaning algorithm returns a unique final set of dirty vertices.*

Proof. Let $\alpha = (a_1, a_2, \ldots, a_p)$ and $\omega = (w_1, w_2, \ldots, w_q)$ be two cleaning sequences with initial configuration b_0; that is, in both cases, the ith vertex is primed after the first $i-1$ have been fired. We assume that in both cases, at the end there are no further primed but unfired vertices. Note that it is enough to prove that

$$\{a_1, a_2, \ldots, a_p\} = \{w_1, w_2, \ldots, w_q\}.$$

Suppose that there is a vertex in ω which is not in α. Let w_i, $1 \leq i \leq q$, be the first such vertex. Consider now the configuration at the final step p of α and ω at step $i-1$. In ω at step i, $b_i^\omega(w_i) \geq d_i^\omega(w_i)$. Since α contains vertices $w_1, w_2, \ldots, w_{i-1}$, then by the time α has finished, w_i has received brushes from the same subset of $w_1, w_2, \ldots, w_{i-1}$ as it did in ω and possibly more. Also, for the same reasons, the number of adjacent dirty edges, $d_p^\alpha(w_i)$ is no greater than $d_i^\omega(w_i)$. In summary,

$$d_p^\alpha(w_i) \leq d_i^\omega(w_i) \leq b_i^\omega(w_i) \leq b_p^\alpha(w_i),$$

but this is a contradiction since w_i is primed in α but unfired. A symmetric argument can be used to show that ω contains all vertices of α. □

In [**153**], it is shown that the requirement that the final configuration be a good configuration for the next cleaning process is automatic, and that a cleaning sequence is the reverse of the original.

Theorem 9.20 ([**153**]). *Given an initial configuration b_0, suppose G can be cleaned yielding the final configuration b_n, where $n = |V(G)|$. Then, with the initial configuration $b_0' = b_n$, G can be cleaned yielding the final configuration $b_n' = b_0$.*

The original problem called for the edges to be cleaned. By insisting that the vertices also be cleaned, even if all the incident edges have been cleaned by incoming brushes, it easy to show that the reverse sequence works. If the sequence had stopped when the edges had been cleaned, then it would be difficult to identify the new cleaning sequence.

Consider the following graph decision problem.

CLEAN: Given a graph G and an integer $k \geq 0$, is $b(G) \leq k$?

In [**96**], they give the following hardness result.

Theorem 9.21 ([**96**]). *CLEAN is **NP**-complete and remains **NP**-complete for bipartite graphs of maximum degree* 6, *planar graphs of maximum degree* 4, *and* 5*-regular graphs.*

Given a graph $G = (V, E)$ and a linear layout $\pi = (a_1, a_2, \ldots, a_n)$ of G, the *imbalance* of a vertex $v \in V$ with respect to π is $\phi_\pi(v) = |r(v) - l(v)|$, and the *total imbalance* $\mathrm{Imb}_\pi(G)$ of an ordering is the sum of the imbalance of each vertex: $\mathrm{Imb}_\pi(G) = \sum_{v \in V} \phi_\pi(v)$. Let $\mathrm{Imb}(G)$ denote the *minimum total imbalance* taken over all possible linear layouts. A vertex v is said to be *imbalanced in a linear layout* π if $\mathrm{Imb}_\pi(v) > 0$.

Consider the following graph decision problem.

BALANCED VERTEX ORDERING: Given an integer $k \geq 0$, does the graph G have a vertex-ordering with total imbalance at most k?

It is known that the BALANCED VERTEX ORDERING is **NP**-complete; see [**127**]. By tying the two problems together, the hardness result for CLEAN follows.

Theorem 9.22 ([**96**]). *For a graph G and a vertex-ordering π of G,*

$$\mathrm{Imb}_\pi(G) = 2b_\pi(G).$$

Proof. By the definition of Imb_π, ϕ_π, b_π, and equations (9.5) and (9.6), we have that

$$\begin{aligned}
\mathrm{Imb}_\pi(G) &= \sum_{v \in V(G)} \phi_\pi(v) \\
&= \sum_{v \in V(G)} \big| |N_\pi^+(v)| - |N_\pi^-(v)| \big| \\
&= \sum_{v \in V(G)} \max\{|N_\pi^+(v)| - |N_\pi^-(v)|, 0\} \\
&\quad + \sum_{v \in V(G)} \max\{|N_\pi^-(v)| - |N_\pi^+(v)|, 0\} \\
&= \sum_{v \in V(G)} b_0(v) + \sum_{v \in V} b_n(v) \\
&= 2b_\pi(G).
\end{aligned}$$

□

Since the CLEAN is **NP**-complete, an important issue is bounds. Given any good configuration of brushes, a *brush path* is the set of edges that a particular brush takes during the cleaning sequence. It is easy to see that every odd vertex has a brush either at the beginning or the end of the sequence. Let $d_o(G)$ be the number of vertices of odd degree in G.

Theorem 9.23 ([**153**]). *For a graph G, $b(G) \geq \frac{d_o(G)}{2}$.*

In practice, the most useful lower bound appears to be given by the *Boundary Edge Theorem*, stated below. Intuitively, given a graph G, let S be an induced subgraph. If the vertices of S are cleaned first, then each edges between S and $G - S$ is cleaned by a different brush since no brush that leaves S can re-enter.

Theorem 9.24 (Boundary Edge Theorem, [**153**]). *Let G be a graph, and let*

$$b_k = \min_{S \subseteq V, |S|=k} \left\{ \sum_{v \in S} \deg(v) - 2|E(G[S])| \right\}. \tag{9.7}$$

Then $b(G) \geq b_k$.

Proof. Let $\{a_1, a_2, \ldots, a_n\}$ be a linear layout that realizes $b(G)$, and let $S = \{a_1, a_2, \ldots, a_k\}$. Then equation (9.7) gives

$$\begin{aligned} b(G) &= \sum_{i=1}^{n} \max\{r(a_i) - l(a_i), 0\} \\ &\geq \sum_{i=1}^{k} \max\{r(a_i) - l(a_i), 0\}. \end{aligned}$$

For any vertex v in a linear layout, $\deg(v) = r(v) + l(v)$; thus,

$$r(a_i) - l(a_i) = \deg(a_i) - 2l(a_i).$$

In this linear layout, each edge of $G[S]$ appears exactly once as a left edge of some a_i, $i = 1, \ldots, k$; hence, $\sum_{i=1}^{k} l(a_i) = |E(G[S])|$.

Therefore, we have that

$$\begin{aligned} b(G) &\geq \sum_{i=1}^{k}(r(a_i) - l(a_i)) \\ &= \sum_{i=1}^{k}(\deg_G(a_i) - 2l(a_i)) \\ &= \left(\sum_{i=1}^{k}\deg_G(a_i)\right) - 2|E(G[S])| \geq b_k. \end{aligned} \qquad \square$$

This bound may be a little difficult to calculate but this result does give an easier-to-calculate lower bound.

Corollary 9.25 ([**153**]). *For a graph G with girth $g \leq \infty$,*

$$b(G) \geq (\delta(G) - 2)g.$$

Upper bounds are harder to find. Alon, Prałat, and Wormald [**5**] have an upper bound, slightly better than that in [**151**], which is derived by exploiting random permutations.

Theorem 9.26 ([**5**]). *If G is a graph with v vertices and e edges, then*

$$b(G) \leq \frac{e}{2} + \frac{v}{4} - \frac{1}{4}\left(\sum_{\substack{v\in V(G),\\ \deg(v) \text{ is odd}}} \frac{1}{\deg(v)+1}\right).$$

For Cartesian products, there is a known reasonable upper bound.

Theorem 9.27 ([**153**]). *If G and H are graphs, then*

$$b(G\square H) \leq |V(H)|b(G) + |V(G)|b(H).$$

However, the proof is based on cleaning one copy of, say G, at a time. The paper contains an example where this approach is not optimal for the product.

The brush numbers for some familiar families of graphs are known or have good approximations, as summarized in the following theorem. Recall that $d_o(G)$ is the number of odd-degree vertices.

Theorem 9.28 ([**153**]). (1) *For any tree T, $b(T) = \frac{d_0(T)}{2}$.*

(2) *If n is even, then $b(K_n) = \frac{n^2}{4}$; otherwise, $b(K_n) = \frac{n^2-1}{4}$.*

(3) *For the complete multipartite graph G with m color classes each of cardinality n, we have that*

$$b(G) = \frac{m^2n^2}{4} + O(mn^2).$$

(4) *For some constants c_1, c_2, we have that*

$$c_1 2^n \leq b(Q_n) \leq c_2 2^n.$$

(5) *For $m, n > 1$, $b(P_m \Box P_n) = m + n - 2$.*

Cleaning processes involve deletion of vertices (and the deletion of incident edges). On random regular graphs, they are ideally set up to use the differential equation method; see [**200**]. Randomly cleaning a d-regular graph (that is, choose a random linear order) means that initially, many vertices of degree d will be cleaned first. For example, if $d = 2$ or $d = 3$, then there will be a second phase where vertices of lower degree are now primed and can be cleaned automatically (see [**154**]). An issue is the number of connected components that the graph possesses. The analysis becomes more complicated for larger d. The algorithm is called *degree-greedy* because the vertex being cleaned is chosen from those with the lowest degree.

Theorem 9.29 ([**5**]). (1) *The brush number of a random d-regular graph is a.a.s.*

$$\frac{n}{4}(d + o(d)).$$

(2) *If u_d is the total number of brushes needed to clean the random d-regular graph using the degree-greedy algorithm, then*

$$\lim_{d\to\infty} \frac{u_d}{dn} = \frac{1}{4}.$$

In particular, for large d, the degree-greedy algorithm a.a.s. achieves the optimal number of brushes up to a lower order term.

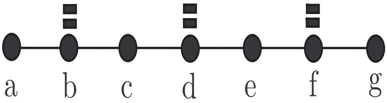


Figure 9.9. An irredundant configuration for P_7 using $B(P_7) = 6$ brooms.

Cleaning with brushes is a minimization problem. In [**153**], the maximization version is introduced under the heading *cleaning with brooms.* Brooms are big brushes, but we will still refer to them as brushes! A configuration is *irredundant* if there is a linear layout that has every broom cleaning at least one edge. In Figure 9.9 if we use the linear layout (a, c, d, b, e, f, g), then every brush gets used once. The *broom number* of a graph G, denoted $B(G)$, is the largest number of brushes in an irredundant good configuration. Since each brush cleans at least one edge, we have that $B(G) \leq |E(G)|$.

Theorem 9.30 ([**155**]). *For a graph G, $B(G) = |E(G)|$ if and only if G is bipartite.*

Theorem 9.26 is based on an argument about averages so it is of no surprise that it also provides a lower bound for $B(G)$.

Theorem 9.31 ([**155**]). *If G is a graph with v vertices and e edges, then*

$$B(G) \geq \frac{e}{2} + \frac{v-1}{4} - \frac{1}{4}\left(\sum_{\substack{v \in V(G),\\ \deg(v) \textit{ is even}}} \frac{1}{\deg(v)+1}\right).$$

An intriguing result is the following.

Theorem 9.32 ([**155**]). *Fix any integer $n \geq 1$. Then for each $k = 0, 1, \ldots, \lfloor n^2/4 \rfloor$, there exist graphs G and G' on n vertices with $B(G) = b(G') = k$. No other value can be obtained.*

Just like brushes, the broom cleaning problem on random regular graphs is approachable by similar methods used for the brush number.

Theorem 9.33 ([**176**]). *For fixed large d, the broom number of a random d-regular graph on n vertices is a.a.s. $\frac{n}{4}(d + \Theta(\sqrt{d}))$.*

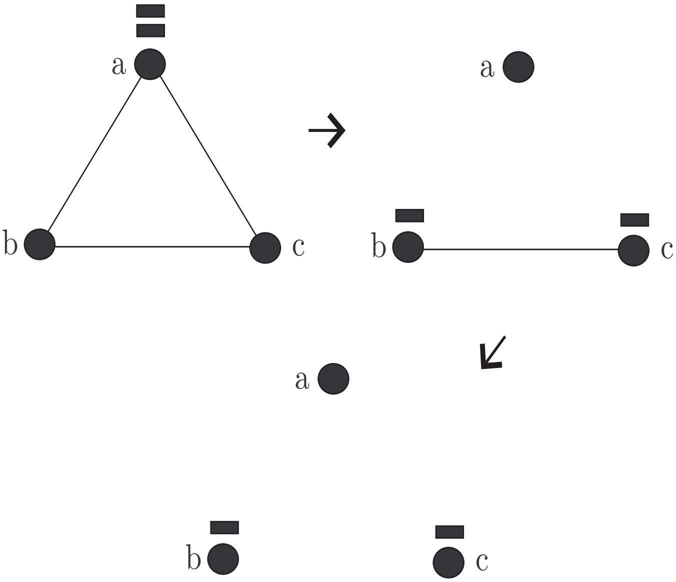


Figure 9.10. Parallel cleaning a triangle.

9.6.2. Parallel Cleaning. Parallel cleaning, introduced in [**95**], is more complicated than ordinary cleaning. For example, in Figure 9.10 two brushes at a are enough to clean the graph, but at the second step both b and c are primed and so both fire, swapping brushes. The final configuration is not a good configuration! It is possible for a good configuration to cause the graph to be cleaned once, twice, even thrice but no more. An immediate question (to which is there is no known answer) is the following: For a graph G, how many times must a good configuration clean G for the cleaning process to go on forever?

Gaspers et al. [**95**] differentiate between the *parallel brush number*, written $bp(G)$, the least number of brushes required to clean G with parallel cleaning, and the *continual parallel brush number*, written $\mathrm{cbp}(G)$, the least number of brushes required so that G is continually cleaned. In [**153**], it was shown that

$$b(G) = bp(G).$$

One can focus on the number of steps required to clean G. If the contaminant grows quickly enough, then every edge may have to be cleaned at every stage. This implies that in the initial and final configurations, each edge must be incident with a primed vertex. The *continual parallel one-step brush number*, written $\mathrm{cbp}_1(G)$, is the minimum number of brushes required to clean every edge on each firing.

Theorem 9.34 ([**95**]). *If G is a graph, then the following equalities hold:*

$$\mathrm{cbp}_1(G) = \begin{cases} |E(G)| & \textit{if } G \textit{ is bipartite,} \\ 2|E(G)| & \textit{otherwise.} \end{cases}$$

For $\mathrm{cbp}(G)$, very little else is known apart from its values on some classes of graphs.

Theorem 9.35 ([**95**]). (1) *For $n \geq 2$,*

$$\mathrm{cbp}(C_n) = \begin{cases} 2 & \textit{if } n \textit{ is even,} \\ 3 & \textit{if } n = 3, \\ 4 & \textit{otherwise.} \end{cases}$$

(2) *For any tree T,* $\mathrm{cbp}(T) = b(T) = bp(T)$.

(3) *For all integers $m, n > 0$,* $\mathrm{cbp}(K_{m,n}) = \lceil mn/2 \rceil$.

(4) *For K_n,*

$$5/16n^2 + O(n) \leq \mathrm{cbp}(K_n) \leq 4/9n^2 + O(n).$$

The proof of item (2) for trees requires the following result.

Lemma 9.36 ([**95**]). *Consider a parallel cleaning sequence cleaning a tree T using $b(T) = bp(T)$ brushes. The set of vertices cleaned at each time-step is an independent set.*

There are many unanswered questions for parallel cleaning; one goal is to find good upper and lower bounds. The authors of [**95**] highlight some problems which should be feasible. For instance, if G is bipartite, then can the difference between $\mathrm{cbp}(G)$ and $b(G)$ be arbitrarily large? Another problem is to find the exact value of $\mathrm{cbp}(K_n)$.

9.6.3. Robot Vacuums. In this model, introduced by Messinger and Nowakowski in [**152**], the contamination does re-occur but at a pace much slower than the time needed to clean the graph. Graphs in this section are allowed both loops and multiple edges. In this case, one could imagine a large building with many corridors. A robot cleans these corridors in a greedy fashion, so that the next corridor cleaned is always the dirtiest to which it is adjacent. (There could be a sensor at each end of the corridor which indicates the time it was last cleaned, thereby relieving the robot from making any real decisions.) Initially, the edges of the graph will be assigned different weights indicating the last time each was cleaned. The robot's initial position is any given vertex. We will also assume that each edge takes the same (unit) amount of time to clean. Weights are taken as negative integers, with the initial weights for the graph G being $-1, -2, \ldots, -|E(G)|$. See Figure 9.11 for two different weightings on the same graph. We leave the reader to show (Exercise 22) that for a connected graph, one robot will eventually clean the graph.

There is a remarkable amount of "self-stabilization" in the robot vacuum process. Let $S(G)$ be the maximum number of steps needed for the robot to clean every edge once. Let $T(G)$ be the time needed for all the edges to be cleaned the first time. The first time the graph

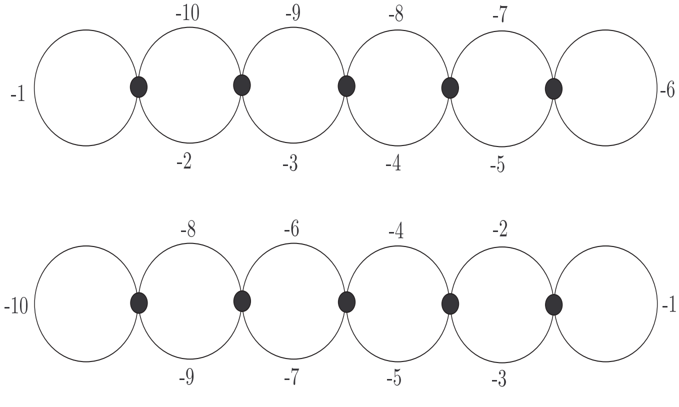


Figure 9.11. In the upper edge-weighting, the robot cleans the graph after 10 steps, while 30 steps are needed in the lower edge-weighting.

is cleaned, the robot spends much time wandering around but after that it becomes rather efficient. An *Eulerian* (*semi-Eulerian*) graph is one which contains a closed (not necessarily closed) walk in which all the edges are distinct, and which visits each edge exactly once. As is well known, a connected graph is Eulerian if and only if every vertex has even degree, while semi-Eulerian graphs are precisely those with at most two vertices of odd degree. (See Exercise 23.)

Theorem 9.37 ([**152**]). *If G is an Eulerian graph, then after the first time every edge has been cleaned* (*that is, after time $T(G)$*) *we have that $S(G) = |E(G)|$, and the initial and final locations of the robot are the same. If G is a semi-Eulerian graph, then $S(G) = |E(G)|$, and the initial and final locations are the two vertices of odd degree.*

How large can $T(G)$ be? Li and Vetta [**139**] give an example that takes exponential time. The *cover time*, written $c(G)$, of a connected graph G is the maximum number of steps, over all initial edge weightings w and all possible starting vertices s, until each edge has been visited. Let C_e be the worst-case cover time over all graphs containing exactly e edges.

Theorem 9.38 ([**139**]). *There exists a constant $d > 0$ such that, for all e,*

$$C_e \geq \lceil d(3/2)^{e/5} - 1/2 \rceil.$$

Moreover, $C_e \leq 3^{e/3+1} - 3$.

If the graph is not Eulerian, then [**152**] settles the question of what the path of the robot becomes for a few cases, but nothing is known in the general case. In [**139**], the case of several robots is introduced. The "self-stabilization" seen in robot vacuum is also a feature of so-called *ant algorithms* (such as the well-known *Langton's ant*, which is capable of simulating a universal Turing machine; see [**91**] and [**134**]). The Eulerian results of [**152**] are reminiscent of those in [**202**] but the lead-in times are different.

9.7. Combinatorial Games

We have mentioned many games in this book, and it is worthwhile mentioning that there is a field of study called *combinatorial game* *theory*. In this subject, there are two players who move alternately, there are no chance devices, and there is perfect information (both players know the state of play at all times). *Winning Ways* [**17**] (all four volumes) is the source of the theory in general. *Lessons in Play* [**3**] gives an introduction to the theory when the games are finite and the last person to move wins the game. We will take a closer look at the game of *Angel and Devil* that has some aspects in common with Cops and Robbers. We will first discuss some of the previously mentioned games in the framework of combinatorial games. Most of this is speculative since this had not been the thrust of the research done on these topics.

9.7.1. Cops and Robbers. This game is not necessarily finite and positions can repeat so this is technically a *loopy* game, although if the cops can win, letting the game go on indefinitely seems a waste of time. Adding an incentive for the cops to finish the game might be needed. Also, if the cops can win, then the robber cannot be allowed to make the last move. The rules can easily be amended so as to not allow the robber to move onto a vertex occupied by a cop. Hill [**113**] considers not allowing a cop or the robber to revisit a vertex. Small examples indicate that a good strategy is to force your opponent into a smaller subgraph than the one you are in and just run him out of moves without having to capture or avoid him. None of the other cops or marshals and robber variants have been considered in this light.

9.7.2. Seepage. In this case, we impose the condition that the game is over when the lake is contaminated or when the Greens have constructed a cutset between the contaminated vertices and the lake. This is an *all-small* game (either both players can move or neither can), and a measure, *atomic weight*, that approximates the game theoretic value, can give information about the game. Specifically, if the atomic weight is a positive integer k, then the Greens can let the Sludge have at least $k-1$ moves before they have to respond, and then only one Green need move! If k is a negative integer, then to win, the Greens need to protect between $|k|-2$ and $|k|+1$ vertices on the next move, and then only one vertex per turn after that to win.

Fortunately (or unfortunately?), the atomic weight does not have to be an integer, and as the situation becomes more complicated so does the atomic weight.

9.7.3. Intelligent Firefighting. For a finite graph the game ends in a finite number of moves but the goals are not well defined enough to know who has won. Nonetheless, a score can be defined: -1 for each vertex that the fire consumes, and $+1$ for each vertex that firefighters have saved. The game *Go* is another scoring game. In the very end of the end-game, each player expects to get one point. The analysis becomes simplified if each player is forced to pay a point (taxed or *cooled* by one) on each move. Such an approach is possible for Intelligent Firefighting, except we tax the firefighters by one and the fire by k.

9.7.4. Cleaning. This is not naturally a game, but more of a puzzle. It can be turned into a game by allowing the two players to put a brush anywhere, and when a vertex is primed it then fires. The player who causes the graph to be cleaned wins. P. Prałat and P. Gordinowicz [**101**] prove that the second player wins on all complete graphs except K_1.

9.7.5. The Angel and Devil. The Angel and the Devil play their game on an infinite chessboard, with one square for each ordered pair of integers (x, y). On his turn, the Devil may eat any square of the board whatsoever; this square is then no longer available to the Angel. The Angel, of power d, can move to any uneaten square (X, Y) that is at most d moves away from its present position (x, y); that is, to any (X, Y) where $|X - x| \leq d$ and $|Y - y| \leq d$. The Devil wins if he can *strand* the Angel; that is, surround him by a moat of eaten squares of width at least d. The Angel wins just if he can continue to move forever. The main question here is: *For which d can the Devil win*?

The problem was introduced in [**17**] volume 3, page 643, and in [**63**], Conway asked whether an Angel of some power can defeat the Devil. In the same paper. Conway explores some strategies that the Angel might try: potential functions that are sensitive to locally eaten squares, potential functions that are sensitive to distant eaten squares, always increase the y-coordinate, never decrease the y-coordinate, or always increase $x + y$. He shows that in all cases and for any d, the Devil will catch the Angel.

In the last five years, a number of breakthroughs have occurred. Kutz [**131**] showed that in three dimensions, an Angel of power 13 can escape. In the same year, and independently, Bollobás and Leader [**20**] also showed the same result, with $d = 20,000$, but the arguments could be tightened to reduce that value.

In 2006 Bowditch [**32**] proved that with $d = 4$, the Angel has a computable winning strategy. And again in the same year, Mathe [**146**] proved the same with $d = 2$.

9.7.6. Hex and Maker- games. Hex is played on a $n \times n$ board where the cells are hexagons. Two opposite sides are colored red and the other two white. One player places red tokens the other white and they attempt to build a (connected) path between the two boundaries of their color. This was originally invented by Piet Hein, and independently but later by John Nash. They both proved that the game cannot end in a tie and there is a first player winning strategy. See [**41**] for a brief but informative history. The game was introduced to a wide audience by Gardner [**93**]. Hex is an example of a *Maker-Maker game*—both players want to make a specific subgraph (or other object). Maker-Breaker is a related set of games where one player wants to make a specific subgraph (or object) and the other wants to prevent this. For example, Seepage, which is related to the Shannon Switching game [**136**], is a Maker-Breaker game.

Exercises

1. Verify the formula (9.1).
2. Give infinitely many examples of graphs where the optimum strategy for the firefighter is to protect vertices which are non-adjacent to burning vertices.
3. Prove Lemma 9.3.
4. Prove that two firefighters are enough to contain a fire breaking out in the infinite hexagonal grid.
5. [**149**] The *infinite triangular grid* T is formed by tiling the plane regularly with equilateral triangles. Prove that if a finite number $k > 0$ of fires break out anywhere in T, then three firefighters suffice to contain the fire.
6. [**148**] The *infinite strong half grid* S is defined as the first and fourth quadrants of the strong grid. Show that two firefighters cannot contain a fire breaking out in S.
7. Prove Theorem 9.6.
8. In Figure 9.4, the source is contaminated and there is only one Green. Who wins when the Greens move first? When the Sludge moves first?
9. [**55**] Derive an $O(n^2)$ algorithm that determines if a rooted directed tree is green-win.
10. [**55**] Prove Theorem 9.10.
11. (a) Derive that $\rho_k(K_n) = \frac{\lceil (n-1)/(k+1) \rceil}{n}$.
    (b) Show that $\rho_k(P_n) = \rho(P_n) = 1 - \frac{2}{n} + \frac{2}{n^2}$.
12. (a) Prove that if H is a subgraph of G, then $tw(H) \leq tw(G)$.
    (b) Use Theorem 9.18 to prove that the $n \times n$ Cartesian grid has treewidth n.
13. Prove Theorem 9.23.
14. Prove items (1), (2), and (3) of Theorem 9.28.
15. Let H be an induced subgraph of G. Show that $b(G) \geq b(H)$.

16. (a) Verify directly (that is, without Theorem 9.30) that $B(P_7) = 6$.
    (b) Generalize (a) to show directly that $B(P_n) = n - 1$.
17. Prove that if H is a subgraph of G, then $B(H) \leq B(G)$.
18. [**155**] Prove that $B(G) = b(p)$ if and only if G is a disjoint union of cliques.
19. Prove Theorem 9.34 in the case that G is bipartite.
20. Prove Theorem 9.35 (1).
21. Verify that it takes 10 (and 30) time-steps to clean the upper (and lower) weightings (respectively) of the graph in Figure 9.11 using the robot vacuum.
22. If G is a finite connected graph, then show that the robot vacuum will eventually clean every edge of G.
23. (a) Prove that a connected graph G is Eulerian if and only if every vertex of G has even degree.
    (b) Prove that a connected graph G is semi-Eulerian if and only if there are at most two vertices of G with odd degree.

Bibliography

[1] I. Adler, Marshals, monotone Marshals, and hypertree-width, *Journal of Graph Theory* **47** (2004) 275–296.

[2] M. Aigner, M. Fromme, A game of cops and robbers, *Discrete Applied Mathematics* **8** (1984) 1–11.

[3] M.H. Albert, R.J. Nowakowski, D. Wolfe, *Lessons in Play,* A K Peters, Ltd., 2007.

[4] R. Albert, H. Jeong, A. Barabási, Diameter of the world-wide web, *Nature* **401** (1999) 130.

[5] N. Alon, P. Prałat, N. Wormald, Cleaning regular graphs with brushes, *SIAM Journal on Discrete Mathematics* **23** (2008/09) 233–250.

[6] N. Alon, J. Spencer, *The Probabilistic Method*, Wiley, New York, 2000.

[7] B. Alspach, Sweeping and searching in graphs: a brief survey, *Matematiche* **59** (2006) 5–37.

[8] B. Alspach, D. Dyer, D. Hanson, B. Yang, Lower bounds on edge searching, *ESCAPE 2007*, B. Chen, M. Paterson, G. Zhang (Eds.), **4614** (2007) 516–527.

[9] B. Alspach, X. Li, B. Yang, Searching graphs and directed graphs, preprint.

[10] T. Andreae, Note on a pursuit game played on graphs, *Discrete Applied Mathematics* **9** (1984) 111–115.

[11] T. Andreae, On a pursuit game played on graphs for which a minor is excluded, *Journal of Combinatorial Theory, Series B* **41** (1986) 37–47.

[12] R.P. Anstee, M. Farber, On bridged graphs and cop-win graphs, *Journal of Combinatorial Theory, Series B* **44** (1988) 22–28.

[13] K. Appel, W. Haken, J. Koch, Every planar map is four colorable, *Illinois Journal of Mathematics* **21** (1977) 439-567.

[14] W. Baird, *Cops, Robbers, and Graphs*, M.Sc. Thesis, Ryerson University, 2011.

[15] W. Baird, A. Bonato, Meyniel's conjecture on the cop number: a survey, Preprint 2011.

[16] A. Berarducci, B. Intrigila, On the cop number of a graph, *Advances in Applied Mathematics* **14** (1993) 389–403.

[17] E.R. Berlekamp, J.H. Conway, R.K. Guy, *Winning ways for your mathematical plays,* Volumes 1, 2, 3, and 4, Second Editions, A K Peters, Ltd., 2001–2004.

[18] B. Bollobás, *Random graphs, Second edition,* Cambridge Studies in Advanced Mathematics, **73**, Cambridge University Press, Cambridge, 2001.

[19] B. Bollobás, G. Kun, I. Leader, Cops and robbers in a random graph, Preprint 2011.

[20] B. Bollobás, I. Leader, The Angel and the Devil in three dimensions, *Journal of Combinatorial Theory, Series A* **113** (2006) 176–184.

[21] A. Bonato, *A Course on the Web Graph*, Graduate Studies in Mathematics, American Mathematical Society, Providence, Rhode Island, 2008.

[22] A. Bonato, E. Chiniforooshan, Pursuit and evasion from a distance: algorithms and bounds, In: *Proceedings of Workshop on Analytic Algorithms and Combinatorics (ANALCO'09)*, 2009.

[23] A. Bonato, E. Chiniforooshan, P. Prałat, Cops and Robbers from a distance, *Theoretical Computer Science* **411** (2010) 3834–3844.

[24] A. Bonato, D. Delić, On a problem of Cameron's on inexhaustible graphs, *Combinatorica* **24** (2004) 35–51.

[25] A. Bonato, N. Hadi, P. Horn, P. Prałat, C. Wang, Models of on-line social networks, *Internet Mathematics* **6** (2011) 285–313.

[26] A. Bonato, G. Hahn, P.A. Golovach, J. Kratochvíl, The capture time of a graph, *Discrete Mathematics* **309** (2009) 5588–5595.

[27] A. Bonato, G. Hahn, C. Tardif, Large classes of infinite k-cop-win graphs, *Journal of Graph Theory* **65** (2010) 334–242.

[28] A. Bonato, G. Hahn, C. Wang, The cop density of a graph, *Contributions to Discrete Mathematics* **2** (2007) 133–144.

[29] A. Bonato, G. Kemkes, P. Prałat, Almost all cop-win graphs contain a universal vertex, Preprint 2011.

[30] A. Bonato, M.E. Messinger, P. Prałat, Fighting intelligent fires in graphs, Preprint 2011.

[31] A. Bonato, P. Prałat, C. Wang, Pursuit-evasion in models of complex networks, *Internet Mathematics* **4** (2009) 419–436.

[32] B.H. Bowditch, The Angel Game in the plane, *Combinatorics, Probability and Computing,* **16** (2007) 345–362.

[33] A. Brandstädt, V.B. Le, J.P. Spinrad, *Graph Classes: A Survey*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 1999.

[34] G. Brightwell, P. Winkler, Gibbs measures and dismantlable graphs, *Journal of Combinatorial Theory, Series B* **78** (2000) 141–169.

[35] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Graph structure in the web, *Computer Networks* **33** (2000) 309–320.

[36] L. Cai, W. Wang, The safety factor of square grids, preprint.

[37] L. Cai, W. Wang, The surviving rate of a graph, *SIAM Journal of Discrete Mathematics*, **23** (2009) 1814–1826.

[38] L. Cai, Y. Cheng, E. Verbin, Y. Zhou, Surviving rates of trees and outerplanar graphs for the Firefighter problem, *SIAM Journal on Discrete Mathematics* **24** (2010) 1322–1335.

[39] P.J. Cameron, *Combinatorics: Topics, Techniques, Algorithms*, Cambridge University Press, Cambridge, 1995.

[40] P.J. Cameron, The random graph, In: *The Mathematics of Paul Erdős*, *II*, Algorithms and Combinatorics, **14**, Springer, Berlin, 1997, pp. 333–351.

[41] G. Campbell, On optimal play in the game of Hex, *INTEGERS* **4** (2004) #G2, 23pp.

[42] G. Cantor, Beiträge zur begründung der transfiniten mengenlehre, *Mathematische Annalen* **49** (1897) 207–246.

[43] J. Chalopin, V. Chepoi, N. Nisse, Y. Vaxés, Cop and robber games when the robber can hide and ride, Technical Report, *INRIA-RR7178*, Sophia Antipolis, France, Jan. 2010.

[44] M. Chastand, F. Laviolette, N. Polat, On constructible graphs, infinite bridged graphs and weakly cop-win graphs, *Discrete Mathematics* **224** (2000) 61–78.

[45] P. Chebyshev, Mémoire sur les nombres premiers, *Mém. Acad. Sci. St. Pétersbourg* **7** (1850) 17–33.

[46] V. Chepoi, Bridged graphs are cop-win graphs: an algorithmic proof, *Journal of Combinatorial Theory, Series B* **69** (1997) 97–100.

[47] E. Chiniforooshan, A better bound for the cop number of general graphs, *Journal of Graph Theory* **58** (2008) 45–48.

[48] M. Chudnovsky, N. Robertson, P. Seymour, R. Thomas, The strong perfect graph theorem, *Annals of Mathematics* **164** (2006) 51–229.

[49] F.R.K. Chung, L. Lu, *Complex graphs and networks*, American Mathematical Society, Providence RI, 2006.

[50] N.E. Clarke, A witness version of the Cops and Robber game, accepted to *Discrete Mathematics*.

[51] N.E. Clarke, *The effects of replacing cops and searchers with technology*, M.Sc. Thesis, Dalhousie University, 1999.

[52] N.E. Clarke, *Constrained Cops and Robber*, Ph.D. Thesis, Dalhousie University, 2002.

[53] N.E. Clarke, A game of Cops and Robber played with partial information, *Congressus Numerantium* **166** (2004) 145–159.

[54] N.E. Clarke, E.L. Connon, Cops, Robber, and alarms, *Ars Combinatoria* **81** (2006) 283–296.

[55] N.E. Clarke, S. Finbow, S.L. Fitzpatrick, M.E. Messinger, R.J. Nowakowski, Seepage in directed acyclic graphs, *Australasian Journal of Combinatorics* **43** (2009) 91–102.

[56] N.E. Clarke, S.L. Fitzpatrick, A. Hill, R.J. Nowakowski, Edge critical Cops and Robber, accepted to *Discrete Mathematics*.

[57] N.E. Clarke, G. MacGillivray, Characterizations of k-copwin graphs, Preprint 2011.

[58] N.E. Clarke, R.J. Nowakowski, Cops, Robber and photo radar, *Ars Combinatoria* **56** (2000), 97–103.

[59] N.E. Clarke, R.J. Nowakowski, Cops, robber, and traps, *Utilitas Mathematica* **60** (2001) 91–98.

[60] N.E. Clarke, R.J. Nowakowski, A tandem version of the Cops and Robber game played on products of graphs, *Discussiones Mathematicae Graph Theory* **25** (2005) 241–249.

[61] N.E. Clarke, R.J. Nowakowski, Tandem-win graphs, *Discrete Mathematics* **299** (2005) 56–64.

[62] Clay Mathematics Institute Millenium Prize Problems. Accessed March 31, 2011. `http://www.claymath.org/millennium/`

[63] J.H. Conway, The angel problem, *Games of No Chance*, R.J. Nowakowski editor, Cambridge University Press, 1996, pp. 3–12.

[64] S. Cook, The complexity of theorem proving procedures, In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, 1971.

[65] R. Crandall, C. Pomerance, *Prime Numbers: A Computational Perspective, 2nd ed.* Springer-Verlag, New York, 2005.

[66] R.M. Damerell, On Moore graphs, *Proceedings of the Cambridge Philosophical Society* **74** (1973) 227–236.

[67] M. Devlin, S. Hartke, Fire containment in grids of dimension three and higher, *Discrete Applied Mathematics* **155** (2007) 2257–2268.

[68] R. Diestel, *Graph theory*, Springer-Verlag, New York, 2000.

[69] R.P. Dilworth, A decomposition theorem for partially ordered sets, *Annals of Mathematics* **51** (1950) 161–166.

[70] P.A. Dreyer, Applications and variations of domination in graphs, Ph.D. Dissertation, Department of Mathematics, Rutgers University, 2000.

[71] J.A. Ellis, I.H. Sudborough, J.S. Turner, The vertex separation and search number of a graph, *Information and Computation* **113** (1994) 50–79.

[72] P. Erdős, Beweis eines Satzes von Tschebyschef, *Acta Sci. Math. (Szeged)* **5** (1930-32) 194–198.

[73] P. Erdős, A. Rényi, On random graphs I, *Publicationes Mathematicae Debrecen* **6** (1959) 290–297.

[74] P. Erdős, A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hungar. Acad. Sci.* **5** (1960) 17–61.

[75] P. Erdős, A. Rényi, Asymmetric graphs, *Acta Mathematica Academiae Scientiarum Hungaricae* **14** (1963) 295–315.

[76] S. Finbow, G. MacGillivray, The Firefighter problem: a survey of results, directions and questions, *Australasian Journal of Combinatorics* **43** (2009) 57–77.

[77] S. Finbow, A.D. King, G. MacGillivray, R. Rizzi, The Firefighter problem for graphs of maximum degree three, *Discrete Mathematics* **307** (2007) 2094–2105.

[78] S. Finbow, P. Wang, W. Wang, On the surviving rate of a graph, preprint.

[79] S.L. Fitzpatrick, *Aspects of domination and dynamic domination*, Ph.D. Thesis, Dalhousie University, 1997.

[80] S.L. Fitzpatrick, R.J. Nowakowski, Copnumber of graphs with strong isometric dimension two, *Ars Combinatoria* **59** (2001) 65–73.

[81] S.L. Fitzpatrick, R.J. Nowakowski, D. Holton, I. Caines, Covering hypercubes by isometric paths, *Discrete Mathematics* **240** (2001) 253–260.

[82] P. Fogarty, *Catching the fire on grids*, M.Sc. Thesis (2003), Department of Mathematics, University of Vermont.

[83] S. Foldes, P.L. Hammer, Split graphs, In: *Proceedings of the 8th South-Eastern Conference on Combinatorics, Graph Theory and Computing*, 1977.

[84] F.V. Fomin, P. Goldvach, J. Kratochvíl, On tractability of the Cops and Robbers game. In: *Fifth IFIP International Conference On Theoretical Computer Science- TCS 2008, IFIP 20th World Computer Congress, TC 1, Foundations of Computer Science*, 2008.

[85] F.V. Fomin, P.A. Golovach, J. Kratochvíl, N. Nisse, Pursuing fast robber in graphs, *Theoretical Computer Science* **411** (2010) 1167–1181.

[86] F.V. Fomin, P.A. Golovach, D. Lokshtanov, Cops and Robber game without recharging, In: *Proceedings of 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2010)*, 2010.

[87] F.V. Fomin, D.M. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science* **399** (2008) 236–245.

[88] P. Frankl, On a pursuit game on Cayley graphs, *Combinatorica* **7** (1987) 67–70.

[89] P. Frankl, Cops and robbers in graphs with large girth and Cayley graphs, *Discrete Applied Mathematics* **17** (1987) 301–305.

[90] A. Frieze, M. Krivelevich, P. Loh, Variations on Cops and Robbers, accepted to *Journal of Graph Theory.*

[91] A. Gajardo, A. Moreira, E. Goles, Complexity of Langton's ant, *Discrete Applied Mathematics* **117** (2002) 41–50.

[92] A. Gardiner, Homogeneous graphs, *Journal of Combinatorial Theory, Series B* **20** (1976) 94–102.

[93] M. Gardner, *Hexaflexagons and Other Mathematical Diversions*, University of Chicago Press, Chicago, 1988.

[94] M. Garey, D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, Freemann, San Francisco, 1979.

[95] S. Gaspers, M.E. Messinger, R.J. Nowakowski, P. Prałat, Parallel cleaning of a network with brushes, *Discrete Applied Mathematics* **158** (2010) 467–478.

[96] S. Gaspers, M.E. Messinger, R.J. Nowakowski, P. Prałat, Clean the graph before you draw it! *Information Processing Letters* **109** (2009) 463–467.

[97] T. Gavenčiak, *Games on graphs*, Master's thesis, Department of Applied Mathematics, Charles University, Prague, 2007.

[98] T. Gavenčiak, Cop-win graphs with maximal capture-time, *Studentská vědecká a odbornáčinnost (SVOC)*, 2008.

[99] Y. Gol'fand, M. Klin, On k-homogeneous graphs, *Algorithmic Studies in Combinatorics*, Work Collect., Moskva (1978) 76–85.

[100] A.S. Goldstein, E.M. Reingold, The complexity of pursuit on a graph, *Theoretical Computer Science* **143** (1995) 93–112.

[101] P. Gordinowicz, P. Prałat, personal communication.

[102] G.R. Grimmett, D.R. Stirzaker, *Probability and Random Processes*, 3rd Edition, Oxford University Press, 2001.

[103] G. Hahn, Cops, robbers and graphs, *Tatra Mountain Mathematical Publications* **36** (2007) 163–176.

[104] G. Hahn, F. Laviolette, N. Sauer, R.E. Woodrow, On cop-win graphs, *Discrete Mathematics* **258** (2002) 27–41.

[105] G. Hahn, G. MacGillivray, A characterization of k-cop-win graphs and digraphs, *Discrete Mathematics* **306** (2006) 2492–2497.

[106] Y.O. Hamidoune, On a pursuit game on Cayley digraphs, *European Journal of Combinatorics* **8** (1987) 289–295.

[107] F. Harary, G.W. Wilcox, Boolean operations on graphs, *Mathematica Scandinavica* **20** (1967) 41–51.

[108] B. Hartnell, Firefighter! An application of domination. Presentation at the *25th Manitoba Conference on Combinatorial Mathematics and Computing*, University of Manitoba, Winnipeg, Canada, 1995.

[109] B. Hartnell, Q. Li, Firefighting on trees: How bad is the greedy algorithm?, In: *Proceedings of the Thirty-first Southeastern International Conference on Combinatorics, Graph Theory and Computing*, 2000.

[110] B. Hartnell, D. Rall, On Vizing's Conjecture, *Congressus Numerantium* **82** (1991) 87–96.

[111] P. Hell, I. Rival, Absolute retracts and varieties of reflexive graphs, *Canadian Journal of Mathematics* **39** (1987) 544–567.

[112] C.W. Henson, A family of countable homogeneous graphs, *Pacific Journal of Mathematics* **38** (1971) 69–83.

[113] A. Hill, *Cops and Robbers: Theme and Variations*, Ph.D. Thesis, Dalhousie University, 2008.

[114] B. Hobbs, J. Kahabka, Underwater Cleaning Technique Used for Removal of Zebra Mussels at the Fitzpatrick Nuclear Power Plant, In: *Proceedings of The Fifth International Zebra Mussel and Other Aquatic Nuisance Organisms Conference, The Sea Grant Nonindigenous Species Site*. 1995.

[115] A.J. Hoffman, R.R. Singleton, On Moore graphs with diameter 2 and 3, *IBM Journal of Research and Development* **4** (1960) 497–504.

[116] J.E. Hopcroft, R. Motwani, J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 3rd Edition*, Addison-Wesley, 2006.

[117] P. Hunter, S. Kreutzer, Digraph measures: Kelly decompositions, games, and orderings, *Theoretical Computer Science* **399** (2008) 206–219.

[118] W. Imrich, H. Izbicki, Associative Products of Graphs, *Monatshefte fur Mathematik* **80** (1975) 277–281.

[119] W. Imrich, S. Klavzar, *Product graphs, Structure and Recognition*, Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[120] A. Isaza, J. Lu, V. Bulitko, R. Greiner, A cover-based approach to multi-agent moving target pursuit, In: *Proceedings of The 4th Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2008.

[121] V. Isler, S. Kannan, S. Khanna, Randomized Pursuit-Evasion with Local Visibility, *SIAM Journal on Discrete Mathematics* **1** (2006) 26–41.

[122] V. Isler, S. Kannan, S. Khanna, Randomized Pursuit-Evasion in a Polygonal Environment, *IEEE Transactions on Robotics* **21** (2005) 864–875.

[123] S. Janson, T. Łuczak, A. Ruciński, *Random Graphs*, Wiley, New York, 2000.

[124] D.V. Jeliazkova, *Aspects of the Cops and Robber Game played with incomplete information*, M.Sc. Thesis, Acadia University, 2006.

[125] G. Joret, M. Kamiński, D.O. Theis, The cops and robber game on graphs with forbidden (induced) subgraphs, *Contributions to Discrete Mathematics* **5** (2010) 40–51.

[126] R.M. Karp, Reducibility among combinatorial problems, In: *Complexity of Computer Computations*, Raymond E. Miller and James W. Thatcher (editors), New York: Plenum, pp. 85–103, 1972.

[127] J. Kára, K. Kratochvíl, D. Wood, On the complexity of the balanced vertex ordering problem, *Discrete Mathematics and Theoretical Computer Science* **9** (2007) 193–202.

[128] S. R. Kotler, E.C. Mallen, K.M. Tammus, Robotic Removal of Zebra Mussel Accumulations in a Nuclear Power Plant Screenhouse, *Proceedings of the Fifth International Zebra Mussel and Other Aquatic Nuisance Organisms Conference, The Sea Grant Nonindigenous Species Site*, 1995.

[129] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Trawling the web for emerging cyber-communities, In: *Proceedings of the 8th WWW Conference*, 1999.

[130] K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fundamenta Mathematicae* **15** (1930) 271–283.

[131] M. Kutz, Conway's angel in three dimensions, *Theoretical Computer Science* **349** (2005) 443–451.

[132] A.H. Lachlan, R.E. Woodrow, Countable ultrahomogeneous undirected graphs, *Transaction of the American Mathematical Society* **262** (1980) 51–94.

[133] C.W.H. Lam, The search for a finite projective plane of order 10, *American Mathematical Monthly* **98** (1991) 305–318.

[134] C.G. Langton, Studying artificial life with cellular automata, *Physica D: Nonlinear Phenomena* **22** (1986) 120–149.

[135] A.S. LaPaugh, Recontamination does not help to search a graph, *Journal of the Association for Computing Machinery* **40** (1993) 224–245.

[136] A. Lehman, A solution to the Shannon switching game, *SIAM Journal on Applied Mathemtics* **12** (1964) 687–725.

[137] L. Levin, Universal search problems, *Problems of Information Transmission* **9** (1973) 265–266. (Russian)

[138] H. Lewis, C.H. Papadimitriou, *Elements of the Theory of Computation, 2nd Edition*, Prentice Hall, Upper Saddle River, NJ, 1997.

[139] Z. Li, A. Vetta, Bounds on the cleaning times of robot vacuums, *Operations Research Letters* **38** (2010) 69–71.

[140] L. Lovász, Operations with structures, *Acta Mathematica Hungarica* **18** (1967) 321–328.

[141] L. Lu, X. Peng, On Meyniel's conjecture of the cop number, Preprint 2011.

[142] T. Łuczak, P. Prałat, Chasing robbers on random graphs: Zig-Zag theorem, *Random Structures and Algorithms* **37** (2010) 516–524.

[143] E.M. Luks, L. Babai, Canonical labeling of graphs, In: *Proceedings of the 15th ACM Symposium on Theory of Computing*, 1983.

[144] M. Maamoun, H. Meyniel, On a game of policemen and robber, *Discrete Applied Mathematics* **17** (1987) 307–309.

[145] G. MacGillivray, P. Wang, On the Firefighter problem, *Journal of Combinatorial Mathematics and Combinatorial Computing* **47** (2003) 83-96.

[146] A. Mathe, The Angel of power 2 wins, *Combinatorics, Probability and Computing* **16** (2007) 363–374.

[147] N. Megiddo, S.L. Hakimi, M.R. Garey, D.S. Johnson, C.H. Papadimitriou, The complexity of searching a graph, *Journal of the Association for Computing Machinery* **35** (1988) 18–44.

[148] M.E. Messinger, *Firefighting on Infinite Grids*, M.Sc. Thesis, Department of Mathematics and Statistics, Dalhousie University, 2004.

[149] M.E. Messinger, Firefighting on the triangular grid, *Journal of Combinatorial Mathematics and Combinatorial Computing* **63** (2007) 37–45.

[150] M.E. Messinger, Firefighting on the strong grid, Preprint 2011.

[151] M.E. Messinger, *Methods of Decontaminating a Network*, PhD. Thesis, Department of Mathematics and Statistics, Dalhousie University, 2008.

[152] M.E. Messinger, R.J. Nowakowski, The robot cleans up, *Journal of Combinatorial Optimization* **18** (2009) 350–361.

[153] M.E. Messinger, R.J. Nowakowski, P. Prałat, Cleaning a network with brushes, *Theoretical Computer Science* **399** (2008) 191–205.

[154] M.E. Messinger, R.J. Nowakowski, P. Prałat, N. Wormald, Cleaning random regular graphs with brushes using a degree-greedy algorithm, In: *Proceedings of the 4th Workshop on Combinatorial and Algorithmic Aspects of Networking (CAAN 2007)*, 2007.

[155] M.E. Messinger, R.J. Nowakowski, P. Prałat, Cleaning with brooms, *Graphs and Combinatorics* **27** (2011) 251–267.

[156] M. Miller, J. Sirán, Moore graphs and beyond: A survey of the degree/diameter problem, *Electronic Journal of Combinatorics* DS14, 2005.

[157] S.A. Moeller, P. Wang, Fire control on graphs, *Journal of Combinatorial Mathematics and Combinatorial Computing* **41** (2002) 19–34.

[158] C. Moldenhauer, Game tree search algorithms for the game of Cops and Robber, M.Sc. Thesis, University of Edmonton, 2009.

[159] C. Moldenhauer, N. Sturtevant, Evaluating strategies for running from the cops, In: *Proceedings of IJCAI*, 2009.

[160] J.W. Moon, *Topics on Tournaments*, Holt, Rinehart and Winston, New York, 1968.

[161] M. Musson, A. Tang, *Cops and Robber with road blocks*, Preprint 2011.

[162] J. Nešetřil, V. Rödl, Products of graphs and their applications, *Graph Theory, Łagów 1981, Lecture Notes in Mathematics 1018*, Springer, Berlin, pp 151–160, 1983.

[163] S. Neufeld, *The game of Cops and Robber*, M.Sc. Thesis, Dalhousie University, 1990.

[164] S. Neufeld, R.J. Nowakowski, A game of cops and robbers played on products of graphs, *Discrete Mathematics* **186** (1998) 253–268.

[165] S. Neufeld, R.J. Nowakowski, A vertex-to-vertex pursuit game played with disjoint sets of edges, *Finite and infinite combinatorics in sets and logic* (Banff, AB, 1991), Kluwer Acad. Publ., 1993, 411, 299–312

[166] R.J. Nowakowski, D.F. Rall, Associative graph products and their independence, domination and coloring numbers, *Discussiones Mathematicae Graph Theory* **16** (1996) 53–79.

[167] R.J. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Mathematics* **43** (1983) 235–239.

[168] P. Prałat, Cleaning random graphs with brushes, *Australasian Journal of Combinatorics* **43** (2009) 237–251.

[169] A. Quilliot, Jeux et pointes fixes sur les graphes, *Thèse de 3ème cycle*, Université de Paris VI, 1978, 131–145.

[170] A. Quilliot, *Problèmes de jeux, de point Fixe, de connectivité et de represésentation sur des graphes, des ensembles ordonnés et des hypergraphes*, Thèse d'Etat, Université de Paris VI, 1983, 131–145.

[171] A. Quilliot, A short note about pursuit games played on a graph with a given genus, *Journal of Combinatorial Theory, Series B* **38** (1985) 89–92.

[172] C.H. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.

[173] T.D. Parsons, Pursuit-evasion in a graph, *Theory and applications of graphs* (Proc. Internat. Conf., Western Mich. Univ., Kalamazoo, Mich., 1976), Springer, 1978, 426-441. Lecture Notes in Mathematics, Vol. 642.

[174] T.D. Parsons, The search number of a connected graph, Proceedings of the Ninth Southeastern Conference on Combinatorics, Graph Theory, and Computing, (Florida Atlantic Univ., Boca Raton, Fla., 1978), *Utilitas Mathematics* (1978) 549–554.

[175] P. Prałat, When does a random graph have constant cop number?, *Australasian Journal of Combinatorics* **46** (2010) 285–296.

[176] P. Prałat, Cleaning random d-regular graphs with brooms, accepted to *Graphs and Combinatorics.*

[177] P. Prałat, Cleaning random graphs with brushes, *Australasian Journal of Combinatorics* **43** (2009) 237–251.

[178] P. Prałat, Graphs with average degree smaller than 30/11 are burning slowly, Preprint 2011.

[179] P. Prałat, N. Wormald, Meyniel's conjecture holds in random graphs, Preprint 2011.

[180] R. Rado, Universal graphs and universal functions, *Acta Arithmetica* **9** (1964) 331–340.

[181] T. Ramanampanoharana, *Jeu de poursuite sur des modèles du web et généralisations*, Masters thesis, Université de Montréal, 2004.

[182] C.S. ReVelle, K.E. Rosing, Defendens imperium romanum: a classical problem in military strategy, *American Mathematical Monthly* **107** (2000) 585–594.

[183] N. Robertson, D.P. Sanders, P. Seymour, R. Thomas, The four-color theorem, *Journal of Combinatorial Theory, Series B* **70** (1997) 2–44.

[184] E.R. Scheinerman, D.H. Ullman, *Fractional graph theory*, Wiley-Interscience, New York, 1997.

[185] B.S.W. Schroeder, The copnumber of a graph is bounded by $\lfloor \frac{3}{2}\text{genus}(G)\rfloor + 3$, *Categorical Perspectives*, Trends Math., Birkhäuser, Boston, MA, 2001, 243–263.

[186] A. Scott, *On the parameterized complexity of finding short winning strategies in combinatorial games*, Ph.D. Thesis, University of Victoria, 2010.

[187] A. Scott, B. Sudakov, A new bound for the cops and robbers problem, Preprint 2011.

[188] P.D Seymour, R. Thomas, Graph searching and a min-max theorem for tree-width, *Journal of Combinatorial Theory Series B* **58** (1993) 22–33.

[189] J. Sheehan, Smoothly embeddable subgraphs, *Journal of the London Mathematical Society* **9** (1974) 212–218.

[190] A. Tang, *Zero-visibility Cops and Robber*, M.Sc. Thesis, Dalhousie University, 2004.

[191] D.O. Theis, The cops and robber game on series-parallel graphs, accepted to *Graphs and Combinatorics.*

[192] R. Tošíc, On cops and robber game, *Studia Scientiarum Mathematicarum Hungarica* **23** (1988) 225–229.

[193] A. Turing, On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society* **42** (1936) 230–265.

[194] V.V. Vazirani *Approximation Algorithms*, Springer-Verlag, Berlin, 2001.

[195] V.G. Vizing, The cartesian product of graphs, *Vyčisl. Sistemy* **9** (1963) 30–43.

[196] B. Wieland, A.P. Godbole, On the domination number of a random graph, *The Electronic Journal of Combinatorics* **8** (2001), #R37.

[197] D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.

[198] A. Wigderson, P, NP and Mathematics—a computational complexity perspective, In: *Proceedings of the ICM'06*, 2007.

[199] N.C. Wormald, Models of random regular graphs, *Surveys in Combinatorics*, J.D. Lamb and D.A. Preece, eds. London Mathematical Society Lecture Note Series, vol. 276, pp. 239–298, Cambridge University Press, Cambridge, 1999.

[200] N.C. Wormald, The differential equation method for random graph processes and greedy algorithms, *Lectures on Approximation and Randomized Algorithms*, PWN, 1999, 73–155.

[201] B. Yandell, *The Honors Class: Hilbert's Problems and Their Solvers,* A. K. Peters Ltd, Nantick, MA, 2003.

[202] V. Yanovski, I.A. Wagner, A.M. Bruckstein, A distributed ant algorithm for efficiently patrolling a network, *Algorithmica* **37** (2003) 165–186.

Index

Photo courtesy of Douglas Hamlyn

Photo courtesy of F. Nowakowski

This book is the first and only one of its kind on the topic of Cops and Robbers games, and more generally, on the field of vertex pursuit games on graphs. The book is written in a lively and highly readable fashion, which should appeal to both senior undergraduates and experts in the field (and everyone in between). One of the main goals of the book is to bring together the key results in the field; as such, it presents structural, probabilistic, and algorithmic results on Cops and Robbers games. Several recent and new results are discussed, along with a comprehensive set of references. The book is suitable for self-study or as a textbook, owing in part to the over 200 exercises. The reader will gain insight into all the main directions of research in the field and will be exposed to a number of open problems.

ISBN 978-0-8218-5347-4

9 780821 853474

STML/61

For additional information
and updates on this book, visit
www.ams.org/bookpages/stml-61