

# A Tight Lower Bound for the Capture Time of the Cops and Robbers Game<sup>\*†</sup>

Sebastian Brandt<sup>1</sup>, Yuval Emek<sup>2</sup>, Jara Uitto<sup>3</sup>, and Roger Wattenhofer<sup>4</sup>

- 1 ETH Zürich, Zürich, Switzerland  
brandts@ethz.ch
- 2 Technion, Haifa, Israel  
yemek@technion.ac.il
- 3 ETH Zürich, Zürich, Switzerland; and  
University of Freiburg, Freiburg, Germany  
jara.uitto@inf.ethz.ch
- 4 ETH Zürich, Zürich, Switzerland  
wattenhofer@ethz.ch

---

## Abstract

For the game of *Cops and Robbers*, it is known that in 1-cop-win graphs, the cop can capture the robber in  $O(n)$  time, and that there exist graphs in which this capture time is tight. When  $k \geq 2$ , a simple counting argument shows that in  $k$ -cop-win graphs, the capture time is at most  $O(n^{k+1})$ , however, no non-trivial lower bounds were previously known; indeed, in their 2011 book, Bonato and Nowakowski ask whether this upper bound can be improved. In this paper, the question of Bonato and Nowakowski is answered on the negative, proving that the  $O(n^{k+1})$  bound is asymptotically tight for any constant  $k \geq 2$ . This yields a surprising gap in the capture time complexities between the 1-cop and the 2-cop cases.

**1998 ACM Subject Classification** G.2.2 Graph Theory

**Keywords and phrases** cops and robbers, capture time, lower bound

**Digital Object Identifier** 10.4230/LIPIcs.ICALP.2017.82

## 1 Introduction

The game of *Cops and Robbers* is a perfect information two-player zero-sum game played on an undirected  $n$ -vertex graph  $G = (V, E)$ , where the first player is identified with  $k \geq 1$  *cops*, indexed by the integers  $0, \dots, k-1$ , and the second player is identified with a single *robber*. In round 0, the cop player chooses the initial (not necessarily distinct) cop locations  $c_0(0), \dots, c_{k-1}(0) \in V$  and following that, the robber player chooses the initial robber location  $r(0) \in V$ . Then, in round  $t = 1, 2, \dots$ , the cop player chooses the next (not necessarily distinct) cop locations  $c_0(t), \dots, c_{k-1}(t) \in V$  under the constraint that  $c_i(t) \in N^+(c_i(t-1))$  for every  $0 \leq i \leq k-1$ , where  $N^+(v)$  denotes the neighborhood of vertex  $v$  in  $G$  including  $v$  itself; following that, the robber player chooses the next robber location  $r(t) \in N^+(r(t-1))$ .

The goal of the cop player is to ensure that  $r(t-1) \in \{c_0(t), \dots, c_{k-1}(t)\}$  for some finite round  $t$ , referred to as *capturing* the robber. Conversely, the goal of the robber player is to

---

\* A full version of this paper can be found at <http://www.disco.ethz.ch/publications/icalp2017-copsandrobbers.pdf>.

† This work was partially supported by ERC Grant No. 336495 (ACDC).



© Sebastian Brandt, Yuval Emek, Jara Uitto, and Roger Wattenhofer;  
licensed under Creative Commons License CC-BY

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).

Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;

Article No. 82; pp. 82:1–82:13



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



avoid being captured indefinitely. Graph  $G$  is said to be a  $k$ -cop-win graph if it admits a cop strategy  $\mathcal{S}$  that guarantees capture. The *capture time* of  $\mathcal{S}$  is defined to be the maximum number of rounds until capture is achieved, assuming optimal play by the robber. The *capture time* of graph  $G$  is then defined to be the minimum capture time of any cop strategy over  $G$  (notice that in this definition, it is assumed that  $k$  is clear from the context).

Bonato et al. [7] studied the capture time in single cop games and proved that every 1-cop-win graph admits a cop strategy that captures the robber in  $\mathcal{O}(n)$  rounds. By considering a path, it is straightforward to verify that this bound is asymptotically tight. A simple configuration-counting argument (see, e.g., [5, 7]) shows that for any constant  $k \geq 2$ , if  $G$  is a  $k$ -cop-win graph, then its capture time is  $\mathcal{O}(n^{k+1})$ . One may suspect that this simple upper bound can be improved as it does not generalize the tight  $\mathcal{O}(n)$  bound in the 1-cop setting. Answering an open question from Bonato and Nowakowski's book [9, Chapter 8], the main result of our paper is that perhaps surprisingly, this is not the case.

► **Theorem 1.** *There exist a universal positive constant  $\alpha$  such that for every  $k \geq 2$ , there exists an infinite family  $\mathcal{G}$  of  $k$ -cop-win graphs such that the capture time of any  $n$ -vertex graph  $G \in \mathcal{G}$  is at least  $(n/(\alpha k))^{k+1}$ . Moreover, the smallest graph in  $\mathcal{G}$  has  $n = \mathcal{O}(k^2)$  vertices.*

Notice that for constant  $k \geq 2$ , this theorem provides an (existential)  $\Omega(n^{k+1})$  lower bound on the capture time in  $k$ -cop-win graphs. Furthermore, it can be extended to non-constant values of  $k = k(n)$  up to the conjectured maximum of  $k(n) = \Theta(\sqrt{n})$  (see the related literature discussion), stating that in some  $k$ -cop-win graphs, the capture time is exponential in  $k$  and stretched exponential in  $n$ .

## Related Literature

The Cops and Robbers game with a single cop was introduced by Quilliot [21] and independently by Nowakowski and Winkler [19] who also provided a full characterization of 1-cop-win graphs. This was generalized to the multiple cop setting by Aigner and Fromme [2] who defined the *cop number* of graph  $G$  to be the minimum number of cops that guarantees that the robber can be captured (that is, the minimum  $k$  for which  $G$  is a  $k$ -cop-win graph). Cast in this terminology, Aigner and Fromme proved that the cop number of any planar graph is at most 3. An upper bound of  $\mathcal{O}(r^2)$  on the cop number of graphs excluding  $K_r$  as minor was established by Andreae [4]; this result lies at the heart of the recent graph decomposition technique of Abraham et al. [1] for the same family of graphs. For general graphs, the maximum possible cop number is still an open question: the famous Meyniel's Conjecture [14, 6] states that this number is  $\Theta(\sqrt{n})$ , where the state of the art is that it is bounded between  $\Omega(\sqrt{n})$  [20] and  $\mathcal{O}(n/2^{(1-o(1))\sqrt{\log n}})$  [17]. Several characterizations of graphs with cop number  $k$  are presented in [11].

As mentioned earlier, Bonato et al. [7] established a tight linear bound on the capture time in 1-cop-win graphs. For  $k > 1$  cops, non-trivial bounds on the capture time in  $k$ -cop-win graphs were obtained mainly in the context of special graph classes, e.g., hypercubes [8] and Cartesian products of trees [18]. To the best of our knowledge, the linear lower bound of [7] is the (asymptotically) best previously known lower bound on the capture time in any class of graphs for any  $k \geq 1$ .

The capture time has been studied also for variants of the classic Cops and Robbers game. For example, the multiple robber setting was investigated by Förster et al. [13] who showed that the capture time may increase linearly with the number of robbers. Kehagias and Pralat [16] analyzed the expected capture time of a *drunk robber* whose strategy is

simply a random walk on the graph. For a broader overview of the results on the game of Cops and Robbers, the reader is referred to the book of Bonato and Nowakowski [9] and recent surveys [3, 10, 12, 15].

## Techniques

Our lower bound proof relies on designing a bad (from the perspective of the cops) graph  $G$  that consists of several components, of which each has a different role (see the overview in Section 2.1). Here, we provide a glimpse into this design from an alternative (strictly informal and somewhat inaccurate) angle that may shed additional light on the techniques we use. At the heart of our construction lies the concept of forcing each entity (cop or robber) to follow a designated (non-simple) path in  $G$ , where in the scope of this discussion, we assume that every vertex in  $G$  admits a self-loop so that these paths may include null moves. Specifically, graph  $G$  contains equally long paths  $\chi_0, \dots, \chi_{k-1}$  and  $\rho$ , referred to in this discussion as the *canonical paths* of the cops and robber, respectively. The best strategy of the cop player is then to assign one cop, say Cop  $i$ , to each path  $\chi_i$  so that  $c_i(t) = \chi_i(t)$  for every  $t$ ; in response to that, the best strategy of the robber is to play  $r(t) = \rho(t)$ . This induces a sequence  $\sigma$  of (distinct) configurations and the analysis is completed by showing that  $\sigma$  is sufficiently long and that the robber is captured only at its end.

The most challenging part in the design of such canonical paths is to prove that the aforementioned strategies are indeed optimal. To that end, we show that if the robber deviates from her canonical path at time  $t$ , then she is either captured immediately or the game shifts forward to a more advanced configuration  $\sigma(t')$  for some  $t' > t$ . Conversely, if some cop deviates from her canonical path at time  $t$ , then the game shifts backwards to a less advanced configuration  $\sigma(t')$  for some  $t' < t$ . The main feature in the latter argument is an *exit component*  $\mathcal{X}$ ; if the robber manages to reach  $\mathcal{X}$ , then she can force the game to shift backwards to the beginning, i.e., to  $\sigma(0)$ . This threat is the key ingredient in the analysis of the cop strategy: we construct the cops' components so that they must strictly follow their canonical paths in order to cover all exits in  $\mathcal{X}$ .

Due to the verbosity of our construction and to the space limitations, we defer all the proofs and the discussion of the case of more than 2 cops to the full version.

## Technical Terminology

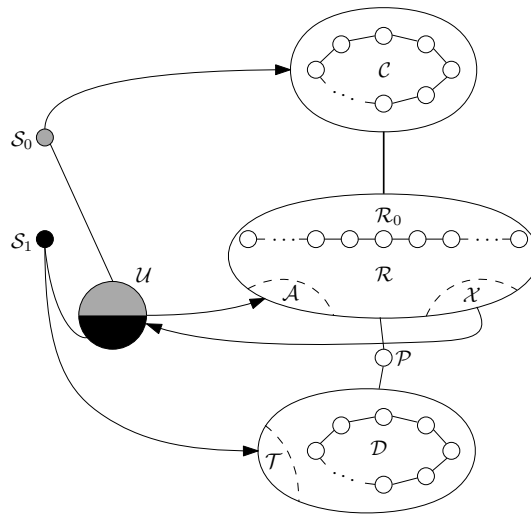
We call the set  $\{c_0(t), c_1(t), \dots, c_{k-1}(t)\}$ , for some  $t$ , a *cop combination*. We say that Cop  $i$  *covers* node  $v$  at time  $t$  if  $v \in N^+(c_i(t))$ . We may omit  $t$  if it is clear from the context. We extend this covering notion to more than one cop and more than one node, e.g., we say that *the cops cover* a set of nodes  $\{v_1, v_2, \dots, v_j\}$  if each of the  $v_i$  is covered by at least one of the cops. Also, we say that a node is *covered*, resp. *uncovered*, if at least one cop covers it, resp. if no cop covers it.

## 2 The Case of 2 Cops

### 2.1 Overview

In this section, we construct a family  $\{G_{\hat{n}}\}_{\hat{n} \geq 3, \hat{n} \equiv 0 \pmod{3}}$  of 2-cop-win graphs, where  $\hat{n} \in \Theta(n)$ .<sup>1</sup> Then, we show that the capture time for 2 cops in  $G_{\hat{n}}$  is  $\Omega(n^3)$  by giving an

<sup>1</sup> Throughout the paper, we denote the number of nodes of a graph by  $n$ .



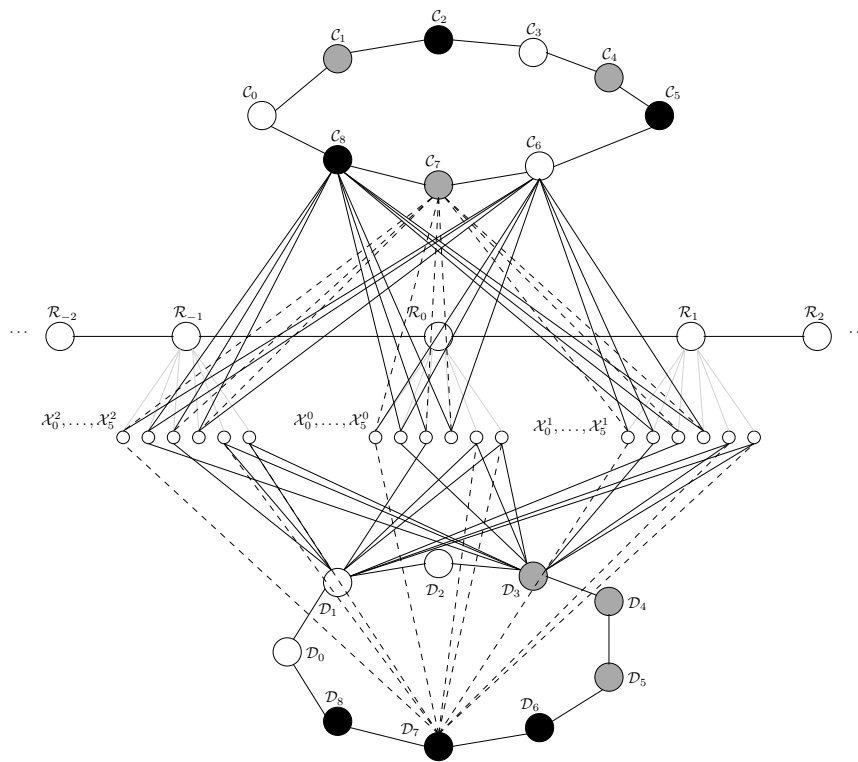
■ **Figure 1** The graph  $G_n$  with its different components.

explicit strategy for the robber that achieves this capture time against any cop strategy. We conclude by presenting a cop strategy for 2 cops that achieves a capture time of  $\mathcal{O}(n^3)$  against any robber strategy, which also serves as a proof that the graphs are indeed 2-cop-win. The other reason for explicitly specifying such a cop strategy (which *must* exist in 2-cop-win graphs, by the aforementioned simple configuration-counting argument) is that it forms the basis for a generalized cop strategy in the case of  $k \geq 2$  cops. For a simplified overview of our graph construction, please refer to Figure 1.

The general idea behind the graph construction and the specified strategies for the cops and the robber is as follows: Our graph  $G_n$  contains a  $\mathcal{U}$  component, where the robber cannot be captured, simply because each node in  $\mathcal{U}$  has enough neighbors so that 2 cops cannot cover all of these neighbors simultaneously. Moreover, the robber can always stay in  $\mathcal{U}$  (because each node in  $\mathcal{U}$  has enough neighbors *in*  $\mathcal{U}$ ) except if the cops go to two special nodes  $\mathcal{S}_0$  and  $\mathcal{S}_1$  which together cover all of  $\mathcal{U}$ .

When the robber is thereby flushed out of  $\mathcal{U}$ , she has to go to the  $\mathcal{R}$  component of the graph. Note that the nodes of  $\mathcal{R}$  induce a simple path on  $G_n$  and that after being flushed out of  $\mathcal{U}$ , the robber is located in the middle of this path. Now, the cops will continuously prevent the robber from escaping  $\mathcal{R}$  and slowly force the robber to one end of the path where they will finally capture her. In order for this to take a long time, each node in  $\mathcal{R}$  is connected to a set of so-called *exits* which are nodes that together form the  $\mathcal{X}$  component of the graph. If the robber should manage to get to some exit, then she will be able to go back to her preferred  $\mathcal{U}$  component, unless the cops go again to the special  $\mathcal{S}_0$  and  $\mathcal{S}_1$  nodes, in which case the robber can go back to the middle of the  $\mathcal{R}$  path and thereby revert to a previous configuration. Hence, in order to capture the robber, the cops have to continuously cover these exits.

Unfortunately for the cops, there are only a few cop combinations that actually cover all exits of a node in  $\mathcal{R}$ . Moreover, only some of these cop combinations are *proper* in the sense that they also prevent the robber from moving back on the  $\mathcal{R}$  path towards the middle which is essential for the cops in order to capture the robber. These proper *exit-covering* cop combinations are described in the following. For an illustration of the underlying graph structure, we refer to Figure 2.

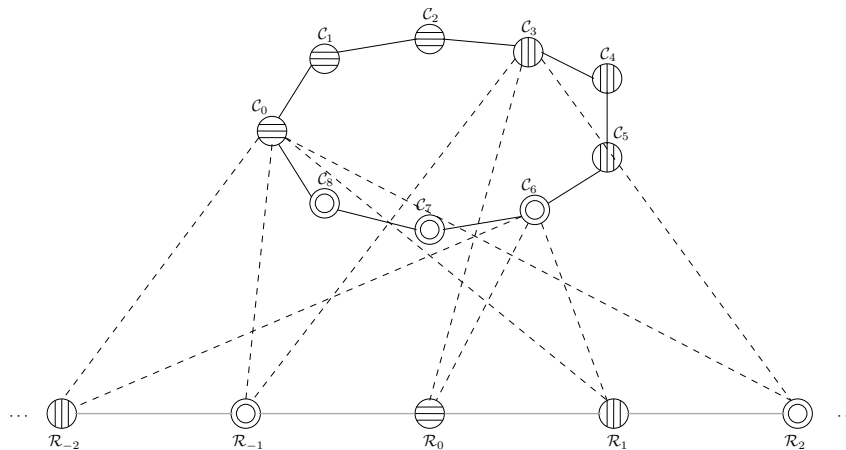


■ **Figure 2** A part of the structure of  $G_{\hat{n}}$  around the  $\mathcal{X}$  nodes. A  $\mathcal{C}$  node and a  $\mathcal{D}$  node together cover the exits of an  $\mathcal{R}$  node if and only if they have the same color.

One cop, say Cop 0, has to be in the  $\mathcal{C}$  component of  $G_{\hat{n}}$  and the other one (Cop 1) in the  $\mathcal{D}$  component. The nodes in the  $\mathcal{C}$  (resp.,  $\mathcal{D}$ ) component induce a simple cycle on  $G_{\hat{n}}$ . Assume for simplicity that the number of nodes in these two cycles are both multiples of 3. Now, we can imagine that the nodes in the  $\mathcal{C}$  cycle are consecutively colored  $0, 1, 2, 0, 1, 2, \dots$  and that the nodes in the  $\mathcal{D}$  cycle are colored  $0, \dots, 0, 1, \dots, 1, 2, \dots, 2$ , resulting in three equally-sized monochromatic blocks. Now, the nodes in  $\mathcal{C}$  and  $\mathcal{D}$  are connected to the nodes in  $\mathcal{X}$  in so that Cop 0 (in  $\mathcal{C}$ ) and Cop 1 (in  $\mathcal{D}$ ) cover all exits of an  $\mathcal{R}$  node if and only if the nodes the two cops are occupying have the same color. Thus, if Cop 0 wants to move, e.g., clockwise, in her  $\mathcal{C}$  cycle, then between any two consecutive steps, she has to wait for Cop 1 to travel roughly a third of her  $\mathcal{D}$  cycle.

Similarly, using an independent color pallet, we color the nodes along the  $\mathcal{R}$  path  $3, 4, 5, 3, 4, 5, \dots$  and color the nodes along the  $\mathcal{C}$  cycle  $3, \dots, 3, 4, \dots, 4, 5, \dots, 5$ . To prevent the robber from moving back towards the middle of her  $\mathcal{R}$  path, we construct  $G_{\hat{n}}$  so that a  $\mathcal{C}$  node covers an  $\mathcal{R}$  node if and only if they do *not* have the same color. Thus, if Cop 0 proceeds along her  $\mathcal{C}$  cycle, then as soon as the color of the  $\mathcal{C}$  node changes, the robber is forced to move one step forward along the  $\mathcal{R}$  path. This accounts for Cop 0 traversing roughly a third of the  $\mathcal{C}$  cycle for each step of the robber along the  $\mathcal{R}$  path. The direction of the robber’s movement (towards either end of the  $\mathcal{R}$  path) is determined by the direction (clockwise or counterclockwise) of the movement of Cop 0 along the  $\mathcal{C}$  cycle. We refer to Figure 3 for an illustration of how the robber is pushed along the  $\mathcal{R}$  by a cop residing in  $\mathcal{C}$ .

Now, we design the graph  $G_{\hat{n}}$  so that the  $\mathcal{C}$ ,  $\mathcal{D}$ , and  $\mathcal{R}$  components consist of roughly  $\hat{n}$  nodes. Thus, the robber takes  $\Omega(\hat{n})$  steps until she is captured, for each of her steps Cop 0



■ **Figure 3** The edge structure between the  $\mathcal{C}$  nodes and the  $\mathcal{R}$  nodes. If a cop is in a  $\mathcal{C}$  node and the robber in an  $\mathcal{R}$  node, then the robber has to make sure that that these two nodes have the same color in order to avoid capture in the next move. By circling clockwise in  $\mathcal{C}$ , the cop can force the robber towards one end of the  $\mathcal{R}$  path, by circling counterclockwise towards the other.

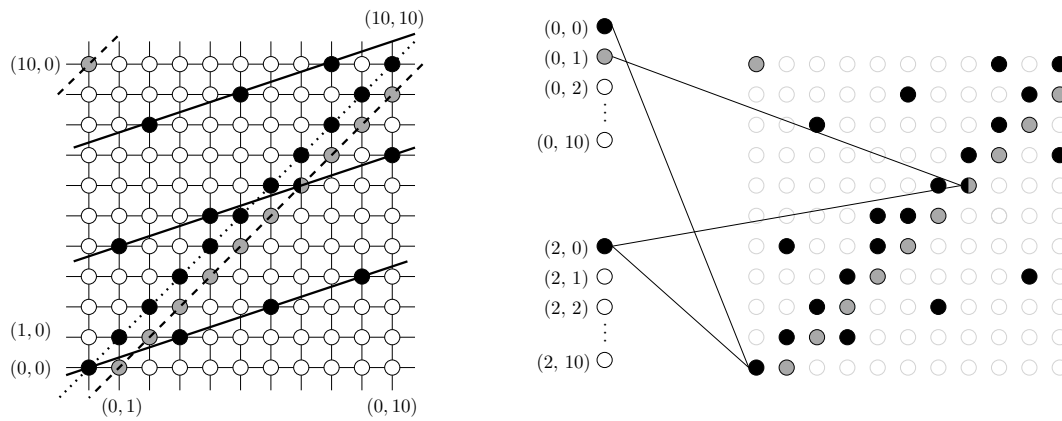
has to take  $\Omega(\hat{n})$  steps, and for each step of Cop 0, Cop 1 has to take  $\Omega(\hat{n})$  steps, resulting in a total capture time of  $\Omega(\hat{n}^3)$ . Since every component of the  $G_{\hat{n}}$  except  $\mathcal{R}$ ,  $\mathcal{C}$  and  $\mathcal{D}$  is of constant size,  $\hat{n}$  is linear in the number  $n$  of nodes. Hence, we obtain a capture time of  $\Omega(n^3)$  for the graph class  $\{G_{\hat{n}}\}_{\hat{n} \geq 3, \hat{n} \equiv 0 \pmod{3}}$ .

Before proceeding to the exact details of the graph construction, four remarks are in order. Firstly, when the robber is flushed out of  $\mathcal{U}$  to  $\mathcal{R}$ , there is an intermediate step between leaving  $\mathcal{U}$  and arriving in  $\mathcal{R}$  for technical reasons: Flushed out of  $\mathcal{U}$ , the robber actually has to go to a graph component called  $\mathcal{A}$ . Then, one cop moves to another graph component denoted  $\mathcal{T}$  from which she covers all of  $\mathcal{A}$  while ensuring that all nodes in  $\mathcal{U}$  the robber could go to are covered. Thus, the robber is flushed out of  $\mathcal{A}$  and is forced to go to  $\mathcal{R}$  so that we can proceed as explained above.

Secondly, since Cop 0 has to be able to cover nodes from  $\mathcal{R}$  when she is in  $\mathcal{C}$ , there have to be edges between  $\mathcal{R}$  and  $\mathcal{C}$ . To prevent the robber from escaping  $\mathcal{R}$  by going to a node in  $\mathcal{C}$  via one of these edges, we augment the graph with a special  $\mathcal{P}$  node. If, after the robber indeed moves to the  $\mathcal{C}$  component, Cop 1 moves to  $\mathcal{P}$  and Cop 0 moves to  $\mathcal{S}_0$ , then together they cover all  $\mathcal{C}$  nodes and all their neighbors, ensuring that the robber will be captured in the next round.

Thirdly, the robber does not have to start in  $\mathcal{U}$ , but in fact it is best for her if she does (in the sense of increasing the capture time against the best cop strategy), provided that she cannot be captured immediately. In turn, this means that the cops should start in  $\mathcal{S}_0$  and  $\mathcal{S}_1$  in order to force the robber to start elsewhere. Even if the robber starts elsewhere, the cop strategy explained above forces the robber to the  $\mathcal{R}$  component. Moreover, the robber can simply start in the middle of the  $\mathcal{R}$  path and the cops cannot avoid having to go through the exit-covering routine explained above.

Fourthly, while the cop strategy explained above chases the robber from the middle of the  $\mathcal{R}$  path to one of its ends, for simplicity, we will formally present a slightly simplified version where the end of the path the robber is chased to is fixed in advance (so the robber will be chased from one end of the path to the other end in the worst case for the cops).



**Figure 4** The construction of  $\mathcal{U}$ . On the left side, we see the  $11 \times 11$  grid constituting  $\mathcal{E}$  and three example lines  $\mathcal{L}_{0,0}$ ,  $\mathcal{L}_{0,1}$  and  $\mathcal{L}_{2,0}$  from  $\mathcal{L}$ . On the right side, we see the nodes of the bipartite graph  $G_{\mathcal{E}, \mathcal{L}}$  where the left-hand column of nodes constitutes (a part of) one side of the bipartition,  $\mathcal{L}$ , and the right-hand grid the other side,  $\mathcal{E}$ . The edges between  $\mathcal{E}$  and  $\mathcal{L}$  are determined by the incidence relation of the nodes and lines in the left-hand  $11 \times 11$  grid.

## 2.2 The Graph Construction

As explained above, the graphs  $G_{\hat{n}}$  we are about to construct contain a component  $\mathcal{U}$  in which the robber cannot be captured and from which the robber can only be flushed out by a specific cop combination outside of  $\mathcal{U}$ . Hence, the subgraph of any  $G_{\hat{n}}$  induced by (the respective)  $\mathcal{U}$  cannot be a 2-cop-win graph. As we want to generalize our graph construction to the case of more than 2 cops, we thus need a way to construct a graph where  $k$  cops cannot capture the robber. For this, inspired by the use of projective planes for constructing graphs with high cop numbers in [20], we will use incidence graphs of objects resembling affine planes. An explicit construction (for the case of 2 cops) is given in the following. For an illustration of the construction, we refer to Figure 4.

Let  $\mathcal{E} = \{\mathcal{E}_{i,j} \mid 0 \leq i \leq 10 \wedge 0 \leq j \leq 10\}$  be a set of *elements* which we can imagine as arrayed in an  $11 \times 11$  grid. Let  $\mathcal{L} = \{\mathcal{L}_{i,j} \mid 0 \leq i \leq 9 \wedge 0 \leq j \leq 10\}$  be a set of *lines* where each  $\mathcal{L}_{i,j}$  is defined as  $\mathcal{L}_{i,j} = \{\mathcal{E}_{h, h(i+1)+j \pmod{11}} \mid 0 \leq h \leq 10\}$ . Thus, each line  $\mathcal{L}_{i,j}$  may be considered as a “line modulo 11” in our grid which goes through the element  $\mathcal{E}_{0,j}$  and whose slope is determined by the parameter  $i$  (or more precisely  $i + 1$ ).

Now consider the *incidence graph*  $G_{\mathcal{E}, \mathcal{L}}$  for  $\mathcal{E}$  and  $\mathcal{L}$  which is defined as follows: The nodes of  $G_{\mathcal{E}, \mathcal{L}}$  are exactly the elements and lines defined above, i.e.,  $V(G_{\mathcal{E}, \mathcal{L}}) = \mathcal{E} \cup \mathcal{L}$ , and there is an edge between some node  $\mathcal{E}_{i,j}$  and some node  $\mathcal{L}_{i',j'}$  if and only if  $\mathcal{E}_{i,j}$  is contained in the set  $\mathcal{L}_{i',j'}$  (i.e., if and only if  $\mathcal{E}_{i,j}$  lies on the line  $\mathcal{L}_{i',j'}$ ). There are no other edges, hence  $G_{\mathcal{E}, \mathcal{L}}$  is bipartite where one side of the bipartition is given by  $\mathcal{E}$  and the other side by  $\mathcal{L}$ .

► **Lemma 2.** *In  $G_{\mathcal{E}, \mathcal{L}}$ , any two nodes in  $\mathcal{E}$  have at most one common neighbor in  $\mathcal{L}$ . Also, any two nodes in  $\mathcal{L}$  have at most one common neighbor in  $\mathcal{E}$ .*

► **Lemma 3.** *Let  $i \in \{0, \dots, 10\}$  be fixed. Any node from  $\mathcal{L}$  has exactly one neighbor in  $G_{\mathcal{E}, \mathcal{L}}$  of the form  $\mathcal{E}_{i,j}$  and exactly one neighbor in  $G_{\mathcal{E}, \mathcal{L}}$  of the form  $\mathcal{E}_{j,i}$ .*

► **Lemma 4.** *Let  $i \in \{0, \dots, 9\}$  be fixed. Any node from  $\mathcal{E}$  has exactly one neighbor in  $G_{\mathcal{E}, \mathcal{L}}$  of the form  $\mathcal{L}_{i,j}$ .*

For the construction of  $G_{\hat{n}}$ , we will *borrow* nodes from  $G_{\mathcal{E}, \mathcal{L}}$  and we will assume that the borrowed nodes take along their relationship concerning edges between them, i.e., there is an

edge between two borrowed nodes in our graph construction if and only if there is an edge between those two nodes in  $G_{\mathcal{E},\mathcal{L}}$ .

We construct  $G_{\hat{n}}$  as given in the following. The vertex set of  $G_{\hat{n}}$  is defined as the (disjoint) union of smaller vertex sets that constitute different parts of the graph with different purposes:

$$\begin{aligned} V(G_{\hat{n}}) &= \mathcal{E} \cup \mathcal{L}^* \cup \mathcal{S} \cup \mathcal{A} \cup \mathcal{T} \cup \mathcal{R} \cup \mathcal{C} \cup \mathcal{D} \cup \{\mathcal{P}\} \cup \mathcal{X}, \text{ where} \\ \mathcal{L}^* &= \{\mathcal{L}_{i,j} \mid 0 \leq i \leq 3 \wedge 0 \leq j \leq 10\} \\ \mathcal{S} &= \{\mathcal{S}_0, \mathcal{S}_1\} \\ \mathcal{A} &= \{\mathcal{A}_0, \dots, \mathcal{A}_3\} \\ \mathcal{T} &= \{\mathcal{T}_0, \dots, \mathcal{T}_3\} \\ \mathcal{R} &= \{\mathcal{R}_{-\hat{n}}, \dots, \mathcal{R}_{\hat{n}}\} \\ \mathcal{C} &= \{\mathcal{C}_0, \dots, \mathcal{C}_{\hat{n}-1}\} \\ \mathcal{D} &= \{\mathcal{D}_0, \dots, \mathcal{D}_{\hat{n}-1}\} \\ \mathcal{X} &= \mathcal{X}^0 \cup \mathcal{X}^1 \cup \mathcal{X}^2 \\ \mathcal{X}^j &= \{\mathcal{X}_0^j, \dots, \mathcal{X}_5^j\} \text{ for all } 0 \leq j \leq 2 \end{aligned}$$

The edges of  $G_{\hat{n}}$  are specified in Table 1. Moreover, we set  $\mathcal{U} = \mathcal{E} \cup \mathcal{L}^*$ . Furthermore, we call the nodes in  $\mathcal{X}$  *exits*, and for each node  $\mathcal{R}_i \in \mathcal{R}$  we call the nodes from  $\mathcal{X}$ , that are connected to  $\mathcal{R}_i$ , the *exits of  $\mathcal{R}_i$* .

The node subsets  $\mathcal{E}$  and  $\mathcal{L}^*$  are borrowed from  $G_{\mathcal{E},\mathcal{L}}$ , but also the (renamed) nodes in  $\mathcal{X}$  are borrowed from  $G_{\mathcal{E},\mathcal{L}}$ : We consider  $\mathcal{X}$  as a subset of  $\mathcal{L} \setminus \mathcal{L}^*$ . To ensure that no node in  $\mathcal{E}$  covers too many exits of some  $\mathcal{R}$  node, the exits of any  $\mathcal{R}$  node are borrowed (disjointly) from a set of  $\mathcal{L} \setminus \mathcal{L}^*$  nodes of the same slope. More precisely,

$$\begin{aligned} \mathcal{X}^0 &\text{ is borrowed from } \{\mathcal{L}_{4,j} \mid 0 \leq j \leq 10\}, \\ \mathcal{X}^1 &\text{ is borrowed from } \{\mathcal{L}_{5,j} \mid 0 \leq j \leq 10\}, \\ \mathcal{X}^2 &\text{ is borrowed from } \{\mathcal{L}_{6,j} \mid 0 \leq j \leq 10\}. \end{aligned}$$

As long as the above conditions are met, we do not care about the explicit choice of  $\mathcal{X}$  as a subset of  $\mathcal{L} \setminus \mathcal{L}^*$ . We obtain the following corollary from Lemma 2:

► **Corollary 5.** *Any two nodes in  $\mathcal{E}$  have at most one common neighbor in  $\mathcal{L}^*$ . Any two nodes in  $\mathcal{L}^* \cup \mathcal{X}$  have at most one common neighbor in  $\mathcal{E}$ .*

## 2.3 Observations

Before specifying asymptotically best strategies for the robber and the cops in  $G_{\hat{n}}$ , we gather some useful observations about the structure of  $G_{\hat{n}}$ . In particular, we examine which cop combinations cover certain neighbors of certain nodes. We start by showing that the cops have to be in  $\mathcal{S}_0$  and  $\mathcal{S}_1$  in order to flush the robber out of  $\mathcal{U}$ .

► **Lemma 6.** *Consider any  $u \in \mathcal{U}$ . The only cop combination not containing  $u$  that covers all neighbors of  $u$  in  $\mathcal{U}$  is  $\{\mathcal{S}_0, \mathcal{S}_1\}$ .*

We proceed by showing that if the robber has been flushed out of  $\mathcal{U}$  to some node  $\mathcal{A}_i$ , then the cops can only make progress by going to  $\{\mathcal{S}_0, \mathcal{T}_i\}$  because otherwise the robber can go back to  $\mathcal{U}$  (or there is no progress if the cops simply stay in  $\{\mathcal{S}_0, \mathcal{S}_1\}$ ).

► **Lemma 7.** *Consider any  $\mathcal{A}_i$ . The only cop combinations not containing  $\mathcal{A}_i$  that cover all neighbors of  $\mathcal{A}_i$  in  $\mathcal{U}$  are  $\{\mathcal{S}_0, \mathcal{S}_1\}$  and  $\{\mathcal{S}_0, \mathcal{T}_i\}$ .*



■ **Table 1** A listing of the edges of  $G_{\hat{n}}$ . In each block, the nodes listed in the right column are the neighboring nodes of the node listed in the left column. For simplicity, we omit a separate block for specifying the neighbors of the  $\mathcal{X}$  nodes. They can be inferred from the other blocks. The  $\mathcal{X}$  nodes do not have any edges connecting them to each other since they are borrowed from the  $\mathcal{L}$  part of  $G_{\mathcal{E},\mathcal{L}}$ .

Node	Neighbors
$\mathcal{E}_{i,j}$	$\mathcal{L}^*$ nodes as determined by $G_{\mathcal{E},\mathcal{L}}$ $\mathcal{S}_i \pmod{2}$ $\mathcal{A}_j \pmod{4}$ $\mathcal{T}_j \pmod{4}$ if $i \pmod{2} = 1$ $\mathcal{X}$ nodes as determined by $G_{\mathcal{E},\mathcal{L}}$
$\mathcal{L}_{i,j}$	$\mathcal{E}$ nodes as determined by $G_{\mathcal{E},\mathcal{L}}$ $\mathcal{S}_i \pmod{2}$ $\mathcal{A}_i \pmod{4}$ $\mathcal{T}_i \pmod{4}$

Node	Neighbors
$\mathcal{S}_0$	$\mathcal{C}_i$ for all $i$ $\mathcal{X}_i^j$ , where $j \in \{0,1,2\}$ and $i \in \{0,1,2\}$ every $\mathcal{E}_{i,j}$ with $i \pmod{2} = 0$ every $\mathcal{L}_{i,j}$ with $i \pmod{2} = 0$
$\mathcal{S}_1$	$\mathcal{T}_i$ for all $i$ $\mathcal{X}_i^j$ , where $j \in \{0,1,2\}$ and $i \in \{3,4,5\}$ every $\mathcal{E}_{i,j}$ with $i \pmod{2} = 1$ every $\mathcal{L}_{i,j}$ with $i \pmod{2} = 1$
$\mathcal{A}_i$	$\mathcal{T}_j$ for all $j$ $\mathcal{R}_0$ $\mathcal{D}_j$ for all $j$ $\mathcal{E}_{j,h}$ for all $h \pmod{4} = i$ $\mathcal{L}_{j,h}$ for all $j \pmod{4} = i$
$\mathcal{T}_i$	$\mathcal{S}_1$ $\mathcal{A}_j$ for all $j$ $\mathcal{T}_j$ for all $j \neq i$ $\mathcal{D}_j$ for all $j$ $\mathcal{P}$ $\mathcal{X}_h^j$ , where $j \in \{0,1,2\}$ and $h \in \{3,4,5\}$ $\mathcal{E}_{j,h}$ for all $j \pmod{2} = 1$ and $h \pmod{4} = i$ $\mathcal{L}_{j,h}$ for all $j \pmod{4} = i$

Node	Neighbors
$\mathcal{R}_i$	$\mathcal{A}_j$ for all $j$ if $i = 0$ $\mathcal{R}_j$ for $j = i - 1$ and $j = i + 1$ $\mathcal{C}_j$ for all $0 \leq j \leq \hat{n}/3$ if $i \pmod{3} = 0$ $\mathcal{C}_j$ for all $\hat{n}/3 \leq j \leq 2\hat{n}/3$ if $i \pmod{3} = 1$ $\mathcal{C}_0$ and $\mathcal{C}_j$ for all $2\hat{n}/3 \leq j \leq \hat{n} - 1$ if $i \pmod{3} = 2$ $\mathcal{P}$ $\mathcal{X}_h^j$ where $j = i \pmod{3}$ and $h \in \{0,1,2,3,4,5\}$
$\mathcal{C}_i$	$\mathcal{S}_0$ $\mathcal{R}_j$ for all $j \pmod{3} = 0$ if $0 \leq i \leq \hat{n}/3$ $\mathcal{R}_j$ for all $j \pmod{3} = 1$ if $\hat{n}/3 \leq i \leq 2\hat{n}/3$ $\mathcal{R}_j$ for all $j \pmod{3} = 2$ if $i = 0$ or $2\hat{n}/3 \leq i \leq \hat{n} - 1$ $\mathcal{C}_j$ for $j \equiv i - 1 \pmod{\hat{n}}$ and $j \equiv i + 1 \pmod{\hat{n}}$ $\mathcal{X}_h^j$ where $j \in \{0,1,2\}$ and $h \in \{0,1,3\}$ if $i \pmod{3} = 0$ $\mathcal{X}_h^j$ where $j \in \{0,1,2\}$ and $h \in \{0,2,3\}$ if $i \pmod{3} = 1$ $\mathcal{X}_h^j$ where $j \in \{0,1,2\}$ and $h \in \{1,2,3\}$ if $i \pmod{3} = 2$
$\mathcal{D}_i$	$\mathcal{A}_j$ for all $j$ $\mathcal{T}_j$ for all $j$ $\mathcal{D}_j$ for $j \equiv i - 1 \pmod{\hat{n}}$ and $j \equiv i + 1 \pmod{\hat{n}}$ $\mathcal{P}$ $\mathcal{X}_h^j$ where $j \in \{0,1,2\}$ and $h \in \{2,4,5\}$ if $0 \leq i \leq \hat{n}/3 - 1$ $\mathcal{X}_h^j$ where $j \in \{0,1,2\}$ and $h \in \{1,4,5\}$ if $\hat{n}/3 \leq i \leq 2\hat{n}/3 - 1$ $\mathcal{X}_h^j$ where $j \in \{0,1,2\}$ and $h \in \{0,4,5\}$ if $2\hat{n}/3 \leq i \leq \hat{n} - 1$
$\mathcal{P}$	$\mathcal{T}_i$ for all $i$ $\mathcal{R}_i$ for all $i$ $\mathcal{D}_i$ for all $i$ $\mathcal{X}_i^j$ where $j \in \{0,1,2\}$ and $i = 3$

The following lemma shows that if the cops allow the robber to go to an exit of some  $\mathcal{R}$  node, then they have to go back to  $\{\mathcal{S}_0, \mathcal{S}_1\}$  in order to prevent the robber from going to  $\mathcal{U}$ .

► **Lemma 8.** *Consider any  $\mathcal{X}_i^j$ . The only cop combination not containing  $\mathcal{X}_i^j$  that covers all neighbors of  $\mathcal{X}_i^j$  in  $\mathcal{U}$  is  $\{\mathcal{S}_0, \mathcal{S}_1\}$ .*

As Lemma 8 already indicates, the cops do not want the robber to be able to go to an exit from an  $\mathcal{R}$  node. The next lemma characterizes the cop combinations from where they can prevent the robber from doing that.

► **Lemma 9.** *Consider any  $\mathcal{R}_i$ . The only cop combinations not containing any  $\mathcal{R}_j$  with  $j \equiv i \pmod{3}$  that cover all exits of  $\mathcal{R}_i$  are  $\{\mathcal{S}_0, \mathcal{S}_1\}$ ,  $\{\mathcal{S}_0, \mathcal{T}_j\}$  for any  $j$ , and  $\{\mathcal{C}_j, \mathcal{D}_h\}$  for any pair  $(j, h)$  satisfying one of the following three conditions:*

1.  $j \pmod{3} = 0$  and  $0 \leq h \leq \hat{n}/3 - 1$ ,
2.  $j \pmod{3} = 1$  and  $\hat{n}/3 \leq h \leq 2\hat{n}/3 - 1$ ,
3.  $j \pmod{3} = 2$  and  $2\hat{n}/3 \leq h \leq \hat{n} - 1$ .

Observe that the cop combinations from Lemma 9 are independent of the choice of the considered  $\mathcal{R}_i$  which implies that these cop combinations cover all nodes in  $\mathcal{X}$ . We call such

■ **Table 2** The robber's strategy.

$\{c_0(t), c_1(t)\}$	$r(t-1)$	$r(t)$
$\neq \{\mathcal{S}_0, \mathcal{S}_1\}$	some node in $\mathcal{U}$	some uncovered node in $\mathcal{U}$
$\{\mathcal{S}_0, \mathcal{S}_1\}$	some node in $\mathcal{U}$	some node in $\mathcal{A}$
$\{\mathcal{S}_0, \mathcal{S}_1\}$ or $\{\mathcal{S}_0, \mathcal{T}_i\}$	$\mathcal{A}_i$	$\mathcal{R}_0$
$\neq \{\mathcal{S}_0, \mathcal{S}_1\}$ and $\neq \{\mathcal{S}_0, \mathcal{T}_i\}$	$\mathcal{A}_i$	some uncovered node in $\mathcal{U}$
not covering all exits of $\mathcal{R}_i$	$\mathcal{R}_i$	some uncovered exit of $\mathcal{R}_i$
covering all exits of $\mathcal{R}_i$	$\mathcal{R}_i$	the uncovered node from $\{\mathcal{R}_{i-1}, \mathcal{R}_i, \mathcal{R}_{i+1}\}$ with smallest absolute index; if all are covered, stay in $\mathcal{R}_i$
$\neq \{\mathcal{S}_0, \mathcal{S}_1\}$	$\mathcal{X}_i^j$	some uncovered node in $\mathcal{U}$
$\{\mathcal{S}_0, \mathcal{S}_1\}$	$\mathcal{X}_i^j$	some node from $\{\mathcal{R}_{-1}, \mathcal{R}_0, \mathcal{R}_1\}$

a cop combination *exit-blocking*. Furthermore, we call an exit-blocking cop combination *proper* if it does not contain a node from  $\mathcal{S}$  (i.e., it consists of a node from  $\mathcal{C}$  and a node from  $\mathcal{D}$ ). Lastly, we call a proper exit-blocking cop combination  $\{\mathcal{C}_i, \mathcal{D}_j\}$  *forcing* if there exist  $h, h' \in \{-1, 0, 1\}$ ,  $h \neq h'$ , such that the cops cover all  $\mathcal{R}_{h \pmod{3}}$  and all  $\mathcal{R}_{h' \pmod{3}}$ . A close look at the construction of  $G_{\hat{n}}$  shows that a proper exit-blocking cop combination  $\{\mathcal{C}_i, \mathcal{D}_j\}$  is forcing if and only if  $i \in \{0, \hat{n}/3, 2\hat{n}/3\}$ .

Proper exit-blocking cop combinations prevent the robber from going back (too much) towards the middle of the  $\mathcal{R}$  path since they contain a  $\mathcal{C}$  node which by its nature is connected to every third  $\mathcal{R}$  node. Thus, in order to be able to chase the robber towards one end of the  $\mathcal{R}$  path, the cops have to stay in proper exit-blocking cop combinations.

The  $\mathcal{C}$  node in a forcing proper exit-covering cop combination covers more  $\mathcal{R}$  nodes than the  $\mathcal{C}$  node in a usual proper exit-covering cop combination and thereby forces the robber to move one step towards the end of her  $\mathcal{R}$  path. In order to chase the robber another step, the cops have to go to a forcing proper exit-covering cop combination containing a different  $\mathcal{C}$  node. The following lemma shows a lower bound on the time it takes the cops to go from one forcing proper exit-covering cop combination to another with a different  $\mathcal{C}$  node, while using only proper exit-covering cop combinations on the way. Refer to Figures 2 and 3 for illustrations of the underlying idea.

► **Lemma 10.** *Let  $(\{\mathcal{C}_i, \mathcal{D}_j\} = \{c_0(t), c_1(t)\}, \{c_0(t+1), c_1(t+1)\}, \dots, \{c_0(t+h), c_1(t+h)\}) = \{\mathcal{C}_{i'}, \mathcal{D}_{j'}\}$  be a sequence of proper exit-blocking cop combinations describing the combined movement of the two cops from time  $t$  to time  $t+h$ . If  $\{\mathcal{C}_i, \mathcal{D}_j\}$  and  $\{\mathcal{C}_{i'}, \mathcal{D}_{j'}\}$  are forcing and  $i \neq i'$ , then  $h \geq \hat{n}/3 \cdot (\hat{n}/3 - 1) \in \Omega(\hat{n}^2)$ .*

## 2.4 The Robber's Strategy

Here, we explicitly specify a strategy for the robber that ensures that 2 cops need time  $\Omega(n^3)$  to capture the robber in  $G_{\hat{n}}$ :

If the cops are in  $\mathcal{S}_0$  and  $\mathcal{S}_1$  in round 0, then the robber starts in  $\mathcal{R}_0$ , otherwise the robber starts in some node in  $\mathcal{U}$  that is not covered by any of the cops (which exists by Lemma 6). Depending on where the cops are, the robber moves as specified in Table 2 (as long as she is not captured yet).

We show now that the specified strategy is well-defined, i.e., that the robber can perform any step in the strategy and that no other situations than the specified ones can occur if the robber follows the strategy. For the first part, we go through the table line by line:

■ **Table 3** The cops' strategy.

$r(t-1)$	$(c_0(t-1), c_1(t-1))$	$(c_0(t), c_1(t))$
$\mathcal{A}_i$	$(\mathcal{S}_0, \mathcal{S}_1)$	$(\mathcal{S}_0, \mathcal{T}_i)$
$\neq \mathcal{A}_i$ for all $i$	$(\mathcal{S}_0, \mathcal{S}_1)$	$(\mathcal{S}_0, \mathcal{T}_j)$ for some $j$
arbitrary	$(\mathcal{S}_0, \mathcal{T}_i)$	$(\mathcal{C}_0, \mathcal{D}_0)$
$\neq \mathcal{C}_h$ for all $h$	$(\mathcal{C}_i, \mathcal{D}_j)$	$(\mathcal{C}_{i+1 \pmod{\hat{n}}}, \mathcal{D}_{j+1 \pmod{\hat{n}}})$ if this covers all nodes in $\mathcal{X}$ $(\mathcal{C}_i, \mathcal{D}_{j+1 \pmod{\hat{n}}})$ otherwise
$\mathcal{C}_h$	$(\mathcal{C}_i, \mathcal{D}_j)$	$(\mathcal{S}_0, \mathcal{P})$

By Lemma 6, if the robber is in some node  $u \in \mathcal{U}$ , then she can always go to some uncovered node in  $\mathcal{U}$ , provided the cops are not in  $\mathcal{S}_0$  and  $\mathcal{S}_1$ . She can also go from  $u$  to some node in  $\mathcal{A}$  since any node in  $\mathcal{U}$  has some node in  $\mathcal{A}$  as a neighbor, by the construction of  $G_{\hat{n}}$ . Similarly, the robber can go from any node in  $\mathcal{A}$  to  $\mathcal{R}_0$ . By Lemma 7, if the robber is in some node  $\mathcal{A}_i$ , then she can always go to some uncovered node in  $\mathcal{U}$ , provided the cops are not in  $\mathcal{S}_0$  and  $\mathcal{S}_1$  or in  $\mathcal{S}_0$  and  $\mathcal{T}_i$ . The instructions where to go to from  $\mathcal{R}_i$  are trivially satisfiable. From  $\mathcal{X}_i^j$ , the robber can always go to some uncovered node in  $\mathcal{U}$  if the cops are not in  $\mathcal{S}_0$  and  $\mathcal{S}_1$ , by Lemma 8. She can also go to either  $\mathcal{R}_{-1}$ ,  $\mathcal{R}_0$  or  $\mathcal{R}_1$  from  $\mathcal{X}_i^j$  since each  $\mathcal{X}_i^j$  is connected to exactly one of those three  $\mathcal{R}$  nodes, by the construction of  $G_{\hat{n}}$ .

Moreover since the robber starts in a node in  $\mathcal{U}$  or  $\mathcal{R}$ , Table 2 covers all situations where the robber is in some node in  $\mathcal{U}$ ,  $\mathcal{A}$ ,  $\mathcal{R}$  or  $\mathcal{X}$ , and each instruction ends with the robber being in one of those nodes, the presented strategy specifies what the robber has to do for every possibly occurring situation.

## 2.5 The Cops' Strategy

Now, we explicitly specify a strategy for the cops that ensures that the robber is captured in time  $\mathcal{O}(n^3)$  in  $G_{\hat{n}}$ :

Cop 0 starts in  $\mathcal{S}_0$  and Cop 1 starts in  $\mathcal{S}_1$  in round 0. Depending on where the robber is, the cops move as specified in Table 3. There is one exception however: If a cop can capture the robber immediately, then she does so, overriding any possible instruction from the table.

We show now that the specified strategy is well-defined, i.e., that the cops can actually perform any step in the strategy and that no other situations than the specified ones can actually occur<sup>2</sup> if the cops follow the strategy:

The construction of  $G_{\hat{n}}$  ensures that the cops can actually move from the cop combinations at time  $t-1$  given in Table 3 to the cop combinations at time  $t$ . Since the cops start in  $\mathcal{S}_0$  and  $\mathcal{S}_1$ , the only thing that is left to show is that from  $(\mathcal{S}_0, \mathcal{P})$  (which is the only output combination that is not dealt with on the input side) the cops can capture the robber at time  $t+1$ , provided that the robber is in some  $\mathcal{C}_h$  at time  $t-1$ . For that, it is sufficient to observe that any neighbor of  $\mathcal{C}_h$ , and  $\mathcal{C}_h$  itself, is covered by  $\mathcal{S}_0$  or  $\mathcal{P}$ .

## 2.6 A Lower Bound for the Robber's Strategy

Here, we show that the strategy for the robber specified in Table 2 ensures that the cops need time  $\Omega(n^3)$  to capture the robber in  $G_{\hat{n}}$ . For convenience, we assume throughout the

<sup>2</sup> More precisely, if a situation occurs that is not specified in Table 3, then the cops can capture the robber immediately.

following lower bound considerations that if a cop can capture the robber immediately, then she does so. This certainly cannot worsen any strategy the cops follow.

We start by observing that the set of  $\mathcal{R}$  nodes can be partitioned into three roughly equally-sized sets such that the  $\mathcal{R}$  nodes in each such set have exactly the same exits. As the following lemma shows, if the robber is in an  $\mathcal{R}$  node, then she does not need to worry about a cop being in another  $\mathcal{R}$  node that has (and therefore covers) the same exits, since such a situation cannot occur if the robber follows the specified strategy.

We proceed by determining the nodes the robber can be captured in. Then, using Lemma 11 and Lemma 12, we give a lower bound on the capture time of  $G_{\hat{n}}$ .

► **Lemma 11.** *If the robber follows the strategy specified in Section 2.4, then the following holds: If the robber is in some node  $\mathcal{R}_i$  at time  $t$  and is not captured at time  $t + 1$ , then neither of the 2 cops can be in some node  $\mathcal{R}_j$  with  $j \equiv i \pmod{3}$  at time  $t + 1$ .*

► **Lemma 12.** *If the robber follows the strategy specified in Section 2.4, then she can only be captured in  $\mathcal{R}_{\hat{n}}$  or  $\mathcal{R}_{-\hat{n}}$ .*

► **Lemma 13.** *If the robber follows the strategy specified in Section 2.4, then 2 cops need time  $\Omega(\hat{n}^3)$  to capture the robber in  $G_{\hat{n}}$ .*

## 2.7 An Upper Bound for the Cops' Strategy

While the aim of this work is a lower bound, we need to show that 2 cops can actually capture the robber in  $G_{\hat{n}}$ , in order to use  $G_{\hat{n}}$  as a lower bound graph for the capture time for 2 cops in 2-cop-win graphs. We start by showing that from a proper exit-blocking cop combination the cops can always go to another proper exit-blocking cop combination by doing one of the following: Both cops move to the next node in their respective cycle or only the cop in the  $\mathcal{D}$  cycle moves to the next node.

► **Lemma 14.** *If  $(\mathcal{C}_i, \mathcal{D}_j)$  is an exit-blocking cop combination, then it holds that at least one of  $(\mathcal{C}_i, \mathcal{D}_{j+1 \pmod{\hat{n}}})$  and  $(\mathcal{C}_{i+1 \pmod{\hat{n}}}, \mathcal{D}_{j+1 \pmod{\hat{n}}})$  is an exit-blocking cop combination.*

The following lemma shows that, once the cops reach  $\mathcal{C}_0$  and  $\mathcal{D}_0$ , the robber cannot ever leave  $\mathcal{R}$  without being captured in the next two moves. Then, using Lemma 14 and Lemma 15, we give an upper bound on the capture time of  $G_{\hat{n}}$ .

► **Lemma 15.** *Let  $r(t) \in \mathcal{R}$  and  $(c_0(t+1), c_1(t+1)) = (\mathcal{C}_0, \mathcal{D}_0)$  for some point in time  $t$ . If the robber leaves  $\mathcal{R}$  at some later point in time  $t'$ , i.e., if  $r(t') \notin \mathcal{R}$  for some  $t' > t$ , then the robber will be captured at time  $t'' \leq t' + 2$ , provided the two cops follow the strategy specified in Section 2.5.*

► **Lemma 16.** *If the two cops follow the strategy specified in Section 2.5, then they capture the robber in time  $\mathcal{O}(\hat{n}^3)$  in  $G_{\hat{n}}$ .*

Finally, by Lemma 13 we get that for the case of 2 cops, the capture time of the graph family  $\{G_{\hat{n}}\}_{\hat{n} \geq 3, \hat{n} \equiv 0 \pmod{3}}$  is  $\Omega(\hat{n}^3) \subseteq \Omega(n^3)$  and by Lemma 16 we get that every graph in  $\{G_{\hat{n}}\}_{\hat{n} \geq 3, \hat{n} \equiv 0 \pmod{3}}$  is 2-cop-win. Together, these lemmas yield Theorem 1 for 2 cops.

### The Case of $k > 2$

Our graph construction and the corresponding lower bound proofs follow closely the design of the case of two cops. To accommodate a third cop, Cop 2, we essentially copy the  $\mathcal{D}$  component and ensure, that for every step of Cop 1, Cop 2 has to perform  $\Omega(\hat{n})$  steps. For the case of  $k > 3$  cops, we simply apply this trick inductively. Due to space limitations, we defer the detailed discussion of the case of  $k > 2$  cops to the full version.

---

**References**

---

- 1 I. Abraham, C. Gavoille, A. Gupta, O. Neiman, and K. Talwar. Cops, robbers, and threatening skeletons: padded decomposition for minor-free graphs. In *Proceedings of the 46th ACM Symposium on Theory of Computing, STOC*, pages 79–88, 2014.
- 2 M. Aigner and M. Fromme. A Game of Cops and Robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984. doi:10.1016/0166-218X(84)90073-8.
- 3 B. Alspach. Searching and Sweeping Graphs: a Brief Survey. *Le Matematiche*, 59:5–37, 2006.
- 4 T. Andreae. On a pursuit game played on graphs for which a minor is excluded. *Journal of Combinatorial Theory, Series B*, 41(1):37–47, 1986.
- 5 A. Berarducci and B. Intrigila. On the Cop Number of a Graph. *Advances in Applied Mathematics*, 14(4):389–403, 1993. doi:10.1006/aama.1993.1019.
- 6 A. Bonato and W. Baird. Meyniel’s Conjecture on the Cop Number: a Survey. *Journal of Combinatorics*, 3:225–238, 2012.
- 7 A. Bonato, P.A. Golovach, G. Hahn, and J. Kratochvíl. The Capture Time of a Graph. *Discrete Mathematics*, 309(18):5588–5595, 2009. doi:10.1016/j.disc.2008.04.004.
- 8 A. Bonato, P. Gordinowicz, B. Kinnersley, and P. Pralat. The Capture Time of the Hypercube. *Electr. J. Comb.*, 20(2):P24, 2013.
- 9 A. Bonato and R.J. Nowakowski. *The Game of Cops and Robbers on Graphs*, volume 61 of *Student Mathematical Library*. American Mathematical Society, Providence, RI, 2011.
- 10 A. Bonato and B. Yang. Graph Searching and Related Problems. In *Handbook of Combinatorial Optimization*, pages 1511–1558. Springer New York, 2013. doi:10.1007/978-1-4419-7997-1\_76.
- 11 N.E. Clarke and G. MacGillivray. Characterizations of k-copwin Graphs. *Discrete Mathematics*, 312(8):1421–1425, 2012. doi:10.1016/j.disc.2012.01.002.
- 12 F.V. Fomin and D.M. Thilikos. An Annotated Bibliography on Guaranteed Graph Searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.
- 13 K.-T. Förster, R. Nuridini, J. Uitto, and R. Wattenhofer. Lower Bounds for the Capture Time: Linear, Quadratic, and Beyond. In *22nd International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, pages 342–356, 2015.
- 14 P. Frankl. Cops and Robbers in Graphs with Large Girth and Cayley Graphs. *Discrete Appl. Math.*, 17(3):301–305, June 1987. doi:10.1016/0166-218X(87)90033-3.
- 15 G. Hahn. Cops, Robbers and Graphs. *Tatra Mt. Math. Publ*, 36(163):163–176, 2007.
- 16 A. Kehagias and P. Pralat. Some Remarks on Cops and Drunk Robbers. *Theoretical Computer Science*, 463:133–147, 2012.
- 17 L. Lu and X. Peng. On Meyniel’s Conjecture of the Cop Number. *Journal of Graph Theory*, 71(2):192–205, 2012. doi:10.1002/jgt.20642.
- 18 A. Mehrabian. The Capture Time of Grids. *Discrete Mathematics*, 311(1):102–105, 2011. doi:10.1016/j.disc.2010.10.002.
- 19 R.J. Nowakowski and P. Winkler. Vertex-to-vertex Pursuit in a Graph. *Discrete Mathematics*, 43(2-3):235–239, 1983. doi:10.1016/0012-365X(83)90160-7.
- 20 P. Pralat. When Does a Random Graph Have a Constant Cop Number. *Australasian Journal of Combinatorics*, 46:285–296, 2010.
- 21 A. Quilliot. *Jeux et Pointes Fixes sur les Graphes*. PhD thesis, Universite de Paris VI, 1978.