Karlsruhe Institute of Technology

On the Attacking Cop Number of a Graph

Bachelor's Thesis of
Florian Brendle

At the Department of Informatics
Institute of Theoretical Informatics (ITI)

Reviewer:	PD Dr. Torsten Ueckerdt
Second reviewer:	T.T.-Prof. Dr. Thomas Bläsius
Advisor:	PD Dr. Torsten Ueckerdt

30.07.2025

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself. I have not used any other than the aids that I have mentioned. I have marked all parts of the thesis that I have included from referenced literature, either in their original wording or paraphrasing their contents. I have followed the by-laws to implement scientific integrity at KIT.

Karlsruhe, 30.07.2025

- -

(Florian Brendle)

Abstract

We consider the game of Cops and Attacking Robber, a variant of the game of Cops and Robber in which at each robber turn, the robber may eliminate (at most) one cop by moving onto it. The number of cops that is required to capture the robber in a game of Cops and (Attacking) Robber in a graph G is denoted by $\mathrm{c}(G)$ and $\mathrm{cc}(G)$ respectively.
It is easy to see that $\mathrm{c}(G) \leq \mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$ for all connected graphs G. We prove a conjecture by A. Clow, M. A. Huggan, and M. Messinger [1] that the latter inequality is best possible, i.e. that there are connected graphs G with arbitrarily large cop numbers satisfying $\mathrm{cc}(G) = 2\,\mathrm{c}(G)$.
S. Neufeld and R. Nowakowski [2] show that $\mathrm{c}(G \boxtimes H) \leq \mathrm{c}(G) + \mathrm{c}(H) - 1$ for all connected graphs G and H, where $\boxtimes$ denotes the strong product. We give a similar bound for $\mathrm{cc}(G \boxtimes H)$. Using this result, we reduce computing $\mathrm{c}(G)$ for a graph G to computing $\mathrm{cc}(G')$ for a graph G', thereby proving that computing the latter is EXPTIME-complete.
In some graphs, the cops can capture the robber only if they sacrifice some cops first. We show that there are graphs G so that if $\mathrm{cc}(G)$ cops have to capture a robber in G, the robber can eliminate all cops but one before being captured.

Zusammenfassung

Wir betrachten das Spiel Cops and Attacking Robber, eine Variante des Spiels Cops and Robber, bei der der Robber pro Zug (höchstens) einen Cop eliminieren darf, indem er auf den Cop zieht. Die benötigte Anzahl an Cops, um den Robber in einem Cops and (Attacking) Robber-Spiel zu fangen, wird mit $\mathrm{c}(G)$ bzw. $\mathrm{cc}(G)$ bezeichnet.
Es ist einfach zu sehen, dass $\mathrm{c}(G) \leq \mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$ für alle zusammenhängenden Graphen G gilt. Wir zeigen die Vermutung von A. Clow, M. A. Huggan, und M. Messinger [1], dass letztere Ungleichung bestmöglich ist, also dass es zusammenhängende Graphen G mit beliebig großem $\mathrm{c}(G)$ gibt, für die $\mathrm{cc}(G) = 2\,\mathrm{c}(G)$ gilt.
S. Neufeld und R. Nowakowski [2] zeigen, dass $\mathrm{c}(G \boxtimes H) \leq \mathrm{c}(G) + \mathrm{c}(H) - 1$ für alle zusammenhängenden Graphen G und H gilt, wobei $\boxtimes$ das strong product bezeichnet. Wir verwenden dieses Resultat, um die Berechnung von $\mathrm{c}(G)$ für einen Graphen G auf die Berechnung von $\mathrm{cc}(G')$ für einen Graphen G' zu reduzieren. Damit zeigen wir, dass die Berechnung des Letzteren EXPTIME-vollständig ist.
In manchen Graphen können die Cops den Robber nur fangen, wenn sie davor einige Cops opfern. Wir zeigen, dass es Graphen G gibt, in denen, wenn $\mathrm{cc}(G)$ Cops den Robber fangen müssen, der Robber alle Cops bis auf einen eliminieren kann, bevor er gefangen wird.

Contents

1 Introduction

1.1 Cops and Robber

Cops and Robber is a two-player game with **perfect information**. For more information on games with perfect information, see E. R. Berlekamp, J. H. Conway, and R. K. Guy [3] and J. Beck [4]. Cops and Robber was first introduced by R. Nowakowski and P. Winkler [5], and independently by A. Quilliot [6].

A game of Cops and Robber is played on a graph G. One party consists of the robber, which we denote by R, and the other party consists of some fixed number of cops. First, each cop chooses a vertex of G as its initial position. Then, R chooses its initial position, also a vertex of G. From now on, the cops and R take turns, beginning with a cop turn. At a cop turn, each cop makes one **move**. Moving means either moving along an edge to an adjacent vertex or **standing still**, i.e. remaining on the same vertex. At R's turn, R moves in the same way as a cop. The cops' goal is for any cop to **capture** R, i.e. to move to R's current position. R's goal is to avoid being captured indefinitely. The **cop number** $\mathbf{c}(G)$ of a graph G is the smallest number of cops so that they can capture R in a game of Cops and Robber in G (no matter how R acts)[1].

Cops and Robber has been studied on planar graphs (by M. Aigner and M. Fromme [7], who prove that every planar connected graph has cop number at most 3, and S. Durocher *et al.* [8]) and various products of graphs (by S. Neufeld and R. Nowakowski [2], M. Maamoun and H. Meyniel [9] and B. W. Sullivan and M. Werzanski [10]). A. Berarducci and B. Intrigila [11] gave an algorithm that decides in time $\mathcal{O}\left(|V(G)|^{2k+2}\right)$ whether k cops can capture R in a game of Cops and Robber in a graph G, which was later improved to an algorithm running in time $\mathcal{O}\left(k\ |V(G)|^{k+2}\right)$ by J. Petr, J. Portier, and L. Versteegen [12]. On the other hand, W. B. Kinnersley [13] show that Cops and Robber is EXPTIME-complete. Many variants of Cops and Robber have been studied, e.g. with traps (by N. E. Clarke and R. J. Nowakowski [14]), a faster robber (by J. Chalopin, V. Chepoi, N. Nisse, and Y. Vaxès [15]) and of course an attacking robber. One of the largest open problems about Cops and Robber is Meyniel's conjecture, first mentioned by P. Frankl [16], stating that there is a constant d so that $\mathrm{c}(G) \leq d\sqrt{|V(G)|}$ holds for all connected graphs G (see also E. Chiniforooshan [17], P. Prałat and N. Wormald [18] and the survey by W. Baird and A. Bonato [19]).

1.2 Cops and Attacking Robber

Cops and Attacking Robber was introduced by A. Bonato *et al.* [20] (see also the corresponding master's thesis by A. Haidar [21]).

A game of Cops and Attacking Robber is the same as a game of Cops and Robber, with the additional rule that if R moves to a vertex with cops, one of the cops is **eliminated** from the game (it does not matter which one). R cannot eliminate a cop when choosing its initial position because choosing one's initial position does not count as a move.

Similar to the cop number, the **attacking cop number** $\mathrm{cc}(G)$ of a graph G is the smallest number of cops so that they can capture R in a game of Cops and Attacking Robber in G.

[1]From now on, for conciseness, we omit stating this parenthesized part explicitly.

Cops and Attacking Robber has been studied on bipartite graphs: A. Bonato *et al.* [20] prove that for bipartite connected graphs G, it holds $\mathrm{cc}(G) \leq \mathrm{c}(G) + 2$, and A. Clow, M. A. Huggan, and M. Messinger [1] show that bipartite planar connected graphs have attacking cop number at most 4. A. Clow, M. A. Huggan, and M. Messinger [1] also characterized the triangle-free connected graphs G with $\mathrm{cc}(G) \leq 2$. A. Lacaze-Masmonteil [22] prove that the Cartesian product of k non-empty trees has attacking cop number $\begin{cases} \frac{k}{2}+1 & , k \text{ even} \\ \lfloor \frac{k}{2} \rfloor + 1 \text{ or } \lceil \frac{k}{2} \rceil + 1, & k \text{ odd} \end{cases}$. (Two years later, S. S. Akhtar, S. Das, and H. Gahlawat [23] also show this result under the restriction that all trees are paths.)
It is easy to see that $\mathrm{c}(G) \leq \mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$ for every graph G (for the upper bound, a strategy for $2\,\mathrm{c}(G)$ cops to capture R in a game of Cops and Attacking Robber is to execute the strategy for $\mathrm{c}(G)$ cops to capture R in a game of Cops and Robber, replacing each cop by two cops). Both A. Bonato *et al.* [20] and A. Clow, M. A. Huggan, and M. Messinger [1] investigated how tight the mentioned upper bound is, i.e. how large $\mathrm{cc}(G)$ can be compared to $\mathrm{c}(G)$ for connected graphs G. Using different approaches, they found some connected graphs G with and $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 2k$ for $k = 2$ and $k = 3$ respectively. For connected graphs with larger cop numbers, the question has remained open, but shall be answered in Section 3.

To gain some intuition, we recommend verifying that for $k \geq 3$, it holds $\mathrm{c}(C_k) = \begin{cases} 1, k=3 \\ 2, k \geq 4 \end{cases}$ and $\mathrm{cc}(C_k) = \begin{cases} 1, k=3 \\ 2, 4 \leq k \leq 6 \\ 3, k \geq 7 \end{cases}$.

2 Preliminaries

2.1 Graphs

Note that a game of Cops and (Attacking) Robber in a disconnected graph G can be fully understood by investigating the games in the connected components of G independently; the (attacking) cop number of G is the sum of the (attacking) cop numbers of G's connected components. Thus, we only consider connected graphs in this thesis.

For a vertex v of a graph G, the closed neighborhood $N_G[v]$ of v is the set of v and all vertices adjacent to v.

For a vertex v, among all graphs containing v that we currently consider in an argument, there shall always be exactly one inclusion-maximal (considering the graphs' vertex sets) graph G. We write $N[v]$ and $N(v)$ instead of $N_G[v]$ and $N_G(v)$ for conciseness.

For an edge $\{u, v\}$ of a graph, we use uv as a shorthand notation.

2.2 Different Game Mechanics in Cops and (Attacking) Robber

The states of a Cops and (Attacking) Robber game in a connected graph G with $k \in \mathbb{N}_+$ cops are:

- the initial state where it is the cops' turn, which contains no further information
- the initial state where it is R's turn, which contains the cops' positions
- non-initial states, which contain R's position, the positions of all cops that have not been eliminated and the party whose turn it is

When we say "a state with [SOME ENTITIES]", we implicitly mean "and no other entities". For example, "a state with 3 cops" refers to an initial state where it is R's turn.
By state, we usually mean a non-initial state.

The mechanics of a game define for every state of the game the parties' options to act and how those actions affect the game state. We mostly consider the following mechanics:

Peaceful mechanics are the mechanics of Cops and Robber. They define for the initial states that every entity of the party whose turn it is must choose a vertex as its initial position, and for every non-initial state that every entity of the party whose turn it is must move. The effect of these actions on the game state are as one would expect.

Attacking mechanics are the mechanics of Cops and Attacking Robber. Their only difference to peaceful mechanics is that if R moves to a vertex v on which at least one cop is, one of the cops on v is eliminated from the game (state). (It does not matter which cop is eliminated.) Recall that choosing a vertex as initial position does not count as a move.

In particular, the mechanics do not define in which state to start and do not include winning or win conditions. We use this flexibility of mechanics compared to whole games of Cops and (Attacking) Robber to give strategies for different scenarios than whole games of Cops and (Attacking) Robber. We then use these strategies as parts of strategies for games of Cops and (Attacking) Robber.

2.3 Strategy descriptions: Phases and instructions

Phases:

We break down the description of complex strategies into descriptions of temporally continuous parts of the strategies – **phases**. A phase always begins with a state in which it is the turn of the party for which the described strategy is . A phase ends either with a desired state (i.e. the strategy is done) or with some state in which another phase of the strategy begins. We make sure that these dependencies are acyclic, and that some phase begins with a state in which the desired strategy shall begin. Thus, the phases together form the desired complex strategy.

Instructions:

In descriptions of strategies, we use gray **instruction blocks** for clarity. An instruction in an instruction block may take multiple moves. After executing the instructions inside the instruction block, we implicitly tell the party for which the strategy is to let the opposing party take a turn if it is the opposing party's turn. Thus, after an instruction block, it is always the turn of the party for which the strategy is.

If we only define for a subset of the cops how to move at some cop turn, the other cops shall stand still.

Frequently used instructions:

The instructions we give often have a certain format. Here, we list these formats and explain their actual meaning:

- **"Let ENTITIES do ACTION until CONDITION"**:
  Let ENTITIES execute ACTION. In parallel, at (the begin of) each turn *of ENTITIES*, CONDITION is checked, and if fulfilled, we let ENTITIES stop (to do ACTION).
  This instruction also ends when ACTION is finished.
  For example, ENTITIES could be R, an ACTION could be "stand still" and CONDITION could be "until a cop is adjacent to R".
- **"Let $\mathrm{ENTITIES}_1$ do ACTION_1,
  meanwhile let $\mathrm{ENTITIES}_2$ do ACTION_2
  (, meanwhile let $\mathrm{ENTITIES}_3$ do ACTION_3)"**:
  Let $\mathrm{ENTITIES}_1$ do ACTION_1. In parallel, let $\mathrm{ENTITIES}_2$ do ACTION_2 and let $\mathrm{ENTITIES}_3$ do ACTION_3, until $\mathrm{ENTITIES}_1$ stop/finish with ACTION_1.
- **"Let ENTITY move to v"**:
  Let ENTITY move to v on the shortest possible path.
  We only say that if there is only one shortest path.
  We sometimes say that even if E is already on v.

2.4 Further definitions and notation

At any time in this thesis, for an entity (i.e. cop or robber) E/Ê/E_1/..., we denote its current position by e/ê/e_1/... (i.e. the corresponding lowercase letter). In particular, r is always the vertex on which R currently is.

For some non-initial state where it is R's turn and for some of our two mechanics, we say that a cop C **protects** a vertex v of $N[r]$ if if R moved to v in R's current turn, then some cop on the vertex c could capture R (i.e. move to r) immediately. In other words:

- Under peaceful mechanics, a cop C protects a vertex v of $N[r]$ if and only if $v \in N[c]$.

- Under attacking mechanics, a cop C protects a vertex v of $N[r]$ if and only if either $v = c$ and another cop is on v or $v \in N(c)$.

For an initial state where it is R's turn, we call a vertex r_0 protected if the following holds: If R chose r_0 as initial position, then the cops could capture R immediately.
Under attacking mechanics, we call a cop C protected if c is protected, i.e. if there is another cop in $N[c]$.

When describing a strategy for some cops
and R just moved to a vertex that was protected (or chose a protected vertex as initial position) and capturing R immediately suffices to reach the strategy's desired outcome (usually, the desired outcome is exactly capturing R),
we always let the cops capture R immediately, independently of our description of the strategy.

Thus, in our strategy descriptions, in the cases in which it suffices to capture R immediately, we only have to consider the cases in which R did not move to a vertex that was protected.

3 On the tightness of $\mathrm{cc} \leq 2\,\mathrm{c}$

3.1 Introduction

Recall the following:

Observation 3.1 ($\mathbf{c \leq cc \leq 2\,c}$): For every connected graph G, it holds $\mathrm{c}(G) \leq \mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$.

It has been an open problem how tight the latter inequality is, or in other words, how large $\mathrm{cc}(G)$ can be in terms of $\mathrm{c}(G)$. Investigating this problem, A. Bonato *et al.* [20] prove a lower bound for the attacking cop number of line graphs of certain hypergraphs, and used this to find[2] a connected graph G with $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 2 \cdot 2$, or in other words $\mathrm{cc}(G) - \mathrm{c}(G) = 2$. Considering (nearly) squares of connected graphs, A. Clow, M. A. Huggan, and M. Messinger [1] found connected graphs G with $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 2 \cdot 3$, or in other words $\mathrm{cc}(G) - \mathrm{c}(G) = 3$. However, they used computer assistance to find upper bounds on the cop numbers of the graphs, so their results cannot easily be generalized to higher (attacking) cop numbers. A. Clow, M. A. Huggan, and M. Messinger [1] conjectured the strongest possible version of tightness for the inequality $\mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$, i.e. that for every $k \in \mathbb{N}_+$, there exists a connected graph G with $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 2k$. In this section, we prove this conjecture.

Afterwards, we give an improvement of the upper bound $\mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$ for some connected graphs G.

3.2 $\mathrm{cc} \leq 2\,\mathrm{c}$ is tight

First, we prove the following helpful lower bound, which is a generalization of the lower bound from M. Aigner and M. Fromme [7] that every connected graph G with girth[3] at least 5 has $\mathrm{c}(G) \geq \delta(G)$.

Lemma 3.2 (lower bound for cc when no C_4): If a connected graph G does not contain C_4 as a subgraph, then $\mathrm{cc}(G) \geq \min(\delta(G), \gamma(G))$.

In Lemma 3.2 (lower bound for cc when no C_4), $\delta(G)$ refers to the smallest degree among the vertices of G, and $\gamma(G)$ refers to the domination number[4] of G.

Proof. Let $k := \min(\delta(G), \gamma(G))$. We give a strategy for R to avoid being captured indefinitely in a game of Cops and Attacking Robber against $k - 1$ cops in G:

Since $k - 1 < \gamma(G)$, there is an unprotected vertex r_0 after the cops choose their initial positions.

Let R choose r_0 as initial position.

After the cops' following turn, R is not captured.

---

[2] They claim that the line graph $L(P)$ of the Petersen graph P is such a graph, but forgot to verify that $\gamma(L(P)) > 3$. Unfortunately, it can be seen that $\gamma(L(P)) = 3$. However, one can simply use the generalized Petersen graph GPG(7,2) instead of P; its line graph L(GPG(7,2)) has domination number greater than 3 (because it is 4-regular and has $3 \cdot 7 = 21$ vertices) and satisfies all other requirements of A. Bonato *et al.* [20]'s proof.

[3] The **girth** of a graph G is the length of a shortest cycle contained in G, and ∞ if G does not contain cycles.

[4] A **dominating set** S for a graph G is a set of vertices of G so that every vertex of G is in S or adjacent to a vertex of S. The **domination number** $\boldsymbol{\gamma(G)}$ of a graph G is the size of a smallest dominating set of G.

We observe that a cop C that is alone on a vertex protects at most one vertex in $N(r)$:
- C does not protect c
- if two vertices of $N(r)$ were both also in $N(c)$, they would form a C_4 together with c and r

If at least two cops occupy the same vertex v, they protect at most two vertices in $N(r)$; v (if it is in $N(r)$) and, for the same reason as above, at most one other vertex in $N(r)$.

In summary, the ratio $\frac{\#\text{vertices in } N(r) \text{ protected by cops on } v}{\#\text{cops on } v}$ is at most 1 for vertices v with one cop and for vertices v with at least two cops. Because the number of vertices in $N(r)$ is greater than the number of cops, there is at least one unprotected vertex $u \in N(r)$.

Let R move to u.

Again, R is not captured. We let R repeat moving to an unprotected vertex in $N(r)$ indefinitely. □

Next, we introduce *patterns*, which also play an important role in our construction of connected graphs G with $\mathrm{cc}(G) = 2\,\mathrm{c}(G)$.

Definition 3.3: For $k \in \mathbb{N}_+$, let the **patterns** $\mathbf{pat}_k$ be the set of the words of $([k] \cup \{*\})^k$ that contain exactly one *wildcard symbol* $*$.
A word $w \in [k]^k$ **matches** a pattern $p \in \mathrm{pat}_k$ if the only position at which w differs from p is the one of the wildcard symbol in p.

Note that $|\mathrm{pat}_k| = k^k$.

The concept of patterns can be used to prove the existence of the following graphs, which our construction of connected graphs G with $\mathrm{cc}(G) = 2\,\mathrm{c}(G)$ utilizes.

Lemma 3.4: For all $k \in \mathbb{N}_+$, there exists a connected, k-regular and bipartite graph with girth at least 5.

Proof. Let H be the graph with vertex set $[k]^k \cup \mathrm{pat}_k$ and an edge between each word and pattern that match. Note that H is k-regular and bipartite. Observe that H is connected, and that $C_4 = K_{2,2}$ is not a subgraph of H. The last observation together with H being bipartite implies that H has girth at least 5. □

Theorem 3.5 (connected graphs with $\mathbf{cc} = 2\,\mathbf{c}$): For every $k \in \mathbb{N}_+$, there exists a connected graph G for which $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 2k$.

Proof. For $k = 1$, P_4 fulfills the condition.
Now let $k \geq 2$.

We utilize a connected, k-regular and bipartite graph $H = (V_H, E_H)$ with girth at least 5, e.g. the one from Lemma 3.4. D. Kőnig [24] shows that every k-regular and bipartite graph is k-edge-colorable[5]. Thus, H is k-edge-colorable. Let $\mathrm{col} : E_H \to [k]$ be a k-edge-coloring of H. Each vertex of H is incident to exactly one edge of each color because no two of the k edges incident to it can have the same color. Thus, H contains exactly $\frac{|V_H|}{2}$ edges of each color: one edge per two vertices (its endpoints).

[5]A graph G is called **k-edge-colorable** if there exists a **k-edge-coloring** function $\mathrm{col} : E(G) \to [k]$ for G that assigns different values ("colors") to any two edges that share a vertex.

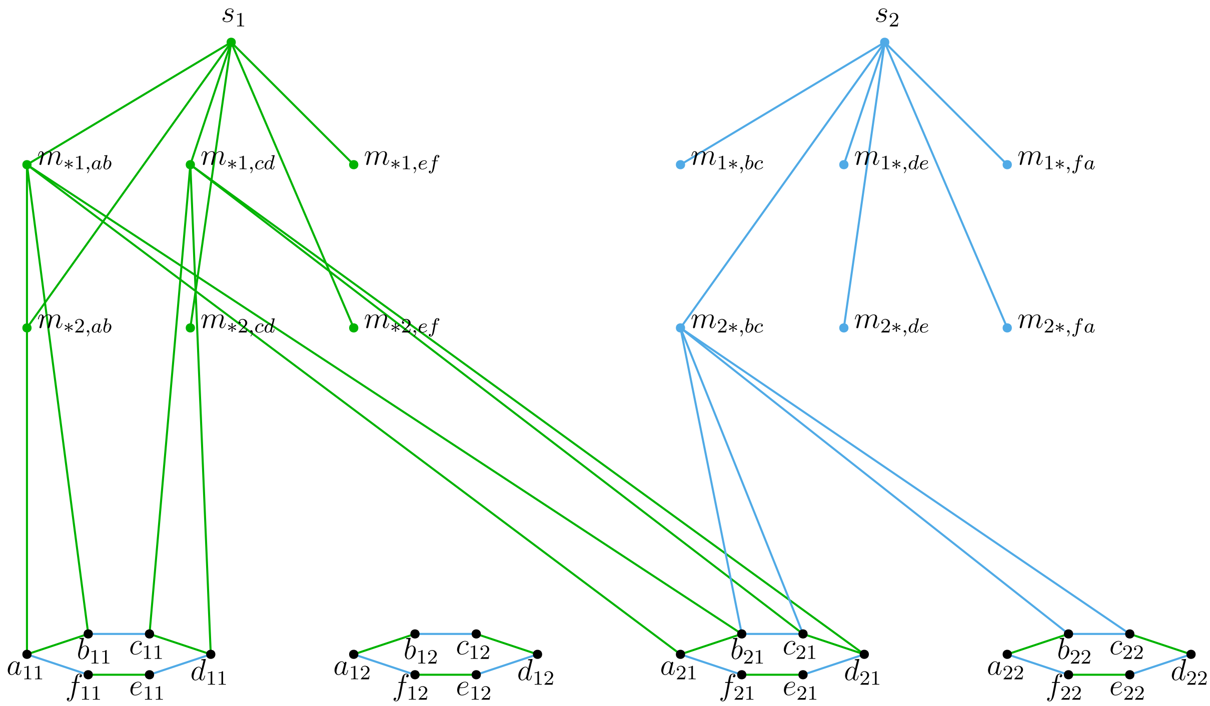

Figure 1: The graph G for $k = 2$, when using a cycle on the vertices $\{a, b, c, d, e, f\}$ as H. For illustration purposes, most of the edges that have a mid vertex and end vertex as endpoint are omitted. However, note that all edges incident to the mid vertices $m_{*1,ab}$, $m_{*1,cd}$ and $m_{2*,bc}$ are depicted.
To gain intuition, we recommend verifying that $m_{*1,ab}$ and $m_{*1,cd}$ only share one neighbor, and that $m_{*1,ab}$ and $m_{2*,bc}$ only share one neighbor.

The graph G fulfilling $\operatorname{cc}(G) = 2\operatorname{c}(G) = 2k$ consists of the following parts:

- **start vertices** $s_1, \ldots, s_k$
- for each $w \in [k]^k$, a copy of H, consisting of an **end vertex** u_w for each vertex u of H and an edge $u_w v_w$ for each edge uv of H
- for each edge $e = uv$ of H and for each $p \in \operatorname{pat}_k$ having the wildcard symbol at the $\operatorname{col}(e)$ -th position, a **mid vertex** $m_{p,e}$ with an edge to the start vertex $s_{\operatorname{col}(e)}$
- for each end vertex u_w and each mid vertex $m_{p,e}$, an edge between them, if u is incident to e and w matches p

See Figure 1 for an illustration of G for $k = 2$ and $H = C_6$.

Observe the following properties of G:

- each start vertex s_c is adjacent to (exactly) the following $k^{k-1}\frac{|V_H|}{2}$ vertices: for each pattern p that has the wildcard symbol at the c-th position and for each edge e of H of color c, the mid vertex $m_{p,e}$
- each mid vertex $m_{p,e=uv}$ is adjacent to (exactly) the following $2k+1$ vertices: the start vertex $s_{\operatorname{col}(e)}$ and for each of the k words w of $[k]^k$ that match p, the end vertices u_w and v_w
- each end vertex u_w is adjacent to (exactly) the following $2k$ vertices:
  for each edge $e = uv$ incident to u in H, the end vertex v_w and the mid vertex $m_{p,e}$, where p is the pattern that w matches and that has the wildcard symbol at the $\operatorname{col}(e)$-th position
- G is connected

Because $\text{cc}(G) \leq 2c(G)$, showing $\text{c}(G) \leq k$ and $\text{cc}(G) \geq 2k$ concludes the proof of the theorem.

First, we show that $\text{c}(G) \leq k$ by giving a strategy for k cops to capture R in a game of Cops and Robber:

Let one cop choose each of the k start vertices as initial position.

R chooses an end vertex u_w as initial position since all other vertices are protected.

In the upcoming cop turn, we shall let the cops move so that afterwards, they protect every vertex in $N[r]$.

Consider an edge $e = uv$ incident to u in H. Let p be the pattern that w matches and that has the wildcard symbol at the $\text{col}(e)$-th position. Note that $m_v := m_{p,e}$ is adjacent to the vertices $s_{\text{col}(e)}$, u_w and v_w.

For each edge $e = uv$ incident to u in H, let the cop on $s_{\text{col}(e)}$ move to m_v.

Because each edge incident to u in H has a different color, each cop is given only one instruction.

Note that after the cops' turn, each of the $2k + 1$ vertices in $N[r]$ is protected. Thus, the cops can capture r in their next turn.

This concludes our cop strategy.

Second, we prove $\text{c}(G) \geq 2k$. Using Lemma 3.2 (lower bound for cc when no C_4), it suffices to show that $C_4 \nsubseteq G$, $\delta(G) \geq 2k$ and $\gamma(G) \geq 2k$.

<u>Claim 1:</u> $C_4 \nsubseteq G$.

Proof of Claim 1. Assume that G contains a subgraph C isomorphic to C_4.

<u>Case 1:</u> C does not contain mid vertices:

The only edges that are not incident to a mid vertex are the ones inside copies of H. Thus, C is subgraph of a copy of H. H has girth at least 5. ↯

<u>Case 2:</u> C contains exactly one mid vertex $m_{p,e}$:

The only non-end vertex to which $m_{p,e}$ is adjacent is $s_{\text{col}(e)}$. Since $s_{\text{col}(e)}$ is only adjacent to mid vertices and C only contains one mid vertex, $s_{\text{col}(e)}$ cannot be in C. Summarizing, the three other vertices of C are each in a copy of H. Because vertices of different copies of H are not adjacent, all three vertices are in the same copy of H. But $m_{p,e}$ is adjacent to at most two vertices per copy of H. ↯

<u>Case 3:</u> C contains at least two mid vertices $m_{p,e}$ and $m_{q,f}$:

Because there are no edges between $m_{p,e}$ and $m_{q,f}$, they have two common neighbors. Each such neighbor is a start vertex or an end vertex since, again, there are no edges between mid vertices.

<u>Case 3.1:</u> p and q have the wildcard symbol at the same position i:

The only start vertex adjacent to both (and either) of $m_{p,e}$ and $m_{q,f}$ is s_i.

Case 3.1.1: p and q differ in some position (other than i):

No word matches both patterns. Because an end vertex's word must match the pattern of a mid vertex to be adjacent to it, no end vertex is adjacent to both $m_{p,e}$ and $m_{q,f}$.

Case 3.1.2: $p = q$ and $e \neq f$:

Since e and f have the same color, no vertex is incident to both of them in H. Because for an end vertex u_w to be adjacent to a mid vertex, the corresponding vertex u must be incident to the edge corresponding to the mid vertex in H, no end vertex is adjacent to both $m_{p,e}$ and $m_{q,f}$.

In conclusion, at most one vertex (the start vertex s_i) is adjacent to both $m_{p,e}$ and $m_{q,f}$. ↯

Case 3.2: p and q have the wildcard symbol at different positions:

Thus, $m_{p,e}$ and $m_{q,f}$ are adjacent to different start vertices.
Let u_w be an end vertex adjacent to both mid vertices. The word w matches both p and q. Because at each position, at least one of p and q has a non-wildcard symbol, w is uniquely determined (if it exists). Also, u is incident to both e and f in H. Since $e \neq f$, u is uniquely determined (if it exists). In summary, u_w is uniquely determined (if it exists).
In summary, there is at most one vertex adjacent to both $m_{p,e}$ and $m_{q,f}$. ↯

□

Claim 2: $\delta(G) \geq 2k$.

Proof of Claim 2. It holds $|V_H| \geq 5$ because H, being a k-regular graph with $k \geq 2$, contains a cycle, and has girth at least 5. Thus, each start vertex has $k^{k-1}\frac{|V_H|}{2} \geq k\frac{5}{2} \geq 2k$ neighbors. Each mid vertex has $2k+1$ neighbors, each end vertex $2k$. □

Claim 3: $\gamma(G) \geq 2k$.

Proof of Claim 3. Let S be a dominating set for G. Each of the $k^k|V_H|$ end vertices is in the closed neighborhood of a vertex of S.
The closed neighborhood of each start vertex does not contain any end vertices, the closed neighborhood of each mid vertex contains $2k$ end vertices and the closed neighborhood of each end vertex contains $k+1 \leq 2k$ end vertices. Thus, S contains at least $\frac{k^k|V_H|}{2k} \geq \frac{k^2 5}{2k} \geq 2k$ vertices. □

□

3.3 A nice improvement of $\mathrm{cc} \leq 2\,\mathrm{c}$ (for some graphs)

We prove that the bound $\mathrm{cc}(G) \leq 2\,\mathrm{c}(G)$ even holds for all connected graphs G when replacing $\mathrm{c}(G)$ by a different cop number $\mathrm{c}^*(G)$, which is smaller than $\mathrm{c}(G)$ for some graphs G.

Definition 3.6 (zugzwang cop number): For a connected graph G, we define $\mathrm{c}^*(G)$ as the smallest number of cops that can capture R in a game of Cops and Robber with the additional rule that R may never stand still.

Observe that $\mathrm{c}^*(G) \leq \mathrm{c}(G) \leq \mathrm{c}^*(G) + 1$ for every connected graph G. (The idea for the upper bound is for the cops to use one additional cop that, always moving to r, forces R to move at almost every robber turn.)

The zugzwang cop number is used by M. Maamoun and H. Meyniel [9], and a similar concept is used by S. Neufeld and R. Nowakowski [2]. (However, both do not define the zugzwang cop number explicitly).

Theorem 3.7 ($\mathbf{cc} \leq \mathbf{2\,c^*}$): For every connected graph G, it holds $\mathrm{cc}(G) \leq 2\,\mathrm{c}^*(G)$.

Proof. Let $k := \mathrm{c}^*(G)$, and let S be a strategy for the k cops to capture R in a game of Cops and Robber in G with the additional rule that R may never stand still. We give a strategy for $2k$ cops to capture R in a game of Cops and Attacking Robber:
We divide the cops into k pairs. Each pair shall take the place of one cop in S.
We also divide the cops into two teams so that each team contains exactly one cop of each pair.

At each cop turn, the strategy we give satisfies the following invariants:
- one of the teams (we call these cops **active**) is following S
- for each cop pair, the contained cops are on two adjacent vertices or on the same vertex

By the second invariant and because R does not move to protected vertices, R never eliminates a cop.

In the initial situation:

For each initial cop position v according to S, we let both cops of one pair of cops choose v as initial position.

Note that after R's first move, our invariants are satisfied.

Now we give the strategy for the remaining moves: We consider an arbitrary cop turn at which our invariants hold.

Case 1: R's last turn was R's initial turn, or R moved from one vertex to a different vertex in its last turn (in particular, R did not stand still):

The strategy S instructs each active cop C how to move.

Let each active cop C follow this instruction, and let the cop paired up with C move to the vertex on which C was at the beginning of the current turn.

Note that our invariants still hold.

Case 2: In R's last turn, R stood still:

Note that S cannot handle this situation.

We tell S that R moved from its current position r_0 to some vertex r' in R's last move. Now, S instructs each active cop C how to move.

Let each active cop C follow this instruction, and let the cop paired up with C move to the vertex on which C was at the beginning of the current turn.

<u>Case 2.1:</u> R stands still again:

We tell S that R moved from r' back to r_0. Now, S instructs each active cop C how to move.

Let each active cop C follow this instruction, and let the cop paired up with C move to the vertex on which C was at the beginning of the current turn.

Note that our invariants hold again.

<u>Case 2.2:</u> R moves to another vertex r_1:

We tell S that we lied when we told S that R moved to r', and that R moved to r_1 instead. Now, S gives us instructions for the positions that the active cops had at the beginning of their last turn. These are exactly the positions to which the non-active cops moved.

Let each non-active cop C follow the corresponding instruction from S, and let the active cop C' paired up with C move to the vertex on which C was at the beginning of the current turn (i.e. the vertex on which C' was before).

This means that the non-active cops become active and the active cops become non-active.

Note that our invariants hold again.

In both cases, our invariants hold again and we made progress in the strategy S. Thus, the cops capture R after finitely many such instructions.

□

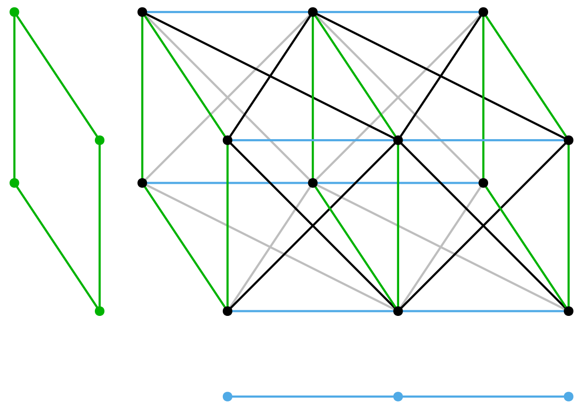

Figure 2: The graphs P_3 and C_4 and their strong product $P_3 \boxtimes C_4$, resembling a pipe.

4 Strong product

4.1 Introduction

Reflexive graphs are graphs in which every vertex v has a **loop**, i.e. an edge vv. In reflexive graphs, every move (in particular, standing still) that an entity can make corresponds to an edge.

To conveniently use this correspondence, in this section, we consider all graphs to be reflexive. For conciseness, we do not state this explicitly for every graph. We also omit loops in figures.

Definition 4.1: For two reflexive[6] graphs G and H, their **strong product** $G \boxtimes H$ is the graph with vertex set $V(G) \times V(H)$ and an edge between two vertices (g, h) and (g', h') if and only if there are edges gg' in G and hh' in H.

See Figure 2 for an example.

Note that the strong product of connected graphs is connected.

Until the next heading, let G and H be connected graphs.

For a vertex $v = (g, h)$ of $G \boxtimes H$, we use the notation $v.G := g$ and $v.H := h$ to get the position of v in one factor.

Note that the one-to-one correspondence between a move (even standing still) in a connected graph and a (for this purpose directed) edge in that graph implies a natural one-to-one correspondence between a move in $G \boxtimes H$ and a pair of a move in G and a move in H.

The previous observation leads to an alternative formulation of the Cops and (Attacking) Robber game in $G \boxtimes H$: Instead of a normal move, every entity (cop and robber) moves in G and in H simultaneously, and two entities are considered to be on the same vertex (for capturing, and eliminating in the attacking version) if and only if they are on the same vertex in G and in H.

Using the alternative formulation, we can see that under peaceful mechanics, capturing R in $G \boxtimes H$ is at least as hard as capturing R in G; in order to capture R in $G \boxtimes H$, a cop needs to capture R in both factors simultaneously.

Let us now look at attacking mechanics. The cops face the same problem as before, but it also

[6]The definition for non-reflexive graphs varies

gets harder for R to eliminate a cop. Thus, capturing R in $G \boxtimes H$ may be easier and may be harder for the cops compared to capturing R in G.

For convenience, we often describe a move or even a strategy in $G \boxtimes H$ with the moves or strategies in each factor.
We also often say "S in G" or "in G, S" for some statement S. This means that if we project the current state in $G \boxtimes H$ onto G, i.e. drop the information about each entity's position in H, the statement S holds for the resulting state in G.

S. Neufeld and R. Nowakowski [2] show $\mathrm{c}(G \boxtimes H) \leq \mathrm{c}(G) + \mathrm{c}(H) - 1$, and B. W. Sullivan and M. Werzanski [10] show that this upper bound is actually an equality:

Theorem 4.2 (cop number of $G \boxtimes H$) (S. Neufeld and R. Nowakowski [2], B. W. Sullivan and M. Werzanski [10]): For two connected graphs G and H, it holds $\mathrm{c}(G \boxtimes H) = \mathrm{c}(G) + \mathrm{c}(H) - 1$.

The main goal of this section is finding a similar upper bound as S. Neufeld and R. Nowakowski [2] for $\mathrm{cc}(G \boxtimes H)$ (if $\mathrm{c}(H) > 1$). Beforehand, we recommend to read their proof of the upper bound for $\mathrm{c}(G \boxtimes H)$, as we use similar ideas for our proof.
The rough proof idea for the new upper bound for $\mathrm{cc}(G \boxtimes H)$ is to partially switch roles between cops and robber: We show that if $\mathrm{c}(H) > 1$, then in any state in H, under peaceful mechanics, the cops can reach a state where almost all cops (more precisely, all cops except one) can avoid R indefinitely. Avoiding R (i.e. avoiding having the same position as R) in H, the cops can capture R in G without having to fear being eliminated by R. Thus, this only requires $\mathrm{c}(G)$ cops. After capturing R with sufficiently many cops in G, the cops can follow R (i.e. always move to r) in G whilst capturing R in H (as in the proof for the upper bound on $\mathrm{c}(G \boxtimes H)$).

4.2 Dismantlings

This section is mostly about how the cops can avoid R indefinitely (at least in most connected graphs) .
Even though we switch the roles of cops and robber later, we formulate our lemmas with the classical/natural role distribution for now.

Our first goal is to find many states in which R can avoid a cop indefinitely. We do this first for specific connected graphs, which do not contain so-called *pitfalls*, then we generalize our result to all connected graphs.

Definition 4.3: A vertex p of a connected graph G is called a **pitfall** if there is another vertex d so that $N[p] \subseteq N[d]$. We then say that d **dominates** p.
Removing p from G means removing p and every incident edge from G.

See Figure 3 for an example.

Note that for a connected graph G, removing a pitfall p from G does not disconnect G.

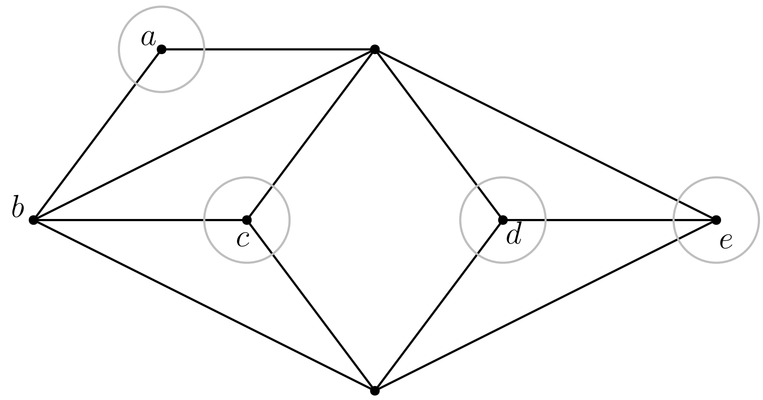

Figure 3: This graph has pitfalls a, c, d and e. For example, a is dominated by each vertex of $N[a] \setminus \{a\}$.

Lemma 4.4 (avoid (no pitfalls)): Let G be a connected graph without pitfalls. In every state in G with R and a cop C satisfying:
- it is R's turn
- $r \neq c$

under peaceful mechanics, R can avoid C indefinitely.

Proof. Since G does not contain pitfalls, c does not dominate r. Thus, there is at least one vertex v in $N[r] \setminus N[c]$.

Let R move to v.

In C's following move, C cannot capture R. Thus, after C's move, we have $r \neq c$ again. R can avoid C indefinitely by repeatedly applying this strategy. □

Note that even though Lemma 4.4 (avoid (no pitfalls)) can be applied to K_1, it simply gives us no state in which R can avoid being captured indefinitely.

Definition 4.5: We call a graph G' a **dismantling** of a connected graph G if G can be reduced to G' by repeatedly removing pitfalls and G' has no pitfalls.
We call the[7] corresponding sequence $((p_1, d_1), ..., (p_k, d_k))$ of pairs of a pitfall and a vertex dominating it (in the remaining graph) the corresponding **dismantling sequence**.

Note that every dismantling of a connected graph G is a connected induced subgraph of G. Also note that every connected graph has at least one dismantling.

Now we generalize the strategy from Lemma 4.4 (avoid (no pitfalls)) to all connected graphs G. More precisely, we generalize the strategy from Lemma 4.4 (avoid (no pitfalls)) from G's dismantling to G. The idea is that using a pitfall should not help the cop to capture R; the cop could just use the dominating vertex instead.

Definition 4.6: For a connected graph G with a pitfall p dominated by a vertex d, we define the function $(p \to d) : V(G) \to V(G - p), v \mapsto \begin{cases} v, v \neq p \\ d, v = p \end{cases}$.
For a connected graph G with a dismantling G' and a corresponding dismantling sequence $((p_1, d_1), ..., (p_k, d_k))$, we call the function $(p_k \to d_k) \circ ... \circ (p_1 \to d_1)$ the corresponding **dismantling function**.

[7] there may be multiple dismantling sequences because one pitfall may be dominated by multiple vertices

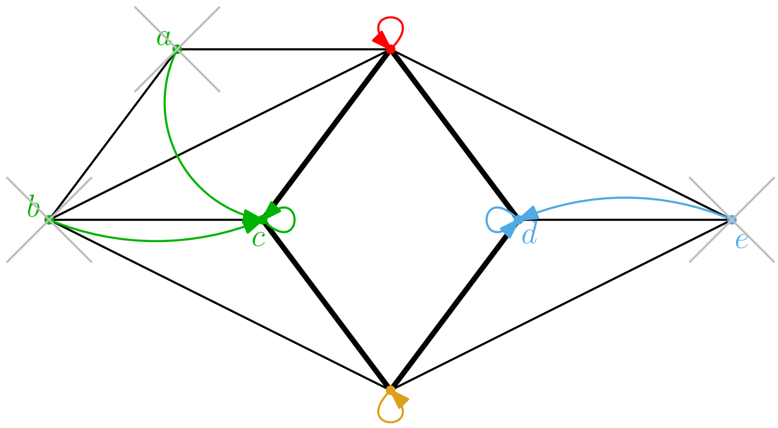

Figure 4: The graph from Figure 3 can be reduced to a dismantling C_4 by successively removing the vertices a, b and e. A corresponding dismantling sequence is $((a, b), (b, c), (e, d))$. The corresponding dismantling function is represented by arrows.

The notation $(p \to d)$ hides what G is; this has to be determined from the context.

See Figure 4 for an example.

Definition 4.7: A function f from the vertex set of a graph G to the vertex set of a graph H is called a **homomorphism** from G to H if for every edge uv of G there is an edge between $f(u)$ and $f(v)$ in H.

Definition 4.8: A graph homomorphism f from a graph G to a subgraph H of G is called a **retraction** from G to H if $f|_H = \text{id}$, i.e. $f(h) = h$ for every vertex h of H.

Note that in Definition 4.8, the graph H is an induced subgraph of G and connected if G is connected.

Lemma 4.9: For a connected graph G with a dismantling G' and a corresponding dismantling sequence $((p_1, d_1), ..., (p_k, d_k))$, the corresponding dismantling function is a retraction.

Proof. Observe that for every i, the function $(p_i \to d_i)$ is a homomorphism and a retraction from $G - p_1 - ... - p_{i-1}$ to $G - p_1 - ... - p_i$. Note that it is only a homomorphism because we consider graphs to be reflexive.
Also observe that for a retraction f from a graph A to a graph B and a retraction G from B to a graph C, the function $g \circ f$ is a homomorphism and a retraction from A to C.
Thus, as composition of the retractions $(p_1 \to d_1), ..., (p_k \to d_k)$, the dismantling function is a retraction. □

A retraction f from a connected graph G to a connected graph H can be used to generalize a strategy that works in H and for peaceful mechanics to a strategy that works in G:

Consider a state in G with an entity E of some party (cops or robber) and an entity O of the other party. Let E be in H. Let S be a strategy for E in a game in H and under peaceful mechanics against one entity of the other party.

We cannot apply S directly because O does not have to be or remain in H. However, we can pretend that O does:
While executing S, we pretend at all times that O is on $f(o)$ instead of o, i.e. always tell S $f(o)$ as O's current position.

At the beginning, the pretended opponent's position is $f(o)$ (and E's position is still e). Thus, S has to "work" in the state in H where E is on e and O is on $f(o)$ (Here it is important that E is on a vertex of H initially).

As the game goes on, S "works" against the pretended opponent because the pretended opponent acts in compliance with peaceful mechanics in H; it only moves along one edge of H per turn because f is a homomorphism (to H).

At each of E's turns, S instructs E how to move in H. E is able to make that move:
- every edge of H the is also an edge of G because f is a retraction
- we required that E starts on a vertex of H

The result of S (usually, that is that a set of states can be reached, or that a set of states can be avoided) applies not to O directly, but to the pretended opponent. However, that may already be useful.

This technique works the same if a party (i.e. the cops) consists of multiple entities.

We use this technique now to generalize the strategy from Lemma 4.4 (avoid (no pitfalls)) from a connected graph's dismantling to the graph itself:

Lemma 4.10 (avoid (with pitfalls)): Let G be a connected graph with a dismantling G' and a corresponding dismantling function f.
In every state in G with R and a cop C satisfying:
- it is R's turn
- $r \in G'$
- $r \neq f(c)$

under peaceful mechanics, R can avoid C indefinitely.

Proof. We give the desired strategy for R:
From now on, pretend that C is on $f(c)$ instead of c. In particular, at the beginning, C is not on r in the pretension. Because of that and $r \in G'$, by Lemma 4.4 (avoid (no pitfalls)), R can avoid C indefinitely in the pretension.

Let R avoid C indefinitely in the pretension.

If C ever captured R in reality, then $f(c) = f(r) = r$ would hold, i.e. C would capture R in the pretension, too. Thus, C does not capture R in reality. □

We have seen that when a cop C fails to capture R without using pitfalls, using the pitfalls does not help C. Now we show that using pitfalls does not help R to avoid C, too.

Lemma 4.11 (capture R from $f(r)$): Let G be a connected graph with a dismantling G' and a corresponding dismantling sequence $((p_1, d_1), ..., (p_k, d_k))$. Let f be the corresponding dismantling function.
In every state in G with R and a cop C satisfying:
- it is R's turn
- $c \in G'$
- $c = f(r)$

under peaceful mechanics, C can capture R.

Proof. We use the following Claim 1, which is a slightly stronger version of this Lemma for $k = 1$, basically stating that an additional pitfall does not make capturing R harder:

Claim 1: Let H be a connected graph with a pitfall p dominated by a vertex d. Let c be a vertex of $H - p$ and let r be a vertex of H.
If in the state in $H - p$ with a robber on $(p \to d)(r)$ and a cop on c where it is the robber's turn, under peaceful mechanics, the cop can capture the robber,
then in the state in H with a robber on r and a cop on c where it is the robber's turn, under peaceful mechanics, the cop can capture the robber.

Proof of Claim 1. We give a strategy for a cop C on c to capture a robber R on r: From now on, pretend that R is on $(p \to d)(r)$ instead of r (Recall that $(p \to d)$ is a retraction). In particular, R's pretended position is $(p \to d)(r)$ at the beginning, as the proposition's premise requires. By the premise, C can capture R in the pretension. Let C do that. Afterwards, it is R's turn and C is on the pretended position of R, i.e. $c = (p \to d)(r)$.
Now stop pretending that R is on $(p \to d)(r)$.

If $r \neq p$, we have $c = (p \to d)(r) = r$ by definition of $(p \to d)$ and are done.
If $r = p$, we know that $c = (p \to d)(r) = d$. Because $N[r] = N[p] \subseteq N[d] = N[c]$, C can capture R immediately after R's next move. ⬚

Because $((p_k \to d_k) \circ \ldots \circ (p_1 \to d_1))(r) = f(r) = c$: In the state in $G - p_1 - \ldots - p_k$ with a robber on $((p_k \to d_k) \circ \ldots \circ (p_1 \to d_1))(r)$ and a cop on c where it is the robber's turn, under peaceful mechanics, the cop can capture the robber.

Using the proposition, we get:
In the state in $G - p_1 - \ldots - p_{k-1}$ with a robber on $((p_{k-1} \to d_{k-1}) \circ \ldots \circ (p_1 \to d_1))(r)$ and a cop on c where it is the robber's turn, under peaceful mechanics, the cop can capture the robber.

Reapplying the proposition repeatedly gives us:
In the state in G with a robber on r and a cop on c where it is the robber's turn, under peaceful mechanics, the cop can capture the robber. □

Since pitfalls are neither really helpful for capturing nor for avoiding, it can be proven that removing them does not change the cop number of a connected graph, i.e. that every connected graphs have the same cop number as its dismantling(s). We only prove a weaker version that is sufficient for our purposes.

Lemma 4.12 (characterization of cop-win graphs) (R. Nowakowski and P. Winkler [5]): A connected graph G has cop number 1 if and only if it has K_1 as dismantling.

Proof. Let G' be a dismantling of G and let f be a corresponding dismantling function.

If $G' = K_1$, one cop C can capture $R \in a$ game of Cops and Robber in G:
Let C choose the vertex of G' as initial position. After R chose its initial position, it holds $f(r) = c$ because f maps every vertex to G'. By Lemma 4.11 (capture R from $f(r)$), C can capture R.

If $G' \neq K_1$, R can avoid one cop C indefinitely in a game of Cops and Robber in G:
First, let C chose its initial position.

Now we can assume that R already has a current position u but it still is R's turn: All R can do in this turn is moving to another vertex v. In reality, R can choose v as initial position directly. Note that this argument does not work with attacking mechanics.
Choose u from $V(G') \setminus \{f(c)\}$ (such vertex exists). By Lemma 4.10 (avoid (with pitfalls)), R can avoid C indefinitely. □

Putting our work together, we can give the desired strategy for the cops to start avoiding R. As we also use the strategy for something else later, we show a slightly different result.

Lemma 4.13 (disperse from $f(r)$): Let G be a connected graph with $\mathrm{c}(G) > 1$, a dismantling G' and a corresponding dismantling function f.
In every state in G with R and some cops satisfying:
- it is the cops' turn
- all cops are in G'
- there is at least one cop in $N_{G'}[f(r)]$,

under peaceful mechanics, the cops can reach[8] a state satisfying:
- it is the cops' turn
- there is a cop C so that $c \in N_{G'}[f(r)]$ and all other cops can avoid R indefinitely.

Proof. The main idea for the avoiding part of this lemma is that if the cops are in G', only the cops on $f(r)$ might not be able to avoid R indefinitely. If one of the cops on $f(r)$ moves away and the other cops remain on $f(r)$, $f(r)$ cannot remain on the same vertex as all of them.

If at the cops' turn, a cop C is on a vertex of G' and not on $f(r)$, we may ignore it and only give a strategy for the other cops to reach a desired state: While the other cops execute that strategy, C can execute a strategy to avoid R indefinitely (such strategy exists by Lemma 4.10 (avoid (with pitfalls))). At the end, a desired state (considering all cops) is reached.

We prove the statement with complete induction on the number k of cops:
Base: $k = 1$:

We are already done.

Step: $k \geq 2$:

Case 1: at most one cop is on $f(r)$:

Choose a cop in $N[f(r)]$, if possible on $f(r)$. We ignore all other cops. We are done.

Case 2: at least 2 cops are on $f(r)$:

We ignore all cops that are not on $f(r)$.
Let one (remaining) cop C move to a vertex in $N_{G'}(f(r)) \setminus \{f(r)\}$, while all other cops remain on $f(r)$. Such vertex exists by Lemma 4.12 (characterization of cop-win graphs).

Then, let R move. Let us denote R's previous position by r_0. Because f is a homomorphism, $f(r)$ is adjacent to $f(r_0)$.

We ignore C if $f(r) \neq c$, and another cop otherwise. Observe that in both cases, there is still a cop in $N_{G'}[f(r)]$.

[8] If we say "can reach", we mean that the desired state can be reached in some number of moves (not necessarily immediately).

Now the premise of our lemma holds, with at most $k - 1$ cops. We can apply the induction hypothesis.

□

4.3 The spreading attacking cop number

Recall that the rough idea for the proof of the upper bound for $\mathrm{cc}(G \boxtimes H)$ was to capture R in G (with multiple cops) while avoiding R in H, and to capture R in H afterwards while following R in G. Because in the first part, the cops avoid R in H, they cannot choose their positions in H when starting with the second part. Thus, they cannot directly execute any strategy to capture R with $\mathrm{cc}(H)$ cops in H.
However, with some extra work, the cops can make sure to be on the same vertex of H when starting with the second part. We will see that from such a state, only one extra cop may be required, compared to when the cops can choose their initial positions freely.

Definition 4.14: For a connected graph G, its *spreading attacking cop number* $\mathrm{cc}_{\text{spread}}(G)$ is the smallest number $k \in \mathbb{N}_+$ for which k cops can capture R in a game of Cops and Attacking Robber with the additional constraint that all cops must choose the same vertex as their initial position.

It can be seen that for any connected graph G, it holds $\mathrm{cc}(G) \leq \mathrm{cc}_{\text{spread}}(G) \leq \mathrm{cc}(G) + 1$: For the latter inequality, the $\mathrm{cc}(G) + 1$ cops can move to each initial position of a cop strategy for capturing R in the cc game successively, dropping off one cop at each initial position (which leaves two cops at the last visited initial position).

Note that if $\mathrm{cc}_{\text{spread}}(G) > 1$, then $\mathrm{cc}_{\text{spread}}(G)$ cops can capture R starting from any vertex.

4.4 An upper bound for $\mathrm{cc}(G \boxtimes H)$

Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$): For all connected graphs G and H with $\mathrm{c}(H) > 1$, we have $\mathrm{cc}(G \boxtimes H) \leq \mathrm{c}(G) + \mathrm{cc}_{\text{spread}}(H) - 1$.

Proof. Let G' be a dismantling of G with a corresponding dismantling function f. Let H' be a dismantling of H.

We describe a strategy for $\mathrm{c}(G) + \mathrm{cc}_{\text{spread}}(H) - 1$ cops to capture $R \in a$ game of Cops and Attacking Robber in $G \boxtimes H$:

We shall ensure that no cop gets captured before Phase 5. For that, we do not let any cop move to a vertex in $N[r] = N[r.G] \times N[r.H]$ *alone*, i.e. without another cop moving to the same vertex.

Phase 0

At the beginning of this phase, we have:
- the cops are to choose their initial positions

- choose a vertex v_0 of $G' \boxtimes H'$ arbitrarily and let all cops choose v_0 as their initial position
- then, let R choose its initial position

At the end of this phase, all cops on same vertex v_0 of $G' \boxtimes H'$. Also, either $f(r.G) = v_0.G$ or $f(r.G) \neq v_0.G$ holds.
If $f(r.G) = v_0.G$, then by Lemma 4.11 (capture R from $f(r)$), under peaceful mechanics, the

cops can capture R in G. This is done in Phase 05, with which our strategy continues in this case.

If $f(r.G) \neq v_0.G$, then by Lemma 4.10 (avoid (with pitfalls)), under peaceful mechanics, every cop can avoid R indefinitely in G. Note that in order to apply Lemma 4.10 (avoid (with pitfalls)), the roles of cop and robber need to be swapped. That is possible because attacking mechanics treat a cop and a robber the same unless one of them moves to the same vertex as an entity of the other type. Also note that if peaceful mechanics apply, the roles of cop and robber can always be swapped since peaceful mechanics always treat a cop and a robber the same.

Note that if every cop avoids R indefinitely in G, the cops can move arbitrarily in H without R being able to capture them in $G \boxtimes H$. This is used in Phase 1, with which our strategy continues in this case.

Phase 05

At the beginning of this phase, we have:
- all cops are on the same vertex v
- in G, under peaceful mechanics, one cop can capture R starting from $v.G$

The main goal of this phase is to capture R in G.

- in G, let the cops execute the same (deterministic) strategy to capture R with one cop
- meanwhile, in H, let the cops stand still

After the cops capture R in G, let R move.

All cops are always on the same vertex. Thus and because there are at least two cops (as $\mathrm{cc}_{\mathrm{spread}}(H) \geq \mathrm{c}(H) \geq 2$), R does not capture a cop (because R does not move to protected vertices).

At the end of this phase, we have:
- all cops are on same vertex v
- $v.G \in N[r.G]$, since the cops captured R

From here, our strategy continues with phase Phase 5.

Phase 1

At the beginning of this phase, we have:
- all cops are on the same vertex v_0 of $G' \boxtimes H'$
- all cops can avoid R indefinitely in G

The main goal of this phase is to enable $\mathrm{c}(G)$ cops to avoid R indefinitely in H.

In this phase, we distinguish two cases:

<u>Case 1:</u> $f(r.H) \neq v_0.H$:

- let the cops make no moves, i.e. directly continue with Phase 2

We know that in this case:
- all cops can avoid R indefinitely in H, by Lemma 4.10 (avoid (with pitfalls))
- all cops can avoid R indefinitely in G

<u>Case 2:</u> $f(r.H) = v_0.H$:

- in H, let the cops execute a strategy according to Lemma 4.13 (disperse from $f(r)$) (we can apply Lemma 4.13 (disperse from $f(r)$) because $\mathrm{c}(H) > 1$)
- meanwhile, in G, let each cop (continue to) avoid R indefinitely

Because every cop avoids R in G, R cannot capture a cop.

When the cops are finished with the strategy according to Lemma 4.13 (disperse from $f(r)$), we know:

- $\mathrm{c}(G) + \mathrm{cc}_{\text{spread}}(H) - 1 - 1 \geq \mathrm{c}(G) + 2 - 1 - 1 = \mathrm{c}(G)$ cops can avoid R indefinitely in H, by Lemma 4.13 (disperse from $f(r)$)
- all cops can avoid R indefinitely in G

At the end of this phase, we have (in both cases):

- $\mathrm{c}(G)$ cops can avoid R indefinitely in H
- all cops can avoid R indefinitely in G

Phase 2

At the beginning of this phase, we have:

- $\mathrm{c}(G)$ cops can avoid R indefinitely in H
- all cops can avoid R indefinitely in G

Let A be the subset of cops that can avoid R indefinitely in H, and let B be the set containing all other cops.

The main goal of this phase is that there is one cop $\hat{C}$ so that $\hat{c}.G \in N[r.G]$, whilst every other cop C can still avoid R indefinitely in G.

First, we show that we can reach our main goal when only considering G:

Lemma 4.16 (capture R with one cop): In every state in G with R and at least $\mathrm{c}(G)$ cops, under peaceful mechanics, the cops can reach a state satisfying:

- it is the cops' turn
- there is a cop $\hat{C}$ so that $\hat{C}$ is in $N[r]$ and all other cops can avoid R indefinitely.

Proof. The main idea is to let the cops remain in G' while more or less capturing R. Afterwards, starting in G', it is easy for most of the cops to avoid R indefinitely (we will use Lemma 4.13 (disperse from $f(r)$)). Using a strategy according to Lemma 4.11 (capture R from $f(r)$), a cop that more or less captured R can finish capturing R.

Note that in any state Q_0 in G with R and at least $\mathrm{c}(G)$ cops, under peaceful mechanics, the cops can capture R:
We can assume that there are exactly $\mathrm{c}(G)$ cops and let the other cops do anything.
There is a strategy S for $\mathrm{c}(G)$ cops to win a game of Cops and Robber in G. In Q_0, the cops can move to the initial positions of S (in multiple turns), and let R move one more time when they are done. We can tell S that the cops started as S instructed and that the robber chose r as its initial position. Now S tells the cops how to capture R.

We give a strategy for the cops:
First, let the cops move to any vertex of G' (in multiple turns).

Now let the cops execute a strategy to capture R. However, without telling the strategy, at each turn, let each cop, instead of moving to the vertex c as instructed, move to $f(c)$.
Observe that moving like this is possible; in the first move because $f|_{G'} = \text{id}$ and because f is a homomorphism, in the following moves because f is a homomorphism.
After that, there is a cop C with $f(r) = f(c) = c$, and all cops are in G'.

Let R move. Now, the conditions of Lemma 4.13 (disperse from $f(r)$) are met. Let the cops execute a strategy according to Lemma 4.13 (disperse from $f(r)$). After that, there is a cop $\hat{C}$ so that $\hat{c} \in N_{G'}[f(r)]$ and all other cops can avoid R indefinitely.

From now on, let all cops except $\hat{C}$ execute a strategy to avoid R indefinitely.

Let $\hat{C}$ move to $f(r)$. Now, by Lemma 4.11 (capture R from $f(r)$), C can capture R. □

- in G, let the cops from A execute a strategy according to Lemma 4.16 (capture R with one cop) (we can apply Lemma 4.16 (capture R with one cop) because $\mathrm{c}(G) > 1$; it can be seen that this follows from the fact that at the beginning of the current phase (Phase 2), all cops can avoid R indefinitely)
- meanwhile, in H, let every cop from A avoid R indefinitely
- meanwhile, in G, let every cop from B avoid R indefinitely
- meanwhile, in H, let the cops from B stand still

Because every cop from A avoids R indefinitely in H, and every cop from B avoids R indefinitely in G, R cannot capture a cop.

At the end of this phase, by Lemma 4.16 (capture R with one cop), there is a cop $\hat{C}$ so that $\hat{c}.G \in N[r.G]$ and all other cops from A can avoid R indefinitely in G.
In summary:
- there is a cop $\hat{C}$ so that $\hat{c}.G \in N[r.G]$ and
- all other cops can avoid R indefinitely in G

Phase 3
At the beginning of this phase, we have:
- there is a cop $\hat{C}$ so that $\hat{c}.G \in N[r.G]$ and
- all other cops can avoid R indefinitely in G

The main goal of this phase is that all cops are on $\hat{c}.H$ in H, whilst making sure that R cannot move into $N[\hat{c}.H]$ in H ("guarding" $\hat{c}.H$).

- in H, let every cop except $\hat{C}$ move to $\hat{c}.H$ (along a shortest path) and remain there until every cop C in on $\hat{c}.H$ in H
- meanwhile, in G, let every cop except $\hat{C}$ avoid R indefinitely in G
- meanwhile, in G, let $\hat{C}$ follow R, i.e. always move to $\hat{c}$
- meanwhile, in H, let $\hat{C}$ remain on the same vertex

Every cop $C \neq \hat{C}$ avoids R indefinitely in G. Thus, R cannot capture C.
At each of R's moves, R can only move to vertices in $N[\hat{c}.G]$ in G. Thus and because R does not move to protected vertices, R never moves into $N[\hat{c}.H]$ in H. Because $\hat{C}$ stands still in H, after $\hat{C}$'s move, $\hat{C}$ and R are still not adjacent in H. Thus, no matter how $\hat{C}$ moves in G, R cannot capture $\hat{C}$.

At the end of this phase:

- all cops are on the same vertex in H
- one cop $\hat{C}$ is in $N[r.G]$ in G

Phase 4

At the beginning of this phase, we have:

- all cops are on the same vertex $h \in H$
- at least one cop is in $N[r.G]$ in G

The main goal of this phase is to capture R in G with more cops. Because the cops continue to guard h, only $\mathrm{c}(G)$ cops are required to capture R in G (instead of $\mathrm{cc}_{\mathrm{spread}}(G)$ cops in the attacking setting). Thus, more cops can capture R in G.

Let A be the set of cops that are currently in $N[r.G]$ in G, and let B be the set of all other cops.

Until A has size $\mathrm{cc}_{\mathrm{spread}}(H)$:

- in G, let the cops from B, which are more than $\mathrm{c}(G) + \mathrm{cc}_{\mathrm{spread}}(H) - 1 - \mathrm{cc}_{\mathrm{spread}}(H) = \mathrm{c}(G) + \mathrm{cc}_{\mathrm{spread}}(H) - 1 - |A| = \mathrm{c}(G) - 1$, i.e. at least $\mathrm{c}(G)$, execute a Cops and Robber strategy to capture R
- meanwhile, in G, let every cop in A follow R
- meanwhile, in H, let every cop stand still

The set A of cops that follow R in G contains at least one cop at the beginning and never shrinks. Thus, by the same argument as in Phase 3, R does not capture a cop.

At the end of this phase:

- $\mathrm{cc}_{\mathrm{spread}}(H)$ cops are in $N[r.G]$ in G
- all cops are on the same vertex in H

Phase 5

At the beginning of this phase, we have:

- $\mathrm{cc}_{\mathrm{spread}}(H)$ cops $C_1, \ldots, C_{\mathrm{cc}_{\mathrm{spread}}(H)}$ are in $N[r.G]$ in G, and
- these cops are on the same vertex in H

- in H, let $C_1, \ldots, C_{\mathrm{cc}_{\mathrm{spread}}(H)}$ execute a spreading attacking Cops and Robber strategy to capture R
- meanwhile, in G, let $C_1, \ldots, C_{\mathrm{cc}_{\mathrm{spread}}(H)}$ follow R

Every elimination of a cop C_i in the game in $G \boxtimes H$ corresponds to an elimination in the game in H. When the cops capture R in the game in H, they also capture R in $G \boxtimes H$.

Note that in order to apply the spreading attacking Cops and Robber strategy, it is necessary that initially, all cops are on the same vertex in H.

At the end of this phase:

- R is captured

□

4.5 Tightness of the upper bound for $\mathrm{cc}(G \boxtimes H)$

Recall that above, we show that $\mathrm{cc}(G \boxtimes H) \leq \mathrm{c}(G) + \mathrm{cc}_{\mathrm{spread}}(H) - 1$ (Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$)) for all connected graphs G and H with $\mathrm{c}(H) > 1$.

This upper bound is tight in a weak sense; it cannot be improved by replacing $\mathrm{cc}_{\mathrm{spread}}(H)$ with $\mathrm{cc}(H)$, which corresponds to subtracting 1 sometimes (and 0 otherwise): The proposed new upper bound for $\mathrm{cc}(Q_3 \boxtimes Q_3)$ would be $\mathrm{c}(Q_3) + \mathrm{cc}(Q_3) - 1 = 2 + 2 - 1 = 3$ (Q_3 is the 3-dimensional hypercube graph). However, using an adaption[9] of the algorithm described by J. Petr, J. Portier, and L. Versteegen [12] to attacking mechanics, we can verify that $\mathrm{cc}(Q_3 \boxtimes Q_3) = 4$.

If we only consider graphs G with $\mathrm{c}(G) = 1$ as first factor in Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$), we obtain that $\mathrm{cc}(G \boxtimes H) \leq \mathrm{cc}_{\mathrm{spread}}(H)$ for all connected graphs G and H with $\mathrm{c}(G) = 1$ and $\mathrm{c}(H) > 1$.
This upper bound is tight in the same sense as before: If we replace $\mathrm{cc}_{\mathrm{spread}}(H)$ by $\mathrm{cc}(H)$ in the upper bound, we obtain $\mathrm{cc}(G \boxtimes H) \leq \mathrm{cc}(H)$ for all connected graphs G and H with $\mathrm{c}(G) = 1$ and $\mathrm{c}(H) > 1$. This is false for $G = P_4$ and $H = C_6$: It holds $\mathrm{cc}(H) = 2$, but using the same algorithm as above, we can verify that $\mathrm{cc}(P_4 \boxtimes C_6) = 3$.
In other words, for a connected graph H with $\mathrm{c}(H) = 1$, "multiplying" it with a connected graph G with cop number 1 may still increase its attacking cop number.

Unlike Theorem 4.2 (cop number of $G \boxtimes H$), the upper bound from Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$) is not always tight, not even if $G = H$, and thus the order of G and H in Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$) does not matter: $\mathrm{c}(C_5) + \mathrm{cc}_{\mathrm{spread}}(C_5) - 1 = 2 + 3 - 1 = 4$, and using the same algorithm as above, we can verify that $\mathrm{cc}(C_5 \boxtimes C_5) = 3$.

4.6 Lower bounds for $\mathrm{cc}(G \boxtimes H)$

We know that for all connected graphs G and H, we have $\mathrm{cc}(G \boxtimes H) \geq \mathrm{c}(G \boxtimes H) = \mathrm{c}(G) + \mathrm{c}(H) - 1$, which also implies $\mathrm{cc}(G \boxtimes H) \geq \mathrm{c}(G)$.
We are not aware of better lower bounds for $\mathrm{cc}(G \boxtimes H)$ that hold for all G and H. On the contrary, we can see that any lower bound for $\mathrm{cc}(G \boxtimes H)$ must sometimes evaluate to less than $\mathrm{cc}(G)$, i.e. under attacking mechanics, it sometimes requires fewer cops to capture R in $G \boxtimes H$ than to capture R in G:
For $k \in \mathbb{N}_+$, let G and H be connected graphs with $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 4k$ and $\mathrm{cc}(H) = 2\,\mathrm{c}(H) = 2k$ (as found in Theorem 3.5 (connected graphs with $\mathrm{cc} = 2\,\mathrm{c}$)), and let H' be $H \boxtimes C_4$.
First, observe that the proof of Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$) even shows that $\mathrm{cc}_{\mathrm{spread}}(G_1 \boxtimes G_2) \leq \mathrm{c}(G_1) + \mathrm{cc}_{\mathrm{spread}}(G_2) - 1$ for all connected graphs G_1 and G_2 with $\mathrm{c}(G_2) > 1$, since the given cop strategy lets all cops start on the same vertex.
Using this observation, Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$) and $\mathrm{cc}_{\mathrm{spread}}(C_4) = 2$, we obtain

$$\mathrm{cc}(G \boxtimes H') \leq \mathrm{c}(G) + \mathrm{cc}_{\mathrm{spread}}(H') - 1 \leq \mathrm{c}(G) + \mathrm{c}(H) + \mathrm{cc}_{\mathrm{spread}}(C_4) - 1 - 1 = 2k + k + 2 - 2 = 3k,$$

[9] To adapt the algorithm to attacking mechanics, we extend the set of states by allowing for each cop that it is eliminated instead of on a vertex. This does not affect the algorithm's asymptotic time complexity.

which is much smaller than $\mathrm{cc}(G) = 4k$.

4.7 Generalization to the strong product of more graphs

Let $G_1, \ldots, G_n$ be connected graphs. Observe that Theorem 4.2 (cop number of $G \boxtimes H$) implies $\mathrm{c}\left(\boxtimes_{i=1}^{n} G_i\right) = \sum_{i=1}^{n} (\mathrm{c}(G_i) - 1) + 1$. We generalize Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$)

to similar lower and upper bounds.

Note that the strong product of graphs is commutative and associative (we treat isomorphic graphs as equal, since the (attacking) cop number is invariant under isomorphism).

Corollary 4.17: For any $n \in \mathbb{N}_+$ connected graphs $G_1, \ldots, G_k, \ldots, G_n$ of which exactly the $k \in \mathbb{N}_+$ first ones have cop numbers greater than 1, we have

$$\sum_{i \leq k} (\mathrm{c}(G_i) - 1) + 1 \leq \mathrm{cc}\left(\boxtimes_i G_i\right) \leq \sum_{i \leq k} (\mathrm{c}(G_i) - 1) + 1 + \min_{i \leq k}\left(\mathrm{cc}_{\mathrm{spread}}(G_i) - \mathrm{c}(G_i)\right).$$

Proof. The lower bound follows from Theorem 4.2 (cop number of $G \boxtimes H$) and Observation 3.1 ($\mathrm{c} \leq \mathrm{cc} \leq 2\,\mathrm{c}$).

For the upper bound, choose $j \in [k]$ minimizing $\mathrm{cc}_{\mathrm{spread}}(G_j) - \mathrm{c}(G_j)$.
Observe that the strong product of graphs is commutative and associative (we treat isomorphic graphs as equal, since the (attacking) cop number is invariant under isomorphism).
Using this observation, Theorem 4.2 (cop number of $G \boxtimes H$) and then Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$), we get

$$\mathrm{cc}(\boxtimes_i G_i) = \mathrm{cc}\left(\left(\boxtimes_{i \neq j} G_i\right) \boxtimes G_j\right) \leq \mathrm{c}\left(\boxtimes_{i \neq j} G_i\right) + \mathrm{cc}_{\mathrm{spread}}(G_j) - 1 \leq \sum_{i \neq j} (\mathrm{c}(G_i) - 1) + 1 + \mathrm{cc}_{\mathrm{spread}}(G_j) - 1.$$

We can omit $\mathrm{c}(G_i) - 1$ for $i > k$ because it is 0 by the premise. □

We find it interesting that the difference between $\mathrm{cc}(\boxtimes_i G_i)$ and $\mathrm{c}(\boxtimes_i G_i)$ is bounded by the smallest $\mathrm{cc}_{\mathrm{spread}}(G_i) - \mathrm{c}(G_i)$ of all G_i with $\mathrm{c}(G_i) > 1$ – adding more factors to the product can only decrease this bound for the difference (however, it might increase the actual difference).

4.8 Cops and Attacking Robber is EXPTIME-complete

A graph together with a number k of cops form an instance of the decision problem Cops and (Attacking) Robber; the problem asks whether the k cops can capture R in a game of Cops and (Attacking) Robber in G.
EXPTIME is the set of decision problems that are solvable by a deterministic Turing machine in exponential time.

Since the algorithm for the non-attacking variant of this problem by A. Berarducci and B. Intrigila [11] runs in time $\mathcal{O}\left(|V(G)|^{2k+2}\right)$, Cops and Robber is in EXPTIME. As mentioned before, the algorithm (as well as the one from J. Petr, J. Portier, and L. Versteegen [12]) can easily be adapted to the attacking variant, without affecting the algorithm's asymptotic time complexity. Thus, Cops and Attacking Robber is in EXPTIME, as well.

Theorem 4.18 (W. B. Kinnersley [13]): Cops and Robber is EXPTIME-complete.

Due to its more complex mechanics, Cops and Attacking Robber seems to be at least as difficult as Cops and Robber. This guess is correct:

Theorem 4.19: Cops and Attacking Robber is EXPTIME-complete.

Proof. As mentioned above, Cops and Attacking Robber is in EXPTIME.

We give a polynomial-time reduction from Cops and Robber to Cops and Attacking Robber: Let (G, k) be a Cops and Robber instance. Let $G_1, ..., G_d$ be the connected components of G. Let $H := G \boxtimes C_4$. Note that $H = \uplus_i (G_i \boxtimes C_4)$, where $\uplus$ denotes the vertex-disjoint union of graphs. The pair $(H, k + d)$ is a Cops and Attacking Robber instance.

By Corollary 4.17, for each $i \in [d]$, it holds

$$\mathrm{c}(G_i) + 1 = \mathrm{c}(G_i) + \mathrm{c}(C_4) - 1 \leq \mathrm{cc}(G_i \boxtimes C_4) \leq \mathrm{c}(G_i) + \mathrm{cc}_{\text{spread}}(C_4) - 1 = \mathrm{c}(G_i) + 1.$$

Using $\mathrm{cc}(G_i \boxtimes C_4) = \mathrm{c}(G_i) + 1$ for each $i \in [d]$, we find

$$\mathrm{cc}(H) = \mathrm{cc}(G \boxtimes C_4) = \sum_i \mathrm{cc}(G_i \boxtimes C_4) = \sum_i (\mathrm{c}(G_i) + 1) = \sum_i \mathrm{c}(G_i) + d = \mathrm{c}(G) + d.$$

Thus, (G, k) is a YES-instance for Cops and Robber if and only if $(H, k + d)$ is a YES-instance for Cops and Attacking Robber. Note that H can be constructed from G in polynomial time. □

5 Eliminated cops

5.1 Introduction

In this section, we investigate the following number:

Definition 5.1 (elimination number): For a connected graph G and a number $k \geq \mathrm{cc}(G)$, let $\mathrm{elim}_k(G)$ be the maximum number of cops that R can eliminate before being captured (if R is ever captured)[10] when R plays a game of Cops and Attacking Robber against k cops in G.
We also define $\mathrm{elim}(G) := \mathrm{elim}_{\mathrm{cc}(G)}(G)$.

In other words, $\mathrm{elim}_k(G)$ is the largest number $x \in \mathbb{N}_0$ for which there is a strategy for R for a game of Cops and Attacking Robber against k cops in G that guarantees that R either eliminates x cops before being captured or avoids being captured indefinitely.

To gain intuition, we recommend verifying that $\mathrm{elim}(C_l) = \begin{cases} 0, l \leq 9 \\ 1, l > 9 \end{cases}$ for every $l \geq 3$.

In this section, we always assume attacking mechanics.

Note that in any given (maybe initial) state in which the cops can capture R, R can eliminate some number of cops before being captured if and only if the cops cannot capture R without loosing that number of cops. In other words: The largest number of cops that R can eliminate before being captured equals the smallest number of cops for which the cops can capture R without loosing more cops.

Imagine that a police department wants to use their available cops to capture robbers multiple times. For this, it is beneficial to loose few cops. On the other hand, observe that for fixed G, $\mathrm{elim}_k(G)$ is monotonically decreasing in k. This means that in order to loose fewer cops, the police department needs to assign more cops to capturing R.

5.2 First observations

In this section, let us fix a connected graph G.

Observe that for all $k \geq \mathrm{cc}(G)$, it holds $\mathrm{elim}_k(G) = 0$ if and only if k is at least the domination number $\gamma(G)$. However, $\gamma(G)$ can be much larger than $\mathrm{c}(G)$ and $\mathrm{cc}(G)$, and assigning $\gamma(G)$ cops to capturing one robber in G may be impractical or impossible for the police department. If the department is ok with sacrificing a single cop, it can use the bound $\mathrm{elim}_{2\,\mathrm{c}^*(G)}(G) \leq 1$, which follows from the proof of Theorem 3.7 ($\mathrm{cc} \leq 2\,\mathrm{c}^*$). This means that with at most twice as many cops as required to capture R at all, the cops can make sure to loose at most one cop while capturing R.
If G is bipartite, the cops can even make sure to loose at most one cop with only two more cops than required to capture R; formally, $\mathrm{elim}_{\mathrm{c}(G)+2}(G) \leq 1$ if G is bipartite. It can be seen that this follows from A. Bonato *et al.* [20]'s proof of the fact that $\mathrm{cc}(G) \leq \mathrm{c}(G) + 2$ if G is bipartite.

As a preparation for the next section(s), observe the following:

[10]From now on, for conciseness, we omit stating this parenthesized part explicitly.

Observation 5.2 (eliminating unprotected cops): In any given state (maybe initial) state, R can eliminate $x \geq 2$ cops before being captured if and only if R can eliminate $x-1$ unprotected cops before being captured.

5.3 Games with $\operatorname{cc}(G)$ cops

In this section, we investigate $\operatorname{elim}(G) = \operatorname{elim}_{\operatorname{cc}(G)}(G)$, i.e. games where the police department uses only the required amount of cops to capture R.
In this scenario, the police department might be able to loose only one cop; by the observation that $\operatorname{elim}_{2\operatorname{c}(G)} \leq 1$, all connected graphs G with $\operatorname{cc}(G) = 2\operatorname{c}(G)$ have $\operatorname{elim}(G) = \operatorname{elim}_{2\operatorname{c}(G)}(G) \leq 1$.
On the other hand, the police department might loose all cops but one. We prove this in the remainder of this section.

The proof is structured as follows:

First, for each $k \geq 3$, we construct a connected graph G_k with attacking cop number k in which R can more or less eliminate exactly one unprotected cop before being captured.

To construct a connected graph G with $\operatorname{cc}(G) = k$ and $\operatorname{elim}(G) = k-1$, we start with G_k. In G_k, R can eliminate one unprotected cop before being captured. Note that directly after eliminating an unprotected cop, R is not captured by the cops. We give R the opportunity to continue the game in G_{k-1} at this moment, by adding a copy of G_{k-1} to G_k and identifying R's current position with some special vertex of G_{k-1}. (Because we cannot predict R's current position, we just do it for every vertex of G_k.) Taking said opportunity, R can eliminate another unprotected cop. We repeat this with G_{k-2} and smaller graphs. Passing through the games in each G_i, R can eliminate all cops but one.

After switching between games from G_k to G_{k-1}, R has to assume that the cops are right behind R. The strategy for R for the game after the switch needs to handle this special starting situation.
On the other hand, the cops have to watch out not to let R move back to G_k again, since they already lost a cop and might not be able to capture R in G_k anymore.
These considerations are reflected in the properties of G_k that we prove:

Lemma 5.3: For every $k \geq 3$, there is a connected graph G_k with two adjacent vertices c_0 and r_0 so that in every state in G_k with k cops, all on c_0, and R:
(1) if it is R's turn and R is on r_0, then R can eliminate an unprotected cop before being captured
(2) if it is the cops' turn, the cops can reach a state where
- it is R's turn
- R is captured
- R eliminated at most one unprotected cop
- R was not on c_0 at any previous robber turn

The last condition in (2) means that the desired strategy must guarantee that the cops capture R immediately if R ever moves to c_0. Equivalently: The desired cop strategy must guarantee that c_0 is protected at all cop turns at which r is adjacent to c_0.

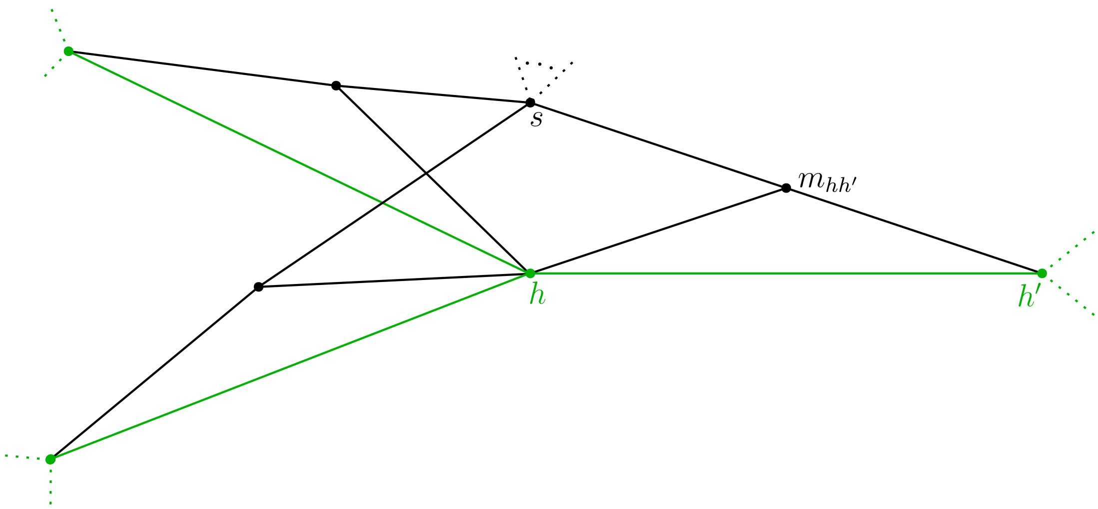

Figure 5: (A part of) the graph G_3 with some vertex h of H and another vertex $h' \in N_H(h)$. Green vertices and edges correspond to vertices and edges of H.

Proof. We construct the desired graph G_k from a k-regular connected graph H with girth at least 5 (we found such H in Lemma 3.4):

- add a vertex c_0
- for each edge $e = uv$ of H, add a *mid vertex* m_e that is adjacent to c_0, u and v

Let r_0 be an arbitrary mid vertex m_{e_0}.

G_3 is depicted in Figure 5.

We show that G_k has the desired properties:

For (1): We give the desired strategy for R:

Let R move to an arbitrary endpoint of e_0.

Now, R is on a vertex of H and no cop is on r.

Until every vertex of $N_H[r]$ is protected, always let R move to an arbitrary unprotected vertex of $N_H[r]$.

Note that R is on a vertex of H and has not been captured.

Every cop C can protect at most one vertex of $N_H(r)$ because $c \neq r$ and H has girth at least 5. By the pigeonhole principle, every cop protects exactly one vertex of $N_H(r)$.
Let $\hat{C}$ be a cop that protects r, and let h be the vertex of $N_H(r)$ that $\hat{C}$ protects.
$\hat{C}$ is not protected by another cop C: Otherwise, due to the structure of G_k, C could not protect a vertex of $N_H(r)$ other than h.

Let R eliminate $\hat{C}$.

We are done.

For (2): We give the desired strategy for the cops:

Since R is not on a protected vertex, R is on a vertex of H.

Let one cop move to each of the mid vertices adjacent to r.

At R's turn, every vertex in $N_H[r]$ is protected. Thus and since R does not move to a protected vertex, R moves to some vertex that is not in $N_H[r]$, i.e. to $c^\dagger$ for some cop $C^\dagger$.
There are at least two cops C and $\hat{C}$ left.

Let $\hat{C}$ move to the vertex on which R was at the beginning.

Observe that at R's turn, every vertex in $N[r]$ is protected. This means that R cannot eliminate another unprotected cop at R's turn, and that afterwards, a cop is in $N[r]$.

Let some cop in $N[r]$ capture R.

Note that throughout the whole strategy, R could never move to c_0 unless c_0 was protected. □

As described before, we now compose the graphs we found in Lemma 5.3 to create graphs in which R can eliminate all cops but one before being captured:

Theorem 5.4 (eliminate cc —1 cops): For every $k \in \mathbb{N}_+$, there is a connected graph G with $\mathrm{cc}(G) = k$ and $\mathrm{elim}(G) = k - 1$.

Proof. For $k = 1$, all graphs with attacking cop number 1 satisfy the conditions. Now let $k \geq 2$.

We first construct graphs F_k that have the desired properties only in certain states:

Claim 1: For every $k \geq 2$, there is a connected graph F_k with two adjacent vertices c_0 and r_0 so that in every state in F_k with k cops, all on c_0, and R:
(1) if it is R's turn and R is on r_0, then R can eliminate $k - 1$ cops before being captured
(2) if it is the cops' turn, the cops can reach a state where
- it is R's turn
- R is captured
- R was not on c_0 at any previous robber turn

Proof of Claim 1. We use induction:
Base: $k = 2$:
We choose $F_2 := C_4$ with c_0 and r_0 as two adjacent vertices of F_2.
Step: $k > 2$:
Let G_k with vertices r_0 and c_0 as in Lemma 5.3. We construct the desired graph F_k from G_k as follows:
Let F_{k-1} with vertices c_1 (originally c_0) and r_1 (originally r_0) as in the induction hypothesis. For each vertex g of G_k, add a copy F_g of F_{k-1} and identify the vertices g of G_k and c_1 of F_g.

Note that F_k is connected.

For (1):
We give the desired strategy for R:

From now on, let R pretend that every cop that is in F_g for some g is on g instead.
Currently, in the pretension, every cop is on c_0.

Let R execute a strategy according to (1) in Lemma 5.3 against the pretended cops in G_k, until R eliminated an unprotected cop (that is possibly indefinitely).

In the pretension, every cop acts in compliance with the attacking mechanics:

- If in reality, a cop C moves inside some copy of F_{k-1}, the cop stands still in the pretension. If the cop moves in G_k in reality, it does so in the pretension as well. In both cases, C makes a valid move inside G_k.
- Whenever R eliminates a cop C in the pretension, R eliminates C in reality as well: In the pretension, C was in $N(r)$ directly before the elimination but not in $N(r)$ at C's preceding turn; otherwise, C could have captured R at C's preceding turn, in contradiction the guarantee of R's strategy (Recall that we only let R execute the strategy until R eliminated an unprotected cop).
  The only way that C could have moved into $N(r)$ (from somewhere else) in the pretension is that C moved along an edge of G_k in reality. Thus, C's positions in reality and in the pretension are the same, which means that R eliminates C in reality as well.

Note that if R was ever captured in reality, R would be captured in the pretension as well, in contradiction to Lemma 5.3.

After R eliminated an unprotected cop in the pretension (which might never happen) and the following cop turn, we know:

- it is R's turn
- there are exactly $k - 1$ cops
- in reality, no cop is in F_r (because in the pretension, R eliminated an unprotected cop in its last move)

Now we let R stop pretending.

Let us fix the copy $F := F_r$ of F_{k-1} and let r_1 and c_1 be the (adjacent) vertices r_1 and c_1 of that copy. Currently, $r = c_1$.

Let R move to r_1.

Now:

- it is R's turn
- R is on r_1
- there are exactly $k - 1$ cops
- no cop is in $V(F) \setminus \{c_1\}$

We let R focus on F, with the same technique as before:

From now on, let R pretend that every cop that is not in F is on c_1.
Currently, in the pretension, every cop is on c_1.

Let R execute a strategy according to the induction hypothesis (i.e. (1) in Claim 1) against the pretended cops in F, i.e. a strategy to eliminate $(k - 1) - 1$ cops before being captured (if ever).

In the pretension, every cop acts in compliance with the attacking mechanics, by the same arguments as before (in particular, every elimination in the pretension corresponds to an elimination in reality).

If R is captured in reality, R is captured in the pretension as well (by the same argument as before). Thus, R must have eliminated $(k-1)-1$ cops already (in pretension and in reality). Together with the previous elimination in G_k, these are $k-1$ eliminations.

For (2):
We give the desired strategy for the cops:

From now on, let the cops pretend that when R is in F_g for some g, R is on g instead. Currently, in the pretension, R is on any vertex of G_k.

Let the cops execute a strategy according to (2) in Lemma 5.3 against the pretended robber in G_k, until the cops can reach a desired state immediately (in the pretension).

In the pretension, R acts in compliance with the (attacking) mechanics: No matter how R moves in reality, in the pretension, R makes a valid move inside G_k.

In reality, R eliminated at most one cop:
- Whenever R eliminates a cop C in reality, R also eliminates C in the pretension: If R moves to a vertex c of G_k in reality, R moves to c in the pretension as well.
  Thus, by Lemma 5.3, R eliminated at most one unprotected cop (in reality).
- Since R never moves to protected vertices (in reality), R did not eliminate a protected cop in reality.

If R ever moved to c_0 in reality, R would have moved to c_0 in the pretension as well. By Lemma 5.3, the cops could capture R immediately in the pretension. Since $r = c_0$ is in G_k, the cops could capture R immediately in reality as well. Since R never moves to protected vertices, this never happens.

Now we let the cops stop pretending. Let g be the vertex of G_k so that R is in F_g. We have:
- it is the cops' turn
- there is a cop $\hat{C}$ in $N_{G_k}[g]$ (that has not been eliminated)
- there are at least $k-1 \geq 2$ cops

Since R does not move to a protected vertex, $\hat{c} \notin N[r]$.

Let $F := F_g$. As long as $\hat{C}$ stands still, because RR never moves to protected vertices, R cannot move to g and thus remains in F.

Let every cop $C \neq \hat{C}$ move to $\hat{c}$ along any shortest path.
When all cops arrived, let all cops move to g.

Now:
- it is the cops' turn
- there are at least $k-1$ cops
- R is in F
- all cops are on g

Recall that g is identified with the vertex c_1 (originally c_0) of the copy F_g of F_{k-1}. As long as the cops (and the fact that R does not move to a protected vertex) prohibit R from moving to g, R remains in F.

Let $k-1$ cops execute a strategy according to our inductive hypothesis in F

Now:
- R is captured

Note that R was never able to move to c_0 unless c_0 was protected.

□

F_k from Claim 1 almost has the desired properties from Theorem 5.4 (eliminate cc −1 cops):
- (2) in Claim 1 implies that $\mathrm{cc}(F_k) \leq k$
- in a certain state with R and k cops, R can eliminate $k-1$ cops before being captured

However, we need that in *every* initial state with k cops, R can eliminate $k-1$ cops before being captured.

To fix this, we create a larger graph containing $k+1$ copies of F_k so that no matter which initial positions the cops choose, there is still a copy of F_k in which (roughly) no cop is. To be precise, we create the desired graph G by taking $k+1$ copies of F_k and identifying their c_0's. We call the emerging vertex c_0. Note that G is connected.

It remains to prove that G has the desired properties.

Claim 2: In every initial state in G with k cops, R can eliminate $k-1$ cops before being captured, i.e. $\mathrm{elim}_k(G) \geq k-1$.

Proof of Claim 2. We describe the desired strategy for R: By the pigeonhole principle, there is one copy F of F_k so that there are no cops in $F - c_0$. Choose the r_0 from that F as initial position. From now on, pretend that every cop that is not in F is on c_0. In particular, all cops are on c_0 in the beginning. Now execute a strategy according to Claim 1 against the pretended cops in F. Observe that by the same arguments as in the inductive proof of (1) in Claim 1, the pretended cops act in compliance with the attacking mechanics, and R being captured in reality implies R being captured in pretension as well. The guarantees from Claim 1 convey to reality: R can avoid being captured indefinitely or eliminate $k-1$ (pretended and real) cops. □

Claim 3: $\mathrm{cc}(G) \leq k$.

Proof of Claim 3. The cop strategy for F_k from Claim 1 only needs slight adjustment to work in G: Let all k cops choose c_0 as initial position. After R chose its initial position in some copy F of F_k, let the cops execute a strategy according to Claim 1 in F. By Claim 1 and because R does not move to a protected vertex, R never moves to c_0 and thus remains in F. By Claim 1, the cops capture R. □

Claim 4: $\mathrm{cc}(G) \geq k$.

Proof of Claim 4. Note that $\mathrm{elim}_i(H) \leq i-1$ for every connected graph H and every $i \geq \mathrm{cc}(H)$.
Recall that for every connected graph H, $\mathrm{elim}_i(H)$ is monotonically decreasing in i.
Applying these observations gives us

$$k-1 \overset{\text{Claim 2}}{\leq} \mathrm{elim}_k(G) \overset{\text{Claim 3},\ \mathrm{elim}_i(G)\searrow}{\leq} \mathrm{elim}_{\mathrm{cc}(G)}(G) \overset{\mathrm{elim}_i(G)\leq i-1}{\leq} \mathrm{cc}(G)-1.$$

The combination of Claim 2, Claim 3 and Claim 4 is what we had to show. □

We remark that assigning one additional cop to capture a robber may save arbitrarily many cops; formally, for each Δ there is a connected graph G and some $k \geq \mathrm{cc}(G)$ so that the difference $\mathrm{elim}_{k+1}(G) - \mathrm{elim}_{k}(G)$ is at least Δ:
Let $G_{\Delta+2}$ be the graph from the proof of Lemma 5.3. Adding one additional cop on s that captures R as soon as R moves to a mid vertex makes sure that no unprotected cop is eliminated. It can be seen that this strategy can be carried over to the final graph in Theorem 5.4 (eliminate cc -1 cops), resulting in a cop strategy that looses at most one cop.

5.4 Games with $\mathrm{cc}(G)$ cops in graphs with higher girth

One might conjecture that the proof of Theorem 5.4 (eliminate cc -1 cops) strongly relies on the triangles in the constructed graphs, and that the cops can make sure to loose fewer cops in connected graphs with higher girth.

This conjecture is false for connected graphs with attacking cop number at most 2; for example, P_7 is a connected graph with $\mathrm{cc}(G) \leq 2$, with girth ∞ and with $\mathrm{elim}(P_7) = \mathrm{cc}(P_7) - 1$.

But if we only consider connected graphs with attacking cop number at least three, the conjecture is partially correct: We prove that R can eliminate at most all cops but two in connected graphs with girth at least 5 and attacking cop number at least 3. However, this bound is already tight: We find connected graphs with arbitrarily large girth in which R can eliminate all but two cops before being captured.
In summary, cycles of length less than 5 are only necessary for R to eliminate one more cop.

Although we have tight results for connected graphs with girth at least 3 and for connected graphs with girth at least d for all $d \geq 5$, we do not know whether for each $k \geq 3$, there is a connected graph G with girth at least 4 and $\mathrm{cc}(G) = k$ in which R can eliminate $k - 1$ cops before being captured.

Our first goal in this section is to prove that R can eliminate at most all cops but two in connected graphs with girth at least 5 and attacking cop number at least 3.

Intuitively, the reason for this that if the cops loose all but two cops, the two cops are too weak to capture R because of the large girth. To be able to exploit this weakness, R must make sure to always have two different directions in which R can move. This is possible if R always moves inside the *2-core* of our given graph:

Definition 5.5: The **2-core** of a graph G is the graph that contains no vertices of degree at most 1 that is obtained by repeatedly removing vertices of degree at most 1 from G.

Observe that the 2-core of a graph G is unique and contains exactly those vertices of G that are part of a cycle in G. Note that the 2-core of a connected graph is connected.

Observe that every connected graph that contains a cycle is the (edge- and vertex-) disjoint union of its 2-core, some trees, and for each tree exactly one edge that connects the tree to the 2-core.

First, we restrict the graphs we consider to those that do not have vertices of degree at most 1, or in other words, graphs that are their own 2-core. For convenience, instead of investigating the number of cops R can eliminate before being captured, we investigate the number of unprotected cops that R can eliminate being captured. Recall that by Observation 5.2 (eliminating unprotected cops), this does not really make a difference.

Lemma 5.6: Let G be a connected graph with girth at least 5 and $\mathrm{cc}(G) \geq 3$ that has no vertex of degree at most 1. Let S be a strategy for $k \in \mathbb{N}_+$ cops to capture R in a game of Cops and Attacking Robber in G.
Then the strategy S guarantees that the cops capture R with at most $k-3$ unprotected cops being eliminated.

Proof. Let us assume the opposite, i.e. that R has a strategy to eliminate $k-2$ cops before being captured in a game of Cops and Attacking Robber against k cops executing S in G. We extend this strategy for R to a strategy to avoid the cops executing S indefinitely, by describing how, in case R ever eliminates $k-2$ unprotected cops, R shall act afterwards: After R ever eliminates the $k-2$-th unprotected cop $C^\dagger$, let the cops move. They do not capture R because $C^\dagger$ was unprotected.

Now:
- it is R's turn
- R is not eliminated
- there are exactly 2 cops left

G has minimum degree 2 and does not contain C_4 as a subgraph.

Exactly as in the proof of Lemma 3.2 (lower bound for cc when no C_4), in this and all future robber turns, an unprotected vertex in $N(r)$ can be found, to which we let R move. This means that R is never captured. □

Theorem 5.7: For every connected graph G with girth at least 5 and $\mathrm{cc}(G) \geq 3$, it holds $\mathrm{elim}(G) \leq \mathrm{cc}(G) - 2$.

Proof. Let G' be the 2-core of G.
The graph G contains a cycle because $\mathrm{cc}(G) > 2$ and every tree T has $\mathrm{cc}(T) \leq 2\,\mathrm{c}(T) \leq 2$. Thus, G' contains at least one vertex.

Recall that G is the disjoint union of G', some trees $T_1, \ldots, T_l$, and for each tree T_i exactly one edge from a vertex of T_i to a vertex v_i of G'.

Let S be a strategy for $\mathrm{cc}(G)$ cops to capture R in a game of Cops and Attacking Robber in G.

Intuitively, we convey S to a "decomposed" strategy that first "works" in G' and then captures R if R is in some tree T_i.
First, we retract S to work in G'. Afterwards, we extend the resulting strategy to G again.

We give a strategy S' for $\mathrm{cc}(G)$ cops to capture R in a game of Cops and Attacking Robber in G':

Let the cops execute S. If a cop C is instructed to move to a vertex of a tree T_i (or to choose a vertex of T_i as initial position), let C move to the vertex v_i (or choose v_i as initial position) instead (but pretend to S that C followed the original instruction).

Note that each cop can always act as we described (i.e. we never instruct a cop C to move to a vertex that is not in $N[c]$).

If R eliminates a cop $C^\dagger$, it looks for S like R eliminates $C^\dagger$, as well: At the previous cop turn, r was not on a vertex $v \in N\left[c^\dagger\right]$ in reality since $c^\dagger$ would have protected v. However, at R's turn, $C^\dagger$ was in $N[r]$ in reality. Thus, S told $C^\dagger$ to move along an edge of G', which means that $C^\dagger$ was in G' afterwards.

When the cops capture R in the pretension, they capture R in reality, as well. This concludes our strategy S'.

By Lemma 5.6, the strategy S' we describe above guarantees that the cops capture R with at most $\mathrm{cc}(G) - 3$ unprotected cops being eliminated.
We extend this strategy to G again, i.e. we give a strategy for $\mathrm{cc}(G)$ cops for a game of Cops and Attacking Robber in G to capture R with at most $\mathrm{cc}(G) - 3$ unprotected cops being eliminated:

From now on, let the cops pretend that if R is in a tree T_i, R is on v_i instead.

Let the cops execute S' against the pretended robber in G', until a cop can capture the robber immediately in the pretension.

As for the first part of the cop strategy we described in the proof of (2) of Claim 1 in the proof of Theorem 5.4 (eliminate cc -1 cops), R acts in compliance with the (attacking) mechanics in the pretension, and R eliminated at most $\mathrm{cc}(G) - 3$ cops in reality.

Now we let the cops stop pretending. We have:
- it is the cops' turn
- there are at least three cops, all in G'
- R is in a tree T_i (since R is not on a protected vertex)
- there is a cop $\hat{C} \in N_{G'}[v_i]$ (that has not been eliminated)

Let one cop $C \neq \hat{C}$ move to $\hat{C}$ on a shortest path. Then, let $\hat{C}$ and C move to r until R is captured.

Since T_i is a tree and only connected to G' via v_i, the cops $\hat{C}$ and C capture R eventually, with at most one (protected) cop being eliminated directly before capturing R.

In summary, R eliminated at most $\mathrm{cc}(G) - 2$ unprotected cops. □

Our next goal is to prove that R can eliminate all but two cops in some connected graphs with arbitrarily large girth. We proceed similar to the proof of this statement without girth restrictions: First, for each $k \geq 4$, we find connected graphs with attacking cop number k and large girth in which R can more or less eliminate an unprotected cop. Then we compose these graphs to create graphs in which R can eliminate all cops but two before being captured.

Lemma 5.8: For every $k \geq 4$ and every $d \geq 3$, there is a connected graph G_k with girth in $[d, \infty)$ and with two adjacent vertices c_0 and r_0 so that in every state in G_k with k cops, all on c_0, and R:
(1) if it is R's turn and R is on r_0, then R can eliminate an unprotected cop before being captured
(2) if it is the cops' turn, the cops can reach a state where
- it is R's turn
- R is captured
- R eliminated at most one unprotected cop
- R was not on c_0 at any previous robber turn

Proof of Lemma 5.8. We reuse the graph from the proof of Lemma 5.3 and just stretch all edges by (more or less) factor d to make cycles longer. A move sequence in the original graph can be (more or less) carried over into the stretched graph (with one move being divided into more or less d new moves). The problem is that not every move sequence in the stretched graph corresponds to a move sequence in the original graph. Thus, showing the same properties for the stretched graph is a lot more complicated than for the original graph.

Let $H = (V_H, E_H)$ be a k-regular connected graph with girth at least 5 (we found such graphs in Lemma 3.4). We create G_k from H:
- replace every original edge of H with a new path of length $2d - 1$
- add a vertex s
- for each edge e=uv of H, add a **mid vertex** m_e with a new path of length d to u, a new path of length d to v and a new path of length d to s

Let c_0 be s and let r_0 be some vertex adjacent to s.

G_4 is depicted in Figure 6.

Note that G_k is connected and has a girth in $[d, \infty)$.

For (1): We give the desired strategy for R:

Let R move to a vertex h of H that is closest to r.

Now:
- R is on a vertex of H
- R has not been captured

Let R stand still until a cop is in $N(r)$ (that may already be the case).

Let $\hat{C}$ be a cop in $N(r)$.
Case 1: $\hat{C}$ is not protected:

Let R eliminate $\hat{C}$.

We are done.

Case 2: $\hat{C}$ is protected:

Observe that for every cop C, there is at most one vertex of $N_H(r)$ to which c has distance at most $2d - 1$. Let C' be a cop that protects $\hat{C}$. Note that $\hat{C}$ and C' have distance at most

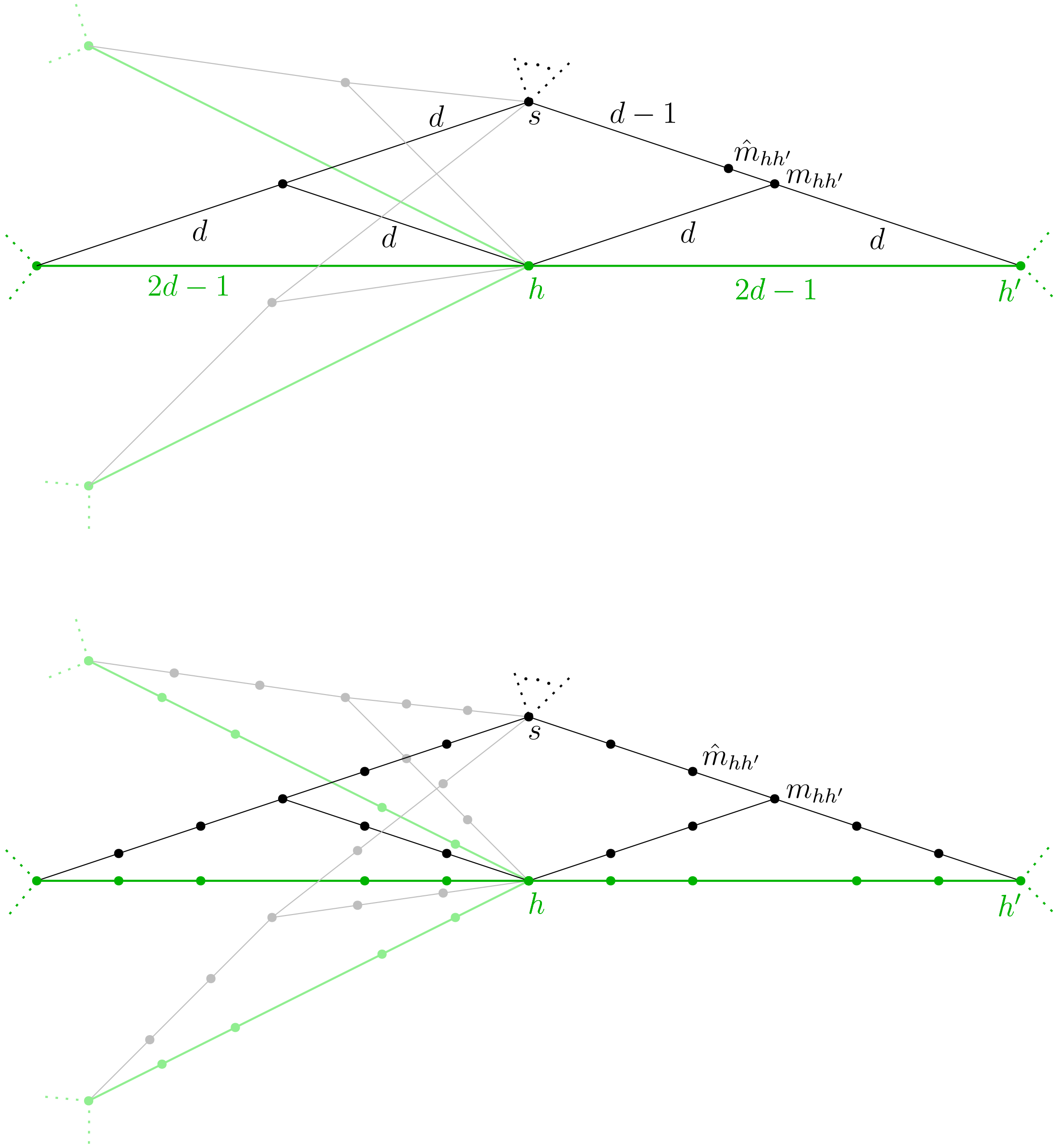

Figure 6: (A part of) the graph G_4 with some vertex h of H and another vertex $h' \in N_H(h)$, at the top for arbitrary d and at the bottom for $d = 3$. Edges with a label l represent paths of length l. For clearness, not all paths are labeled. Green vertices and edges correspond to vertices and (stretched) edges of H. The notation $\hat{m}_{hh'}$ is introduced later.

$2d - 1$ to the same vertex of $N_H(r)$. By the pigeonhole principle, there is a vertex h of $N_H(r)$ to which every cop has distance at least $2d$.
Note that R has distance $2d - 1$ to h.

Let R move to h.

Observe that no cop can capture R in the meantime.

Now:

- R is on a vertex of H
- R has not been captured

This is the same state description as after the first instruction block. We continue our strategy there.

For (2): In the description of the desired cop strategy, we use some additional notation:
For two vertices u and v of G_k with a unique shortest path p from u to v, we denote p by $[u, v]$. We use round brackets instead of square brackets on the left or right side (or both) to exclude u or v respectively (and the corresponding incident edge of p). The resulting paths may have length 0 or even order 0.
We denote the i-th $(0 \leq i \leq \text{dist}(u, v))$ vertex on $[u, v]$ by $[u, v](i)$.

For a large part of our strategy, R will be close to some vertex h of H, while each cop is close to a mid vertex $m_{hh'}$ with distance d to h, preventing R from moving to some $m_{hh'}$ or to h'. Because of the structure of G_k, every cop can only do this for one $h' \in N_H(h)$. Since there are as many cops as vertices in $N_H(h)$, every cop C is responsible for exactly one mid vertex $m(C)$ (and the corresponding endpoint of $m(C)$ in H). We call such a function m that assigns each cop a different mid vertex with distance d to h an ***h*-surrounding assignment**. For a cop C, we denote $m(C)$ by m_C.

Because our strategy is long, we first give a *rough* overview of it, textually as well as visually (in Figure 7):
After some preparation, the cops can (reach the desired state already, or) reach a state where for some vertex h of H to which R is close, a cop is on each of the k mid vertices closest to h. Then, we let the cops approach h simultaneously. As they do, R must move closer and closer to h, as well; if R moves further away from h than the cops, towards a vertex $h_2 \in N_H(h)$, the cop that is closest to h_2 can cut R off at h_2, whilst the other cops continue moving to h and then capture R (We call this part "catching" R).
At the end of this process, (the cops reached their goal already, or) R is on h with one cop directly next to R on the path to each of the k closest mid vertices, and has to choose between trying to evade to some $h_2 \in N[h]$ (which ends with R being caught as before), and eliminating a cop.
If R chooses to eliminate a cop, the remaining cops can again cut off R's possible escape routes and then capture R.

We now give the desired cop strategy. Depending on R's position in the starting state, we start with phase Prep-A or with phase Prep-B.

Phase Prep-A
Initially:
- there are k cops, all on s
- there is a mid vertex $m_0 = m_{h_1 h_2}$ so that R is in $(s, m_0]$ or in (m_0, h_1)

For a mid vertex m, we denote $[m, s](1)$ by $\hat{m}$. For example, for two adjacent vertices h_1 and h_2 of H, $\hat{m}_{h_1 h_2}$ is $\left[m_{h_1 h_2}, s\right](1)$.

Let $\hat{C}$ and C' be two cops.

- Let $\hat{C}$ move to m_0 until $\text{dist}(\hat{c}, \hat{m}_0) \leq \text{dist}(r, m_0)$
- meanwhile, let C' do nothing for one turn and move directly behind $\hat{C}$ at every following cop turn

Because C' always protects $\hat{C}$, R does not capture $\hat{C}$.

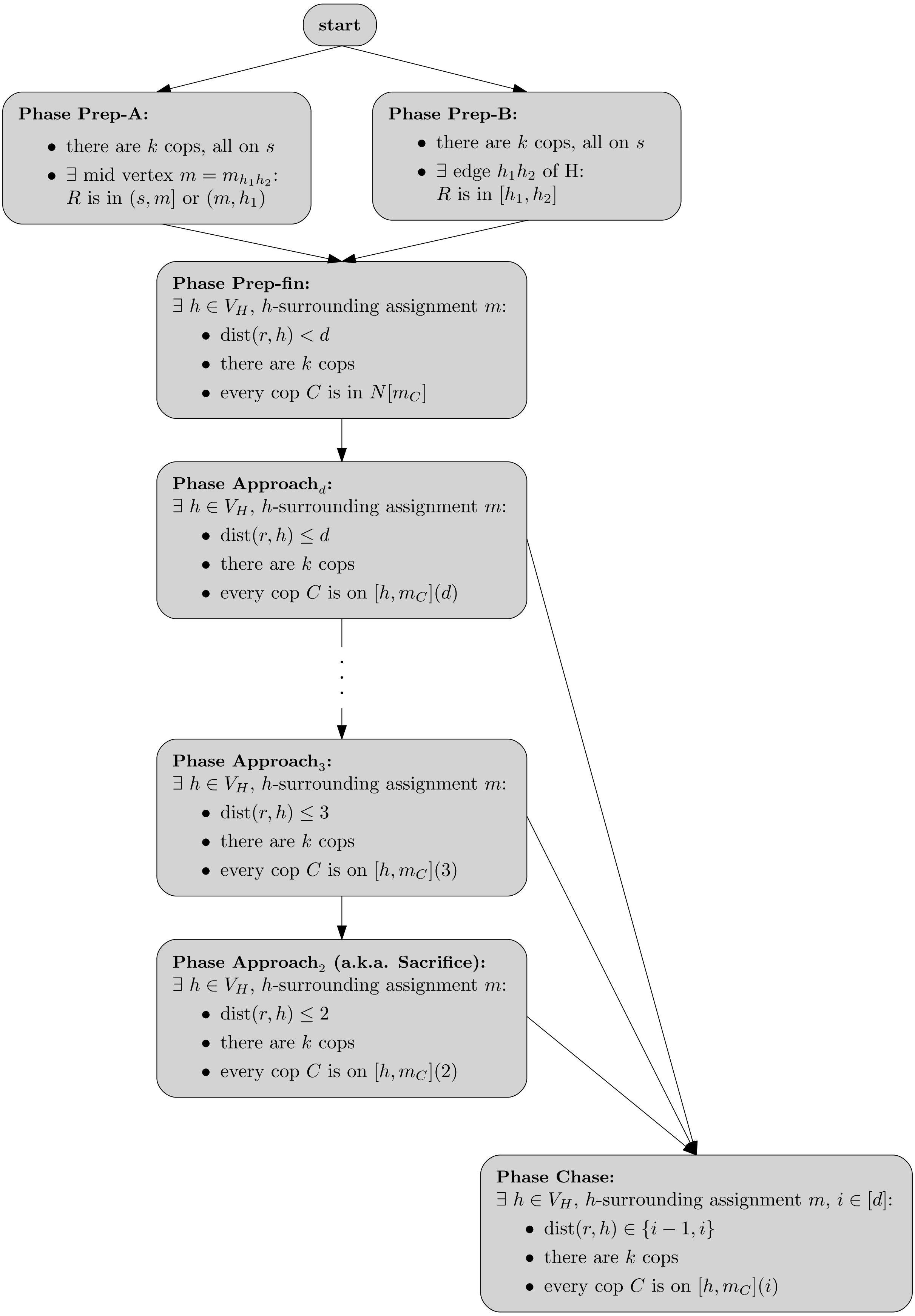

Figure 7: The phases of our cop strategy. Each phase contains a description of the possible/allowed initial states. An edge from a phase P to a phase Q represents that in (some case of) phase P, the strategy continues with phase Q. The vertical order of the phases is the same as in the textual description of our strategy.

In case $\hat{C}$ ever reaches $\hat{m}_0$, we have $\operatorname{dist}(\hat{c}, \hat{m}_0) = 0 \leq \operatorname{dist}(r, m_0)$. Thus, the cops stop before $\hat{C}$ reaches m_0.

In case R ever moves to h_1 or to h_2, the cops stop because then, $\operatorname{dist}(\hat{c}, \hat{m}_0) \leq d = \operatorname{dist}(r, m_0)$.

R is never in $[s, \hat{c}]$, and that when R is in $(\hat{c}, m_0]$, the cops do not stop.

Note that the last observations together imply that R is now on $[h, m_0)$ for some vertex h of $\{h_1 h_2\}$.

We show that $\operatorname{dist}(c', s) \leq \operatorname{dist}(r, h)$:
<u>Case 1:</u> The cops did not move (in this phase):

> The inequality is trivially satisfied because $\operatorname{dist}(c', s) = 0$.

<u>Case 2:</u> The cops moved at least once (in this phase):

> We need to show that the difference $\operatorname{dist}(r, h) - \operatorname{dist}(c', s)$ is nonnegative. Since $\operatorname{dist}(m_0, r) + \operatorname{dist}(r, h) = \operatorname{dist}(m_0, h) = d = \operatorname{dist}(m_0, s) = \operatorname{dist}(m_0, c') + \operatorname{dist}(c', s)$, the difference is the same as $\operatorname{dist}(c', m_0) - \operatorname{dist}(r, m_0)$, which is the same as $\operatorname{dist}(\hat{c}, \hat{m}_0) + 2 - \operatorname{dist}(r, m_0)$.
> Note that in one (cop or robber) turn, $\operatorname{dist}(\hat{c}, \hat{m}_0) + 2 - \operatorname{dist}(r, m_0)$ changes by at most 1. Also, it was larger than 2 at the previous cop turn. Thus, it is larger than 0, which proves our claim.

In summary:
- R is in $[h, m_0)$
- $\operatorname{dist}(\hat{c}, \hat{m}_0) \leq \operatorname{dist}(r, m_0)$
- $\operatorname{dist}(c', s) \leq \operatorname{dist}(r, h)$

Now let m be an h-surrounding assignment with $m_{\hat{C}} = m_0$.

Our next goal is to let every cop C move to $\hat{m}_C$ and to show that meanwhile, R cannot move too far away from h.

It holds $\operatorname{dist}(c', \hat{m}_{C'}) = \operatorname{dist}(c', s) + \operatorname{dist}(s, \hat{m}_{C'}) \leq \operatorname{dist}(r, h) + d - 1 = \operatorname{dist}(r, m_{C'}) - 1$.
For every cop C other than C', we have $\operatorname{dist}(c, \hat{m}_C) \leq d - 1 \leq \operatorname{dist}(r, m_C) - 1$ as well.
This means that every cop C can move to $\hat{m}_C$ before R can move to m_C. In particular, each cop C can move to $\hat{m}_C$ without being eliminated (meanwhile or directly afterwards), and

$$\text{this takes at most } \operatorname{dist}(c, \hat{m}_C) \leq \operatorname{dist}(r, m_C) - 1 \leq \operatorname{dist}(r, h) + d - 1 \text{ moves.} \qquad (*)$$

We do that:

Let every cop C move to $\hat{m}_C$.

Since $\hat{C}$ reaches $\hat{m}_{\hat{C}}$ before R can reach m_0, R cannot move to m_0 without being captured. Thus, R either remains in (m_0, h) or moves over h. If R moves over h, by $(*)$, R only makes $d - 1$ more move afterwards. In either case, we now have $\operatorname{dist}(r, h) \leq d - 1$.

In summary, now we have:
- $\operatorname{dist}(r, h) < d$
- there are k cops

- every cop C is on $\hat{m}_C$

We continue with phase Prep-fin.

Phase Prep-B

Initially:
- there are k cops, all on s
- R is in $[h_1, h_2]$ for an edge h_1h_2 of H

Let $\hat{C}$ be a cop. Let $m_0 := m_{h_1h_2}$.

Let $\hat{C}$ move to $[h_2, m_0](1)$ until R is on h_1 or on h_2.

Case 1: R is on h for some vertex h of $\{h_1, h_2\}$:

As in Phase Prep-A, we assign every cop C a different mid vertex, called m_C, with distance d to h. We choose an assignment with $m_{\hat{C}} = m_0$.

Let every cop C that is not on m_C move to $[m_C, c](1)$.

Note that this takes at most $d - 1$ moves (in particular, also for $\hat{C}$). Thus, R can only make $d - 1$ moves, too, and no cop is eliminated.

Now:
- $\text{dist}(r, h) < d$
- there are k cops
- every cop C is in $N[m_C]$

We continue with phase Prep-fin.

Case 2: R is in (h_1, h_2) and $\hat{C}$ is on $[h_2, m_0](1)$:

Note that as long as $\hat{C}$ stands still, $\hat{C}$ protects h_2.

Let $h := h_1$.
As before, we assign every cop C a different mid vertex, called m_C, with distance d to h. We choose an assignment with $m_{\hat{C}} = m_0$. Let C' be a cop other than $\hat{C}$.

We threaten to trap R in (h, h_2) if R does not move to h:

Let C' move to $[h, m_{C'}](1)$ until R is on h.

Case 2.1: R is not on h:

C' is in $N(h)$, and R is trapped in (h, h_2); if the cops stood still from now on, R could not leave (h, h_2) without moving to a protected vertex.

Let a cop C that is not $\hat{C}$ or C' move to c'. Afterwards, let C' and C move to h and then to h_2.

Now:
- R is captured
- there are at least $k - 1$ cops

We are done.

Case 2.2: R is on h:

Note that $\text{dist}(c, m_C) \leq d$ for every cop C. As in Case 1:

Let every cop C that is not on m_C move to $[m_C, c](1)$.

This takes at most $d - 1$ moves (in particular, also for $\hat{C}$). Thus, R can only make $d - 1$ moves, too, and no cop is eliminated.

Now:
- $\text{dist}(r, h) < d$
- there are k cops
- every cop C is in $N[m_C]$

We continue with phase Prep-fin.

Phase Prep-fin

Initially, there is a vertex h of H and an h-surrounding assignment m so that:
- $\text{dist}(r, h) < d$
- there are k cops
- every cop C is in $N[m_C]$

Case 1: For every cop C, R is not on $[m_C, h](1)$:

Let every cop C move to m_C.

No cop is eliminated.
Now:
- $\text{dist}(r, h) \leq d$
- there are k cops
- every cop C is on m_C

We continue with phase Approach_d.

Case 2: R is on $[m_{\hat{C}}, h](1)$ for some cop $\hat{C}$:

Since $d \geq 2$, R is not in $N[m_{\hat{C}}]$ for every cop $C \neq \hat{C}$.

We threaten to trap R on $(h, m_{\hat{C}})$ if R does not move towards h:

Let every cop $C \neq \hat{C}$ move to m_C.

Since $m_{\hat{C}}$ is protected by $\hat{C}$, R only had two options to move to a non-protected vertex:

Case 2.1: R stood still:

Let two cops that are not $\hat{C}$ move to h and then to $\hat{c}$.

Note that meanwhile, R cannot move to h unless h is already protected.

Now:
- R is captured
- there are at least $k - 1$ cops

We are done.

Case 2.2: R moved to $[m_{\hat{C}}, h](2)$:

Note that $\text{dist}(r, h) = d - 2 < d$.

Let $\hat{C}$ move to $m_{\hat{C}}$.

Now:
- $\text{dist}(r, h) \leq d$, since $\text{dist}(r, h)$ changes by at most one per robber turn
- there are k cops
- every cop C is on m_C

We continue with phase Approach$_d$.

Phase Approach$_i$ $(3 \leq i \leq d)$
Initially, there is a vertex h of H and an h-surrounding assignment m so that:
- $\text{dist}(r, h) \leq i$
- there are k cops
- each cop C is on $[h, m_C](i)$

For each cop C, let h_C be the vertex of H with $m_{hh_C} = m_C$, i.e. the vertex of H other than h that is closest to m_C.

There is a cop $\hat{C}$ so that R is in $[h, h_{\hat{C}}]$ or in $[h, m_{\hat{C}}]$, i.e. a cop "below" which R is. (If $r = h$, we can choose any cop as $\hat{C}$.)

Case 1: R is in $(h, h_{\hat{C}}]$ and $\text{dist}(r, h) \in \{i - 1, i\}$:

We continue with phase Chase.

Case 2: R is in $[h, m_{\hat{C}})$ and $\text{dist}(r, h) \leq i - 3$, or R is in $(h, h_{\hat{C}}]$ and $\text{dist}(r, h) \leq i - 2$:

Let every cop C move to $[c, h](1)$, i.e. one vertex closer to h.

No cop is eliminated.

Now:
- $\text{dist}(r, h) \leq i - 1$, since $\text{dist}(r, h)$ changes by at most one per robber turn
- there are k cops
- every cop C is on $[h, m_C](i - 1)$

We continue with phase Approach$_{i-1}$ (which may be phase Approach$_2$).

Case 3: R is on $[h, m_{\hat{C}}](i - 2)$:

As in phase Prep-fin, we threaten to trap R on $(h, \hat{c})$ if R does not move towards h:

Let every cop $C \neq \hat{C}$ move to $[h, c](i - 1)$.

Case 3.1: R stood still in its last move:

Now, $\text{dist}(c, h) = \text{dist}(r, h) + 1$ for every cop $C \neq \hat{C}$, and it is the cops' turn of course.

Let two cops C_1 and C_2 other than $\hat{C}$ move to h and then to $\hat{c}$.

Note that C_1 and C_2 reach a vertex in $N(h)$ before R can reach h.

Now:

- R is captured
- there are at least $k-1$ cops

We are done.

Case 3.2: R moved towards h in its last move:

We have $\text{dist}(r, h) = i - 3$.

Let $\hat{C}$ move to $[h, \hat{c}](i-1)$.

Now:
- $\text{dist}(r, h) \leq i - 2$
- there are k cops
- every cop C is on $[h, m_C](i-1)$

We continue with phase S_{i-1} (which may be phase Approach_2).

Phase Approach_2 (a.k.a. Sacrifice)

Initially, there is a vertex h of H and an h-surrounding assignment m so that:
- $\text{dist}(r, h) \leq 2$
- there are k cops
- each cop C is on $[h, m_C](2)$

We use the notation h_C as before.

Since R is not on a protected vertex, R is in $[h, h_{\hat{C}}]$ for some cop $\hat{C}$.

Case 1: $r \neq h$:

As in case 1 in Phase S2, we continue with phase Chase.

Case 2: $r = h$:

Let every cop C move to $[h, c](1)$.

Case 2.1: R is on $[h, h_{\hat{C}}](1)$ for some cop $\hat{C}$:

As before, we continue with phase Chase.

Case 2.2: R just eliminated a cop $C^\dagger$:

Let $\hat{C}_1$, $\hat{C}_2$ and C' be three different cops (note that there are at least three cops left).

Let G_R be the union of the three paths from $m_{C^\dagger}$ to h, to s and to $h_{C^\dagger}$ respectively.
Let $G_{C'}$ be the union of the three paths from $m_{C'}$ to h, to s and to $h_{C'}$ respectively.
There is an isomorphism f from G_R to $G_{C'}$ that maps s to s and h to h.
Note that $f(r) = c'$.

- let $\hat{C}_1$ and $\hat{C}_2$ move to $h_{C^\dagger}$
- meanwhile, let C' mirror R's behavior in G_R in $G_{C'}$, i.e. always move to $f(r)$

Note that $\hat{C}_1$ and $\hat{C}_2$ reach $[h_{C^\dagger}, h](1)$ before R can reach $h_{C^\dagger}$, and that C' captures R as soon as R moves to s or to h. Thus, R cannot leave G_R.

G_R is a tree, with $\hat{C}_1$ and $\hat{C}_2$ on one leaf (once they reached $h_{C^\dagger}$). If C' continues to prevent R from leaving G_R over s of h, $\hat{C}_1$ and $\hat{C}_2$ can capture R by always moving towards R:

- let $\hat{C}_1$ and $\hat{C}_2$ move to r until a cop has captured R
- meanwhile, let C' mirror R's behavior in G_R in $G_{C'}$, i.e. always move to $f(r)$

Now:
- R is captured
- R eliminated at most one unprotected cop (and maybe another protected cop)

We are done.

Phase Chase

Initially, there is a vertex h of H, an h-surrounding assignment m and a number $i \in [d]$ so that:
- $\operatorname{dist}(r, h) \in \{i-1, i\}$
- there are k cops
- each cop C is on $[h, m_C](i)$

We use the notation h_C as before.

Since R is not on a protected vertex, R is on $(h, h_{\hat{C}})$ for some cop $\hat{C}$.

Let $\hat{h}_{\hat{C}} := [h_{\hat{C}}, m_{\hat{C}}](1)$.

The plan is to let $\hat{C}$ move to $m_{\hat{C}}$ and then to $\hat{h}_{\hat{C}}$, cutting R off at $h_{\hat{C}}$, while two other cops cut R off at h and then capture R.

We have $\operatorname{dist}\left(\hat{c}, \hat{h}_{\hat{C}}\right) = d - i + d - 1 = 2d - 1 - i \leq \operatorname{dist}(r, h_{\hat{C}})$. This means that $\hat{C}$ can reach $\hat{h}_{\hat{C}}$ (and then protect $\hat{h}_{\hat{C}}$) before R can reach $h_{\hat{C}}$.

Let C_1 and C_2 be two cops other than C. We have $\operatorname{dist}(c_1, h) - 1 = i - 1 \leq \operatorname{dist}(r, h)$. This means that C_1 can reach a vertex in $N(h)$, thus protecting h, before R can reach h.

Let $\hat{C}$ move to $\hat{h}_{\hat{C}}$ and let C_1 and C_2 move to h and then to $h_{\hat{C}}$.

Note that from the time they reach h, C_1 and C_2 are always on the same vertex.

Now:
- R is captured
- there are at least $k - 1$ cops

We are done.

Note that throughout the whole strategy, R could never move to s unless s was protected. □

As in the proof without girth restrictions, we compose the graphs we found in Lemma 5.8 to create graphs with arbitrarily large girth in which R can eliminate all cops but two before being captured:

Theorem 5.9 (eliminate $\operatorname{cc} - 2$ cops in graphs with large girth): For every $k \geq 3$ and every $d \geq 3$, there is a connected graph G with girth at least d, $\operatorname{cc}(G) = k$ and $\operatorname{elim}(G) = k - 2$.

Proof. This proof works very similarly to the one of Theorem 5.4 (eliminate cc -1 cops), since the latter only uses the properties stated in Lemma 5.3 (and not details about our actual construction).

We first construct graphs F_k that have the desired properties only in certain states:

Claim 1: For every $k \geq 3$, there is a connected graph F_k with girth in $[d, \infty)$ and with two adjacent vertices c_0 and r_0 so that in every state in F_k with k cops, all on c_0, and R:
(1) if it is R's turn and R is on r_0, then R can eliminate $k-2$ cops before being captured
(2) if it is the cops' turn, the cops can reach a state where
- it is R's turn
- R is captured
- R was not on c_0 at any previous robber turn

Proof of Claim 1. We can reuse the inductive proof of Claim 1 from Theorem 5.4 (eliminate cc -1 cops), only with the different induction base $F_3 := C_d$ and of course a graph according to Lemma 5.8 instead of Lemma 5.3 in the induction step.
Note that the construction of F_k for $k > 3$ in the induction step does not introduce new cycles. □

We also construct the final graph G from F_k as in the proof of Theorem 5.4 (eliminate cc -1 cops). Note that no new cycles are created in this step as well. Thus, G has girth at least d.

Claim 2: $\text{elim}_k(G) \geq k - 2$.

Claim 3: $\text{cc}(G) \leq k$.

Like the construction of G, the proofs for Claim 2 and Claim 3 are similar to their counterparts in the proof of Theorem 5.4 (eliminate cc -1 cops).

Claim 4: $\text{cc}(G) \geq k$.

Proof of Claim 4. Similar to the proof of Claim 4 in Theorem 5.4 (eliminate cc -1 cops), we have

$$k - 2 \overset{\text{Claim 2}}{\leq} \text{elim}_k(G) \overset{\text{Claim 3, } \text{elim}_i(G)\searrow}{\leq} \text{elim}_{\text{cc}(G)}(G) \overset{\text{Theorem 5.7}}{\leq} \text{cc}(G) - 2.$$

□

The combination of Claim 2, Claim 3 and Claim 4 is what we had to show. □

6 Conclusion

The following questions remain open:

6.1 On the tightness of $\mathrm{cc} \le 2\,\mathrm{c}$

We show in Theorem 3.5 (connected graphs with $\mathrm{cc} = 2\,\mathrm{c}$) that for every $k \in \mathbb{N}_+$, there are connected graphs G with $\mathrm{cc}(G) = 2\,\mathrm{c}(G) = 2k$.

Our proof heavily relies on the triangles that the graph G we constructed contains: We used the fact G does not contain C_4 as a subgraph to conveniently lower bound the attacking cop number by the minimum degree of G. If we want to keep using this lower bound and avoid triangles as subgraphs, the graph we construct has girth at least 5. However, M. Aigner and M. Fromme [7] show that the cop number of a graph with girth at least 5 is at least its minimum degree. This means that our lower bound on the attacking cop number, which is also the minimum degree, becomes worthless for showing that the attacking cop number is much greater than the cop number; our approach does not work anymore.

Thus, we believe that any proof of a statement like Theorem 3.5 (connected graphs with $\mathrm{cc} = 2\,\mathrm{c}$) for connected graphs with larger girth would require finding more sophisticated robber strategies (than the one we found in Lemma 3.2 (lower bound for cc when no C_4)), and therefore probably an entirely different approach.

Question 6.1: How large can $\mathrm{cc}(G)$ be in terms of $\mathrm{c}(G)$ for connected graphs G with girth at least 4, or arbitrarily large girth?

6.2 Strong product

In Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$), we prove that for all connected graphs G and H with $\mathrm{c}(H) > 1$, we have $\mathrm{cc}(G \boxtimes H) \le \mathrm{c}(G) + \mathrm{cc}_{\mathrm{spread}}(H) - 1$.

We only show that Theorem 4.15 (upper bound for $\mathrm{cc}(G \boxtimes H)$) is tight in a weak sense (see Section 4.5).

Question 6.2: Are there connected graphs G and H with $\mathrm{c}(G) = k$ and $\mathrm{cc}_{\mathrm{spread}}(H) = l$ with $\mathrm{cc}(G \boxtimes H) = \mathrm{c}(G) + \mathrm{cc}_{\mathrm{spread}}(H) - 1$ for every $k \ge 2$ and $l \ge 1$?

One starting point may be to understand why $\mathrm{cc}(Q_3 \boxtimes Q_3) = 4$ (so far, we can only verify this with computer assistance), and then trying to generalize this construction.

As mentioned in Section 4.6, we are not aware of any non-trivial lower bounds for $\mathrm{cc}(G \boxtimes H)$ that hold for all connected graphs G and H.

Question 6.3: Is there a non-trivial lower bound for $\mathrm{cc}(G \boxtimes H)$ for all connected graphs G and H?

As it has been done for the cop number, the attacking cop number can also be investigated for other graph products, e.g. for the Cartesian product $\square$.

We believe that we can prove the following:

Conjecture 6.4: For all connected graphs G and H, it holds $\mathrm{cc}(G \square H) \le \mathrm{cc}(G) + \mathrm{cc}(H) + 1$.

6.3 Eliminated cops

For a connected graph G and a number $k \ge \mathrm{cc}(G)$, we define $\mathrm{elim}_k(G)$ as the maximum number of cops that R can eliminate before being captured (if R is ever captured) when

R plays a game of Cops and Attacking Robber against k cops in G. We also define $\mathrm{elim}(G) := \mathrm{elim}_{\mathrm{cc}(G)}(G)$.
Since we introduce this number, there is much about it to be discovered. We give some exemplary directions.

We know that for a bipartite connected graph G, it holds $\mathrm{elim}_{\mathrm{c}(G)+2}(G) \leq 1$. However, as discussed at the end of Section 5.3, the elimination number can decrease by an arbitrary amount when introducing one additional cop. Thus, we do not believe that we can use our bound for $\mathrm{elim}_{\mathrm{c}(G)+2}$ to derive any bound for $\mathrm{elim}_{\mathrm{c}(G)+1}$ or $\mathrm{elim}_{\mathrm{c}(G)}$.

Question 6.5: How large can $\mathrm{elim}(G)$ be in terms of $\mathrm{cc}(G)$ for bipartite connected graphs G?

Question 6.6: How large can $\mathrm{elim}_{\mathrm{c}(G)+1}(G)$ be in terms of $\mathrm{cc}(G)$ for bipartite connected graphs G?

We note that in our construction of connected graphs G with $\mathrm{elim}(G) = \mathrm{cc}(G) - 2$, the cycles formed by the paths $[h_1, h_2]$, $\left[h_2, m_{h_1 h_2}\right]$ and $\left[m_{h_1 h_2}, h_1\right]$ for two adjacent vertices h_1 and h_2 of H have odd length, and that reducing or increasing the length of one of the three paths without adjusting the other lengths as well breaks the cops' or R's strategy.
We believe that at least for Question 6.6, a non-trivial upper bound can be found by giving a cop strategy that sometimes delays a cop's moves, similar to the one in A. Clow, M. A. Huggan, and M. Messinger [1]'s proof for the fact that $\mathrm{cc}(G) \leq \mathrm{c}(G) + 2$.

Generalizing Question 6.6 to all connected graphs, we obtain the following:

Question 6.7: How large can $\mathrm{elim}_{\mathrm{cc}(G)+1}(G)$ be in terms of $\mathrm{cc}(G)$ for connected graphs G?

Recall that for the connected graph G_k we constructed in Theorem 5.4 (eliminate cc -1 cops), it holds $\mathrm{elim}_{\mathrm{cc}(G_k)+1} = 1$. (This also holds for the connected graphs we constructed in Theorem 5.9 (eliminate cc -2 cops in graphs with large girth).)

Question 6.8: Can we find (more) upper bounds for $\mathrm{elim}(G)$ for connected graphs G that have certain properties?

One such property is k-degeneracy (for $k \in \mathbb{N}_+$), a concept related to 2-cores.

6.4 Bipartite connected graphs

A. Bonato *et al.* [20] show that for every bipartite connected graph G, it holds $\mathrm{cc}(G) \leq \mathrm{c}(G) + 2$. A. Clow, M. A. Huggan, and M. Messinger [1] ask whether this bound is best possible. We give some evidence indicating that it might be improved:

Definition 6.9: Let us call a vertex v of a connected graph G an N-cut-vertex if $G - N[v]$ has multiple connected components.

Observation 6.10: If a connected graph G has no N-cut-vertex, it holds $\mathrm{cc}(G) \leq \mathrm{c}(G) + 1$.

Proof sketch. We give the idea how the cop strategy from the proof of A. Bonato *et al.* [20] can be modified: The strategy uses two cops $\hat{C}_1$ and $\hat{C}_2$, which always move together, to prohibit R from standing still for too long. The only time at which this is important is when r is adjacent to a vertex v to which some other cop C would like to move. In this case, since $G - N[r]$

is connected, $\hat{C}_1$ can move to c without moving into $N[r]$. Then, $\hat{C}_1$ can move to v, being protected by C, and forcing R to move away from r. Thus, $\hat{C}_2$ is no longer required. ⬚

This means that in a connected graph with $\mathrm{cc}(G) = \mathrm{c}(G) + 2$, there is an N-cut-vertex. Furthermore, every strategy for R to avoid the $\mathrm{c}(G) + 1$ cops indefinitely in G heavily utilizes such an N-cut-vertex v to keep cops from moving through $N[v]$, thereby "freezing" a current game configuration.
This sounds implausible enough for us to conjecture the following:

Conjecture 6.11: It holds $\mathrm{cc}(G) \leq \mathrm{c}(G) + 1$ for every bipartite connected graph G.

Bibliography

[1] A. Clow, M. A. Huggan, and M. Messinger, "Cops and attacking robbers with cycle constraints," *Discrete Applied Mathematics*, vol. 373, pp. 327–342, 2025, doi: https://doi.org/10.1016/j.dam.2025.05.019.

[2] S. Neufeld and R. Nowakowski, "A game of cops and robbers played on products of graphs," *Discrete mathematics*, vol. 186, no. 1–3, pp. 253–268, 1998.

[3] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning ways for your mathematical plays, volume 4*. AK Peters/CRC Press, 2004.

[4] J. Beck, "Combinatorial games," *Encyclopedia of Mathematics and its Applications*, vol. 114, 2008.

[5] R. Nowakowski and P. Winkler, "Vertex-to-vertex pursuit in a graph," *Discrete Mathematics*, vol. 43, no. 2, pp. 235–239, 1983, doi: https://doi.org/10.1016/0012-365X(83)90160-7.

[6] A. Quilliot, "Jeux et pointes fixes sur les graphes," 1978.

[7] M. Aigner and M. Fromme, "A game of cops and robbers," *Discrete Applied Mathematics*, vol. 8, no. 1, pp. 1–12, 1984, doi: https://doi.org/10.1016/0166-218X(84)90073-8.

[8] S. Durocher *et al.*, "Cops and Robbers on 1-Planar Graphs," in *Graph Drawing and Network Visualization*, M. A. Bekos and M. Chimani, Eds., Cham: Springer Nature Switzerland, 2023, pp. 3–17.

[9] M. Maamoun and H. Meyniel, "On a game of policemen and robber," *Discrete Applied Mathematics*, vol. 17, no. 3, pp. 307–309, 1987.

[10] B. W. Sullivan and M. Werzanski, "Lazy cops and robbers on product graphs." 2016.

[11] A. Berarducci and B. Intrigila, "On the Cop Number of a Graph," *Advances in Applied Mathematics*, vol. 14, no. 4, pp. 389–403, 1993, doi: https://doi.org/10.1006/aama.1993.1019.

[12] J. Petr, J. Portier, and L. Versteegen, "A faster algorithm for Cops and Robbers," *Discrete Applied Mathematics*, vol. 320, pp. 11–14, 2022, doi: https://doi.org/10.1016/j.dam.2022.05.019.

[13] W. B. Kinnersley, "Cops and Robbers is EXPTIME-complete," *Journal of Combinatorial Theory, Series B*, vol. 111, pp. 201–220, 2015, doi: https://doi.org/10.1016/j.jctb.2014.11.002.

[14] N. E. Clarke and R. J. Nowakowski, "Cops, robber and traps," *Utilitas Mathematica*, pp. 91–98, 2001.

[15] J. Chalopin, V. Chepoi, N. Nisse, and Y. Vaxès, "Cop and Robber Games When the Robber Can Hide and Ride," *SIAM Journal on Discrete Mathematics*, vol. 25, no. 1, pp. 333–359, 2011, doi: 10.1137/100784035.

[16] P. Frankl, "Cops and robbers in graphs with large girth and Cayley graphs," *Discrete Applied Mathematics*, vol. 17, no. 3, pp. 301–305, 1987, doi: https://doi.org/10.1016/0166-218X(87)90033-3.

[17] E. Chiniforooshan, “A better bound for the cop number of general graphs,” *Journal of Graph Theory*, vol. 58, no. 1, pp. 45–48, 2008, doi: https://doi.org/10.1002/jgt.20291.

[18] P. Prałat and N. Wormald, “Meyniel's conjecture holds for random d-regular graphs,” *Random Structures & Algorithms*, vol. 55, no. 3, pp. 719–741, 2019, doi: https://doi.org/10.1002/rsa.20874.

[19] W. Baird and A. Bonato, “Meyniel’s conjecture on the cop number: A survey,” *The Journal of Combinatorics*, vol. 3, pp. 225–238, 2013, [Online]. Available: https://api.semanticscholar.org/CorpusID:18942362

[20] A. Bonato *et al.*, “The Robber Strikes Back,” in *Computational Intelligence, Cyber Security and Computational Models*, G. S. S. Krishnan, R. Anitha, R. S. Lekshmi, M. S. Kumar, A. Bonato, and M. Graña, Eds., New Delhi: Springer India, 2014, pp. 3–12.

[21] A. Haidar, “The CC-Game: A Variant Of The Game Of Cops And Robbers,” 2012.

[22] A. Lacaze-Masmonteil, “Some Problems on the Game of Ambush Cops and Robbers,” 2019.

[23] S. S. Akhtar, S. Das, and H. Gahlawat, “Cops and Robber on Butterflies and Solid Grids,” in *Algorithms and Discrete Applied Mathematics*, A. Mudgal and C. R. Subramanian, Eds., Cham: Springer International Publishing, 2021, pp. 272–281.

[24] D. Kőnig, “Graphok és alkalmazásuk a determinánsok és a halmazok elméletére,” *Mathematikai és Természettudományi Ertesito*, vol. 34, pp. 104–119, 1916.