



HAL
open science

Cops and Robbers on Dynamic Graphs: Offline and Online Case

Stefan Balev, Juan Luis Jiménez Laredo, Ioannis Lamprou, Yoann Pigné, Eric Sanlaville

► **To cite this version:**

Stefan Balev, Juan Luis Jiménez Laredo, Ioannis Lamprou, Yoann Pigné, Eric Sanlaville. Cops and Robbers on Dynamic Graphs: Offline and Online Case. *Discrete Mathematics and Theoretical Computer Science*, 2023, vol. 25:1 (1), 10.46298/dmtcs.8784 . hal-02927320v5

HAL Id: hal-02927320

<https://normandie-univ.hal.science/hal-02927320v5>

Submitted on 7 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

manages to eventually lie at the same vertex as the robber, i.e., captures the robber. Otherwise, the robber wins if it can indefinitely avoid capture.

Cops and robbers applications are found in many fields, for example in motion planning (36), routing (23), network security (11), and distributed computing (6).

Thus far, Cops and Robbers literature has focused on (several variations of) the game taking place on static graphs. Little is known with respect to Cops and Robbers games played on *dynamic graphs*. Dynamic graphs, sometimes called temporal networks (28) or time-varying graphs (13), have received a lot of attention as they capture realistic scenarios where a graph evolves over time periodically or intermittently. Several models have been considered, for example see (12; 19; 29). In this paper, we introduce Cops and Robbers taking place on dynamic graphs in the following setting. A dynamic graph is represented by a (possibly infinite) sequence of subgraphs of the same static graph, which is the *underlying graph*. In other words, the (connected) underlying graph evolves over a series of discrete time steps under a set of evolution rules. We consider an offline and an online case. In the latter, in order to avoid trivial cases, we impose the natural constraint that the graph produced at each time step has to be connected.

Related Work. The preliminary question in mind is to compute the minimum number of cops needed to capture the robber on some (static) graph family.

Definition 1. *The cop number of a graph G , denoted $c(G)$, is the minimum number of cops needed to ensure that the robber is eventually captured, regardless of the robber's strategy.*

Problems related to the cop number have been studied heavily over the last four decades. Originally, Quillot (34), and independently Nowakowski and Winkler (32), characterized graphs with cop number equal to 1, otherwise referred to as *cop-win* graphs. The set of (di)graphs with cop number equal to $k > 1$ was characterized in (14; 18). Building on these notions, a general framework for characterizing discrete-time pursuit-evasion games was developed in (10).

There is a lot of literature regarding the cop number of specific graph classes. Aigner and Fromme (2) proved $c(G) \leq 3$ for any planar graph G . Frankl (17) proved a lower bound for graphs of large girth. Other works include (3; 16; 26).

Moving onto general graphs, Meyniel conjectured \sqrt{n} cops are always sufficient to capture the robber in any graph. The current state of the art is $\mathcal{O}(n/2^{(1-o(1))\sqrt{\log n}})$ proved independently in (24; 35). Yet, the conjecture remains unresolved. On the contrary, the conjecture was proved positive for binomial random graphs (33); relevant works include (7; 11; 25). The cop number is also related to various width parameters, for example see (1). Finally, there is a book (8) capturing recent activity on Cops & Robbers.

The computational complexity of the corresponding decision problem is also worth a note. *Given a graph G and an integer k , does $c(G) \leq k$ hold?* Recently, Kinnersley (22) answered the question by proving EXPTIME-completeness. With respect to algorithmic results, for a fixed constant k , there is a polynomial time algorithm to determine whether $c(G) \leq k$ (4). Other algorithmic results include (9) (capture from a distance), and (10) (generalized Cops and Robbers).

With respect to Cops and Robbers games played on dynamic graphs, there is preliminary work by Erlebach and Spooner (15). They examine the game on edge-periodic graphs, where each edge e is present at time steps indicated by a bit-pattern of length l_e used periodically and ad infinitum as evolution rule. Let $\text{LCM}(L)$ denote the least common multiple for input lengths l_e . The paper presents a $\mathcal{O}(\text{LCM}(L) \cdot n^3)$ algorithm to determine whether the graph is 1-cop-win as well as some other results on cycle graphs. Later on, in (30), NP- and W[1]-hardness results are provided for (parametrizations of) Cops and Robbers

played on temporal edge-periodic graphs. Further bounds and hardness results for cycle graphs in this model are given in (31).

Our Results. We consider two dynamic graph scenarios and present preliminary results for a (classic-style) Cops and Robbers game taking place in them. At each discrete time step of evolution, the current graph instance is fixed, then the cops take their turn, and finally the robber takes its turn. Note that movement may be restricted due to the possibly limited topology of each instance.

In the offline case, the cop and the robber know the whole evolution sequence (up to some finite time horizon T) a priori. For an underlying graph on n vertices, we prove that deciding whether it is 1-cop-win can be done in time complexity $O(n^3T)$; see Theorem 3. To do so, we employ a reduction to another game, a reachability game, played now on the configuration graph (Lemma 1). Our results extend to deciding k -cop-win graphs (Corollary 1), and an exponential time algorithm for determining the exact value of the cop-number (Corollary 2).

In the online case, no knowledge is given to the players regarding graph dynamics. The only restriction imposed is that, at each time step, the realized graph instance needs to be connected. We consider sparse graphs and show that the cop number is at most $\lambda + 1$ for underlying graphs with $n - 1 + \lambda$ edges; see Theorem 4. Moreover, we demonstrate a (nearly tight) graph family where $\lambda - 1$ cops are necessary (and sufficient) to ensure cop victory; see Theorem 5.

Outline. In Section 2, we present introductory notions and notation on the dynamic graphs used and on the game of Cops and Robbers played on them. In Sections 3 and 4, we formalize our definitions for the respective scenario: In Section 3, we consider the offline case, whereas in Section 4, we consider the online case. In Section 5, we make concluding remarks.

2 Preliminaries

Dynamic Graphs. Let $G = (V, E)$ stand for a (static) graph to which we refer to as the *underlying graph* of the model. We assume G is *simple*, i.e., not containing loops or multi-edges, and *connected*, i.e., there exists a path between any two vertices in G . No further assumptions are made on the topology of G . An edge from vertex $v \in V$ to vertex $u \in V$ is denoted as $(v, u) \in E$, or equivalently $(u, v) \in E$. We refer to the edges of the underlying graph as the *possible edges* of our model. We denote the number of vertices of G by $n = |V|$ and the number of its edges by $m = |E|$. For any vertex $v \in V$, we denote its *open neighborhood* by $N(v) = \{u : (v, u) \in E\}$ and its *closed neighborhood* by $N[v] = N(v) \cup \{v\}$. The (static) *degree* of $v \in V$ in G is given by $d(v) = |N(v)|$.

The dynamic graph evolves over a sequence of discrete time steps $t \in \mathbb{N}$. We consider two cases with respect to time evolution. First, $t = 1, 2, 3, \dots, T$, that is, t takes consecutive values starting from time 1 up to a *time horizon* $T \in \mathbb{N}$ given as part of the input. In this case, we define a dynamic graph \mathcal{G} with a time horizon T as $\mathcal{G} = (G_1, G_2, \dots, G_T)$ (Section 3). Second, $t = 1, 2, 3, \dots$, that is, we consider the sequence of time steps t evolving ad infinitum (Section 4).

For any t , let $G_t = (V_t, E_t)$ be the graph instance realized at time step t , where $V_t = V$ and $E_t \subseteq E$: all vertices of the underlying graph G are present at each time step, whereas a possible edge $e \in E$ may either be *present/alive*, i.e., $e \in E_t$, or *absent/dead*, i.e., $e \notin E_t$ at time t . For any vertex $v \in V$, we denote by $N_t(v) = \{u : (v, u) \in E_t\}$ its *available neighborhood* at time t . Similarly, let $N_t[v] = N_t(v) \cup \{v\}$ refer to the *available closed neighborhood* at time t .

Cops and Robber on Dynamic Graphs. We play a game of Cops and Robbers on a dynamic graph evolving under the general model defined above. There are two players: \mathcal{C} controlling $k \geq 1$ ($k \in \mathbb{N}$) cop tokens and \mathcal{R} controlling one robber token. Initially, \mathcal{C} places its k tokens on the vertices of the underlying graph. Notice that we allow *multiple cops to lie on the same vertex*. Afterward, \mathcal{R} chooses an initial placement for the robber. Round 0 is over. From now on, for every $t \geq 1$, first, the current graph instance G_t is fixed and, second, a round of the game takes place. A *round* consists of two *turns*, one for \mathcal{C} and one for \mathcal{R} , in this order of play. \mathcal{C} may move any of its cops lying on a vertex v to any vertex in $N_t[v]$. Note that all cops controlled by \mathcal{C} move simultaneously during \mathcal{C} 's turn. After \mathcal{C} 's turn is over, \mathcal{R} may move the robber lying on a vertex u to any vertex in $N_t[u]$. \mathcal{C} wins the game if, after any player's turn, the robber lies on the same vertex as a cop. \mathcal{R} wins if it can perpetually prevent this from happening.

A *cop-strategy*, respectively a *robber-strategy*, is a set of movement decisions for the cops, respectively the robber. Having knowledge of the current positions, and the current graph instance G_t (as well as all future graph instances only in the offline case), the cops/robber decide on a move for round t according to the rules of the game. A dynamic graph is called *k-cop-win*, if there exists a cop-strategy such that k cops win the game against any robber-strategy. For $k = 1$, we say that such a dynamic graph is *cop-win*.

3 Offline Case

In the offline case, we are given a dynamic graph \mathcal{G} with a time horizon T , namely $\mathcal{G} = (G_1, G_2, \dots, G_T)$, where both \mathcal{C} and \mathcal{R} have complete knowledge of the evolution sequence. That is, both players are aware of $G_t = (V, E_t)$, for all $t = 1, 2, \dots, T$, *a priori*. Let $c_{\text{off}}(\mathcal{G})$ stand for the *temporal cop number (offline case)*, the worst-case minimum number of cops required to capture a robber when the whole sequence $\mathcal{G} = (G_1, G_2, \dots, G_T)$ is given as input to both players. If the robber is not captured within T rounds, for any cop strategy, then the dynamic graph is robber-win. Overall, this is a Cops and Robbers game on a dynamic graph with bounded time horizon. From now on, we refer to it as the *offline case*. The results presented in this section can be viewed by the reader as an extension/completion of the work in (15) on edge-periodic graphs.

Configuration Graph. The task we tackle in the offline case is to characterize the set of given inputs (\mathcal{G}, T) , which are cop-win, i.e., one cop can always capture the robber within the T rounds of play. To do so, we first construct a directed *configuration graph* capturing the cop and robber motion on \mathcal{G} . Then, we can play another game, a *reachability game* (21) to be defined later, on the configuration graph which corresponds to the original cop and robber game played on \mathcal{G} and derive our result this way. We define the directed configuration graph as $P = (S, A)$, where S refers to configuration *states* (vertices) and A to arcs from one state to another state which is a feasible potential next state.

The vertex set S consists of all four-tuples of the form (c, r, p, t) , where $t \in \{1, 2, \dots, T\}$ indicates the time step or round of play t , $p \in \{\mathcal{C}, \mathcal{R}\}$ indicates whether it is the cop's or the robber's turn to play, $c \in V$ is the position of the cop just before p 's turn takes place in round t , and $r \in V$ is the position of the robber just before p 's turn takes place in round t .

The arc set A contains the arcs below, for all $x, y \in V$ and $t \in \{1, 2, \dots, T\}$, such that both the dynamics of the graph and the game moves are represented:

- (1) if $z \in N_t[x]$ and $t \in \{1, 2, \dots, T\}$, then $((x, y, \mathcal{C}, t), (z, y, \mathcal{R}, t)) \in A$, and
- (2) if $z \in N_t[y]$ and $t \in \{1, 2, \dots, T - 1\}$, then $((x, y, \mathcal{R}, t), (x, z, \mathcal{C}, t + 1)) \in A$

Case (1) arcs represent the cop's turn at round t , where the cop moves within its closed neighborhood available at time t , the robber retains its position, and, after the cop moves, it is the robber's turn at round t . Respectively, case (2) arcs represent the robber's turn at round t , where the robber moves within its closed neighborhood available at time t , the cop retains its position, and, after the robber moves, it is the cop's turn, but at the next round, namely round $t + 1$.

Let us now consider the size of P . By the definition of the states $s \in S$, it holds for the number of vertices $|S| \in \mathcal{O}(n^2T)$. Considering the set of arcs A , each vertex in S has at most n arcs leaving it, therefore we obtain $|A| \in \mathcal{O}(n^3T)$.

Before we proceed utilizing the configuration graph, let us add some auxiliary, yet necessary, states and arcs to capture the round of initial cop and robber placement, that is, round 0. This way, we ensure the full correspondence of the reachability game played on P to the cop and robber game played on \mathcal{G} . Note that all state and arc additions discussed hereunder do not affect the order of magnitude of the size of P . Let S contain also the states $(\emptyset, \emptyset, \mathcal{C}, 0)$, and $(x, \emptyset, \mathcal{R}, 0)$, for all $x \in V$. State $(\emptyset, \emptyset, \mathcal{C}, 0)$ captures the situation at round 0 before the cop's turn: neither the cop nor the robber have been placed yet on V . States $(x, \emptyset, \mathcal{R}, 0)$ capture the situation at round 0 before the robber's turn: the cop has been placed and it is the robber's turn to be placed. Overall, we have added an extra $n + 1$ states in S . We now proceed adding the necessary arcs in A to make the transitions from one turn to the next. For each $x \in V$, we add an arc $((\emptyset, \emptyset, \mathcal{C}, 0), (x, \emptyset, \mathcal{R}, 0)) \in A$, that is, n extra arcs in total. For each $x, y \in V$, we add an arc $((x, \emptyset, \mathcal{R}, 0), (x, y, \mathcal{C}, 1)) \in A$, that is, n^2 extra arcs in total.

Reachability. We now employ the configuration graph P constructed above by playing another two-player game on it referred to in literature as a *reachability game* (5; 21; 27). The goal is to define a reachability game, which corresponds exactly to the Cops and Robbers game (offline case), and so be able to utilize known results in this area to prove our cop-win characterization (Theorem 3). The connection of a (classic) Cops and Robbers game to a reachability game was first identified in (20). Other results, in (11; 18), on cop-win characterizations employ similar tools without explicitly stating the reduction to reachability.

A reachability game is played by two players, \mathcal{C} and \mathcal{R} , where we maintain the notation such that it corresponds to players in the game of Cops and Robbers. The two players play alternately on a directed graph $D = (V_D, A_D)$, where V_D is partitioned into two player-respective subsets, that is, $V_D = V_{\mathcal{C}} \cup V_{\mathcal{R}}$ and $V_{\mathcal{C}} \cap V_{\mathcal{R}} = \emptyset$. Moreover, $V_{\mathcal{C}}$, respectively $V_{\mathcal{R}}$, is a pairwise non-adjacent set of vertices, that is, for any $x, y \in V_{\mathcal{C}}$, respectively $x, y \in V_{\mathcal{R}}$, it holds $(x, y) \notin A_D$ and $(y, x) \notin A_D$. A single token is initially placed on a vertex $v \in V_D$. If $v \in V_{\mathcal{C}}$, then \mathcal{C} plays and moves the token to a vertex $u \in V_{\mathcal{R}}$ for which it holds $(v, u) \in A_D$. Then, it is \mathcal{R} 's turn: \mathcal{R} chooses to move the token to a vertex $w \in V_{\mathcal{C}}$ for which it holds $(u, w) \in A_D$. Note that either player has to move the token across an available arc in A_D . The game proceeds in this fashion for an indefinite number of rounds. Player \mathcal{C} wins, if the token eventually arrives to a designated *target vertex set* $Tar \subseteq V_D$. Otherwise, if for any \mathcal{C} -strategy a vertex in Tar can never be reached, then \mathcal{R} wins. In a nutshell, the reachability game played on $D = (V_D, A_D)$ is defined by the tuple $(V_{\mathcal{C}}, V_{\mathcal{R}}, Tar)$. Theorem 1 demonstrates that, depending on the initial token placement, there exists a winning strategy for one of the two players for *any input* $(V_{\mathcal{C}}, V_{\mathcal{R}}, Tar)$ and D . Moreover, by Theorem 2, it can be decided in time linear to the size of the directed graph D .

Theorem 1 ((5; 27)). *Consider a reachability game $(V_{\mathcal{C}}, V_{\mathcal{R}}, Tar)$ played on a directed graph $D = (V_D, A_D)$. V_D can be partitioned into two sets $W_{\mathcal{C}}$ and $W_{\mathcal{R}}$ such that, if the token is initially placed on $w \in W_p$, then there exists a winning strategy for player $p \in \{\mathcal{C}, \mathcal{R}\}$.*

Theorem 2 ((5; 21)). *There exists an algorithm computing winning sets W_C and W_R for a reachability game (V_C, V_R, Tar) played on a directed graph $D = (V_D, A_D)$ in time $\mathcal{O}(|V_D| + |A_D|)$.*

Let us now consider a reachability game taking place in our constructed configuration graph P : let $D = P$, $V_D = S$, and $A_D = A$. For any $(c, r, p, t) \in S$, let $(c, r, p, t) \in V_p$ where $p \in \{C, R\}$. Finally, let $Tar = \{(x, x, p, t) \mid x \in V, p \in \{C, R\}, t \in \{1, \dots, T\}\}$. We can now use the just defined sets V_C, V_R, Tar to prove Lemma 1 and then, as a consequence, our main result in Theorem 3.

Lemma 1. $\mathcal{G} = (G_1, G_2, \dots, G_T)$ is cop-win, if and only if, for a reachability game (V_C, V_R, Tar) played on $P = (S, A)$, where $V_p = \{(c, r, p, t) \in S \mid c, r \in V, t \in \{0, 1, \dots, T\}\}$, for $p \in \{C, R\}$, and $Tar = \{(x, x, p, t) \in S \mid x \in V, p \in \{C, R\}, t \in \{0, 1, \dots, T\}\}$, it holds $(\emptyset, \emptyset, C, 0) \in W_C$.

Proof: If \mathcal{G} is cop-win, the cop has a winning strategy s^* , which for any input cop position c , robber position r , and time step of evolution t before the cop's turn, provides the next placement for the cop token, say c' . Moreover, eventually, for some $t \leq T$, the cop is guaranteed to lie at the same vertex as the robber. Using the above strategy, Player C playing the reachability game on P also has a winning strategy: If the token lies at state $(c, r, C, t) \in S$, then C moves the token to (c', r, R, t) . Note that, since s^* is a feasible (winning) strategy for the Cops and Robbers game, by construction of P it holds $((c, r, C, t), (c', r, R, t)) \in A$. Also, since in s^* the cop eventually lies at the same vertex as the robber, the reachability token eventually reaches a vertex $(x, x, p, t) \in Tar$, for some $x \in V$, $p \in \{C, R\}$, and $t \leq T$. Hence, C wins the reachability game.

On the other hand, if $(\emptyset, \emptyset, C, 0) \in W_C$, by Theorem 1, there exists a winning strategy for C when the token lies on $(\emptyset, \emptyset, C, 0)$. For any $((c, r, C, t), (c', r, R, t)) \in A$ chosen by C as part of its winning strategy, by construction of P , the cop has a feasible move from c to c' at time t . Since C 's strategy is winning, the token eventually traverses an arc $((c, x, C, t), (x, x, R, t)) \in A$, where $(x, x, R, t) \in Tar$. Respectively, the cop will traverse $(c, x) \in E_t$ and capture the robber. \square

Theorem 3. *Given a dynamic graph $\mathcal{G} = (G_1, G_2, \dots, G_T)$ in the offline case, we can decide if $c_{\text{off}}(\mathcal{G}) = 1$, that is, if \mathcal{G} is cop-win, in time $\mathcal{O}(n^3T)$.*

Proof: By Lemma 1, it holds $c_{\text{off}}(\mathcal{G}) = 1$, if and only if, for a reachability game (V_C, V_R, Tar) played on $P = (S, A)$, where V_C, V_R, Tar are defined according to the statement of Lemma 1, it holds $(\emptyset, \emptyset, C, 0) \in W_C$. By Theorem 2, we decide whether $(\emptyset, \emptyset, C, 0) \in W_C$ in time $\mathcal{O}(|S| + |A|) = \mathcal{O}(n^2T + n^3T) = \mathcal{O}(n^3T)$. \square

An important remark is that, in Theorem 1 (5; 21), the winning strategy derived for player $p \in \{C, R\}$ is *memoryless*; see Proposition 2.18 in (27). In other words, it only depends on the current position of the token, and not on any past moves. By the reduction presented in Lemma 1, the winning strategy for the cop/robber is also memoryless: it only depends on the current positions of the cop, the robber, and the time step of evolution.

Let us conclude this part by explaining how the above framework can be generalized and therefore used to determine whether a dynamic graph $\mathcal{G} = (G_1, G_2, \dots, G_T)$ is k -cop-win, when $k > 1$. Since k cops are placed on V throughout the game, it suffices to expand our definition of states by substituting the cop position by a k -tuple of cop positions. That is, the state set S of the configuration graph P now contains tuples of the form $((c_1, c_2, \dots, c_k), r, p, t)$, where c_i , for $i \in \{1, 2, \dots, k\}$, denotes the location of the i -th cop in V . For the arc set, for C , we add $((c_1, c_2, \dots, c_k), r, C, t), ((c'_1, c'_2, \dots, c'_k), r, R, t) \in A$, if for each

$i \in \{1, 2, \dots, k\}$ it holds $c'_i \in N_t[c_i]$. For \mathcal{R} , we get $((c_1, c_2, \dots, c_k), r, \mathcal{R}, t), ((c_1, c_2, \dots, c_k), r', \mathcal{C}, t+1)) \in A$, if it holds $r' \in N_t[r]$. Again, we include auxiliary states and arcs to cater for the initial placements. Overall, we now get $|S| \in \mathcal{O}(n^{k+1}T)$, and $|A| \in \mathcal{O}(n^{2k+1}T)$, since for the dominant-in-magnitude number of \mathcal{C} -turn arcs there exist at most n^{2k} cop transitions from (c_1, c_2, \dots, c_k) to $(c'_1, c'_2, \dots, c'_k)$. By reapplying the whole framework with $Tar = \{((c_1, c_2, \dots, c_k), r, p, t) \mid c_i = r \text{ for some } 1 \leq i \leq k\}$ we conclude:

Corollary 1. *Given a dynamic graph $\mathcal{G} = (G_1, G_2, \dots, G_T)$ and an integer $k \geq 1$ in the offline case, we can decide if $c_{\text{off}}(\mathcal{G}) \leq k$, i.e., if \mathcal{G} is k -cop-win, in time $\mathcal{O}(n^{2k+1}T)$.*

We may now run a search utilizing the result in Corollary 1 and derive an exponential time algorithm to determine the exact value of $c_{\text{off}}(\mathcal{G})$.

Corollary 2. *For some dynamic graph \mathcal{G} , with an associated time horizon T , the problem of determining the exact value of $c_{\text{off}}(\mathcal{G})$ is in EXPTIME.*

4 Online Case

In the online case, we are given an underlying graph $G = (V, E)$ and an indefinite number of discrete time steps of evolution $t = 1, 2, 3, \dots$, that is, time evolution may take place ad infinitum. At each time step t , an instance $G_t = (V_t, E_t)$ is realized, where $V_t = V$, $E_t \subseteq E$. The *only assumption* we make on the topology of generated instances, is that we require each G_t to be *connected*. Note that this is a widely used assumption in several dynamic graphs appearing in literature (12; 28). Removing this assumption could lead to trivial cases where, for instance, the k cops or the robber lie indefinitely on isolated vertices.

Initially, the cops and then the robber place themselves on V before the appearance of G_1 . In the general case, neither the cops nor the robber have any knowledge about the evolution sequence. The cops and the robber, taking turns in this order, make their respective moves in G_t , then G_{t+1} is generated, and so forth. Similarly to the offline case, a token at vertex v moves to a vertex in $N_t[v]$ (all the cops move simultaneously). Let $c_t(G)$ stand for the *temporal cop number*, the worst-case minimum number of cops required to capture a robber for an underlying graph evolving like described above. In our analysis, we consider worst-case scenarios for the temporal cop number; a different type of analysis is left for future work. In other words, for our bounds to follow, one may assume the robber controls the dynamics of G to its advantage. Hence, at round t , the robber defines instance G_t according to the above restrictions.

Preliminary Bounds. As a warm up, let us consider two special cases for the topology of the underlying graph: a tree, and a complete graph.

Proposition 1. *For any tree T , it holds $c_t(T) = 1$.*

Proof: Since, for any time step t , G_t must be connected, it follows $G_t = T$. Since the topology of the tree remains static over time, it holds $c_t(T) = c(T)$. It suffices to verify $c(T) = 1$ for any T (26). \square

Proposition 2. *For any complete graph K_n , $n \geq 2$, it holds $c_t(K_n) = n - 1$.*

Proof: It holds $c_t(K_n) \leq n - 1$, since we can initially place $n - 1$ cops on $n - 1$ distinct vertices at K_n . The robber places itself on the only cop-free vertex. Then, G_1 is realized: since G_1 is connected, there exists at least one edge connecting a cop-vertex to the robber-vertex. The corresponding cop traverses that edge and captures the robber at the first cop turn.

We now demonstrate $c_t(K_n) > n - 2$. The $n - 2$ cops are initially placed on vertices of V . The robber places itself on a cop-free vertex. Regardless of the cops placement, at any time, there are always at least two cop-free vertices. Without loss of generality, for each time t , G_t is a path $v_1, v_2, \dots, v_{n-1}, v_n$ with the currently cop-free vertices at one end of the path. For example, the robber lies on v_1 , and v_2 (and possibly other vertices) are cop-free. The cops move during their turn, but they cannot capture the robber: Since the number of cop-free vertices is at least two, a cop can only reach a vertex at distance at least one from the robber (v_2). The robber remains at its position indefinitely and avoids capture. \square

The above propositions cast some intuition on the relationship between the (static/classical) cop number $c(G)$ and our introduced temporal cop number $c_t(G)$. For the static case, it is easy to see that if G is either a tree or a clique then $c(G) = 1$. However, in the temporal case $c_t(T) = 1$ for a tree T , and $c_t(K_n) = n - 1$ for any clique on $n \geq 2$ vertices. Intuitively, the denser the underlying graph is, the more leeway there is for the robber due to worst-case dynamics. Overall, for any graph G , $c_t(G) \leq n - 1$, since initially placing the $n - 1$ cops on distinct vertices guarantees an edge between a cop-vertex and the robber-vertex in G_1 due to connectedness. Thus, for the ratio of the two cop numbers, we get $1 \leq c_t(G)/c(G) \leq n - 1$.

We now provide a preliminary bound on $c_t(\cdot)$ by considering a subset of sparse graphs, that is, underlying graphs with at most linear number of edges.

Theorem 4. *For any graph $G = (V, E)$, $m = n - 1 + \lambda$, it holds $c_t(G) \leq \lambda + 1$.*

Proof: To describe the cop-winning strategy, let us define a partition of the vertices into V_C and V_R such that $V = V_C \cup V_R$ and $V_C \cap V_R = \emptyset$. Intuitively, V_C stands for the *cop-secured* vertices, i.e., vertices the robber will never be able to visit, whereas V_R stands for the vertices (possibly) still within the eventual reach of the robber. More precisely, the cop strategy below builds a sequence of partitions (V_C, V_R) where V_C is a set of vertices the robber will never be able to visit, V_R contains the other vertices and the cardinality of V_R strictly decreases at each time step. This strategy may not be the fastest as V_R may contain robber-unreachable vertices, but this is not required for the proof.

Consider the situation before some round t . Let T denote some (arbitrary) spanning tree of G . We refer to the edges of T as the *black edges* and to any path consisting only of black edges as a *black path*. We refer to all other edges, which are exactly λ , as the *blue edges*. Suppose there is one cop at one extremity of each blue edge. Note that several cops may lie on the same vertex. We refer to these cops as the *blue cops*. One last cop, the *black cop*, is placed on some other (blue-cop free) vertex, say $x \in V$. The robber is on a cop-free vertex, say $r \in V$. For a visual assistance for the rest of the proof, please refer to Figure 1.

Consider the spanning tree T : there exists a unique (black) path from x to r . Let (x, x') be the first edge of this path. If this edge is removed from the black tree T , T is split into two black subtrees containing x and x' respectively, namely T_x and $T_{x'}$. Then, let $V_C = V(T_x)$ and $V_R = V \setminus V_C = V(T_{x'})$. Notice that it holds $r \in V_R$ with this partition.

By construction, the cut associated to (V_C, V_R) contains exactly one black edge, (x, x') , plus (possibly) some blue edges. Since G_t is connected for all time steps t , at least one edge associated to the cut is present in E_t . If the black edge (x, x') is present, then the black cop moves from x to x' during the cops turn. Otherwise, if only a blue edge, say (v, v') , where $v \in V_C$, is present, then the associated cop moves from v to v' (or remains at v' if it were already there). Now, we *swap* the role of the two edges. That is, (v, v') becomes a black edge, and its associated cop becomes the black cop, whereas (x, x') becomes a blue edge, and its associated cop becomes a blue cop. By construction, the set of black edges defines a new black spanning tree T' : the unique black path from v to v' is replaced by the new black edge (v, v') .

(Notice that, in the previous swap-less case, we trivially had $T' = T$). Afterwards, the robber may move at its turn; we still refer to its position by r . Even after the robber moves, it holds $r \in V_R \setminus \{v'\}$: there is no edge the robber could use to reach V_C since all cut-edges are protected, and v' is occupied by the black cop.

Before the next round of the game, let us now reapply the method used to obtain the partition on G_{t+1} . Let $x = v'$ stand for the black cop's current position, and set $T = T'$. Consider again the unique black path from x to r , and denote by (x, x') its first edge. By construction of T , there is a unique black path from x to all vertices of V_C . Hence, if T is split as before into two subtrees after removing edge (x, x') , the resulting subtree T_x contains x and also all vertices of former V_C (and possibly more vertices). Then, let us set $V_C = V(T_x)$ and $V_R = V \setminus V_C = V(T_{x'})$. As there is still one cop on one extremity of each blue edge, the vertices of V_C are unreachable by the robber.

Let us now consider the very first step. We start from an arbitrary spanning tree, denoted by T , whose edges are the black ones, the other being the blue ones. For the initial positions, let us place one cop at one extremity of each blue edge. One last cop, the *black cop*, is placed on some other (blue-cop free) vertex, say $x \in V$. Then, the robber chooses a cop-free vertex, say $r \in V$, for its initial place. Edge (x, x') , and sets V_C and V_R are similarly defined, hence the vertices of V_C are unreachable by the robber. The cardinality of V_R is at most $n - 1$.

If we inductively apply the above method for the cops, it follows that after each round, the number of vertices of V_R , which contains the vertices reachable by the robber, is strictly decreased. It will eventually reach the value of zero and the robber will be captured in at most n rounds. \square

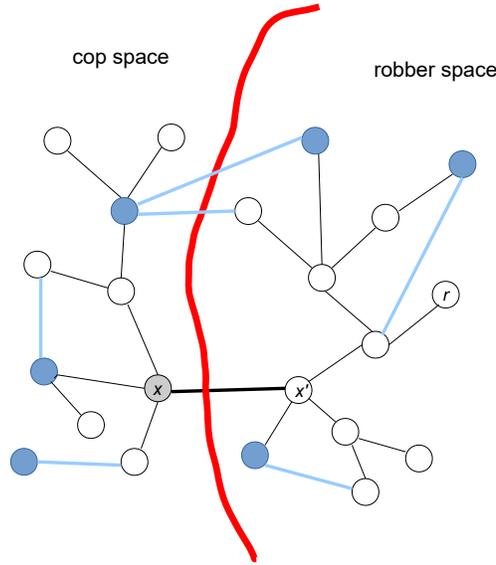


Fig. 1: A depiction for the proof of Theorem 4. The black cop lies on x , with x' at the other side of the cut. Colored vertices/edges indicate blue cops/edges.

The above result provides a better upper bound than the easy to see $c_t(G) \leq n - 1$, for sparse graphs when $\lambda \leq n - 3$. For $\lambda = 1$, cycle graphs are a tight example. We demonstrate the result is nearly tight for certain graph families, see Theorem 5 in the next part of this section.

A nearly tight graph family for sparse graphs. We hereby consider the graph family $\mathbb{G} = \{G_\lambda \mid \lambda \bmod 2 = 1 \wedge \lambda \geq 5\}$ for sufficiently large *odd* values of $\lambda \in \mathbb{N}$. We define the vertex set as $V(G_\lambda) = \{v_1, v_2, \dots, v_{2\lambda-2}, v'_1, v'_2, \dots, v'_{2\lambda-2}\}$. For $i = 1, 2, \dots, 2\lambda - 2$, let $(v_i, v_{(i+1) \bmod (2\lambda-2)}) \in E(G_\lambda)$ and $(v_i, v'_i) \in E(G_\lambda)$. Note, we assume $v_0 \equiv v_{2\lambda-2}$ for the use of modulo above. Also, for $i = 1, 3, 5, \dots, 2\lambda - 3$, let $(v'_i, v'_{i+1}) \in E(G_\lambda)$. Overall, it holds $n = |V(G_\lambda)| = 4\lambda - 4$ and $m = |E(G_\lambda)| = 2(2\lambda - 2) + (2\lambda - 2)/2 = 5(2\lambda - 2)/2 = 5\lambda - 5 = n - 1 + \lambda$. Intuitively, G_λ is a cycle on $2\lambda - 2$ vertices where another $\lambda - 1$ disjoint 4-cycles are attached. An example depiction is given in Figure 2. Notice that G_λ becomes a tree by the removal of λ edges, for example, the $\lambda - 1$ edges (v'_i, v'_{i+1}) , for $i = 1, 3, 5, \dots, 2\lambda - 3$ and another edge (v_j, v_{j+1}) for some $j \in \{1, 2, \dots, 2\lambda - 3\}$.

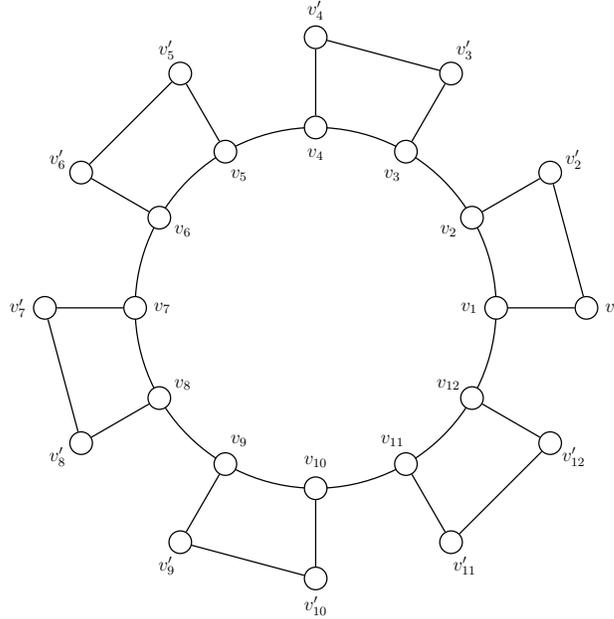


Fig. 2: The graph $G_7 \in \mathbb{G}$.

Theorem 5. For any $G_\lambda \in \mathbb{G}$, it holds $c_t(G_\lambda) = \lambda - 1$.

This theorem is a direct consequence of the two lemmata that follow, which demonstrate the corresponding (worst-case) upper and lower bound strategies.

Lemma 2. For any $G_\lambda \in \mathbb{G}$, it holds $c_t(G_\lambda) \leq \lambda - 1$.

Proof: We present a strategy for $\lambda - 1$ cops to win against the robber under any dynamics and/or robber strategy. Initially, the $\lambda - 1$ cops are placed as follows: place one cop at v_i for each $i = 2, 6, 10, \dots, 2\lambda - 4$

and for each $i = 3, 7, 11, \dots, 2\lambda - 3$. To verify, since there are two sequences of cops using a distance 4 step, overall the number of cops is $2(2\lambda - 2)/4 = \lambda - 1$. For an example placement on G_7 , see Figure 3. Then, the robber places itself at some cop-free vertex. By symmetry of G_λ and cop placement, without loss of generality, we assume the robber places itself on some vertex in $R := \{v_4, v_5, v'_3, v'_4, v'_5, v'_6\}$. In the cop strategy we will now propose, the robber will never be able to escape this set of vertices. Therefore, we restrict the proof to the subgraph induced by $\{v_2, v_3, v_4, v_5, v_6, v_7, v'_3, v'_4, v'_5, v'_6\}$, see Figure 4a, and will demonstrate how the four cops in this subgraph can always capture a robber with an initial placement within R . For all robber turns below, we assume the robber always remains within R ; by our strategy, it is impossible for the robber to move outside R since it would mean “jumping” over a cop.

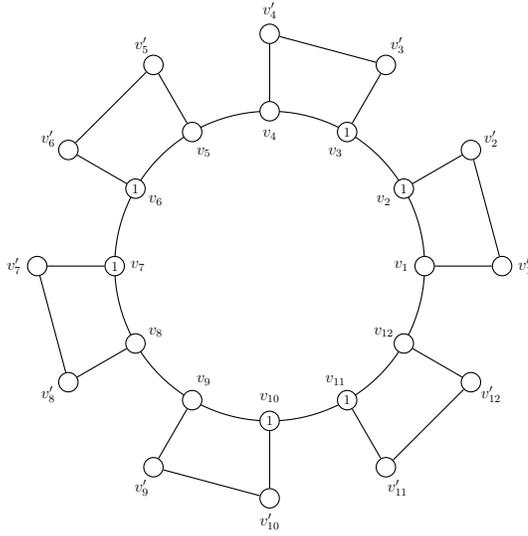


Fig. 3: The initial positions for cops in graph $G_7 \in \mathbb{G}$. In all figures, an integer within a vertex stands for the number of cops currently placed on the vertex.



(a) Initial positions for cops in subgraph. (b) After the first move: a cop moved from v_7 to v_6 .

Fig. 4: The first move of the cop strategy

The cops’ strategy is the following. Since the instance needs to be connected at each time step, at least one edge in $\{(v_2, v_3), (v_6, v_7)\}$ is available. By symmetry, without loss of generality, assume (v_6, v_7) is present and the cop on v_7 moves to v_6 , see Figure 4b. It suffices to prove that the cops have a winning

strategy starting from this configuration.

In the next round (following Figure 4b), the cops move as follows. If (v_6, v'_6) is available, then one of the cops moves to v'_6 (Case 1, Figure 5a). Otherwise, if (v_6, v'_6) is not available, in order to ensure connectivity of the instance, either (v_6, v_5) is available, and one cop moves to v_5 (Case 2, Figure 6a), or (v_2, v_3) is available and one cop moves to v_3 , so that two cops lie on v_3 (Case 3, Figure 7a). In all cases, the robber takes its turn in R .

In Case 1, at least one cop can move to vertex v'_5 (Case 1a, Figure 5b), v_5 (Case 1b, Figure 5c), or v_3 (Case 1c, Figure 5d) by connectivity of the instance. In Case 1a, one cop can move either to v_5 or to v_3 , otherwise the instance is disconnected. If a cop moves to v_5 , the robber lies within $\{v_4, v'_3, v'_4\}$ and at the next step a cop traverses either (v_5, v_4) (hence the robber is trapped on v'_3 or v'_4 and loses within the next two rounds) or (v_2, v_3) (hence at the next round either a cop arrives at v_4 and the robber is blocked or a cop arrives at v'_3 and the robber loses at the following round). If starting from Case 1a a cop moves to v_3 , then in the next round a cop can move to $\{v_5, v_4, v'_3\}$: if a cop moves to v_5 , in the next round a cop can move to $\{v_4, v'_3\}$ and the robber loses in at most two rounds, otherwise, if a cop moves to v_4 or v'_3 , the robber again loses in at most two rounds regardless its strategy. In Case 1b, the robber loses if it was on v'_5 . Otherwise, a cop can traverse either (v_5, v_4) or (v_2, v_3) and the rest of the strategy is the same as for Case 1a. In Case 1c, at least one cop can move to a vertex in $\{v_5, v'_5, v_4, v'_3\}$. Similarly to above, in all subcases, it is easy to see that the cops win in at most two steps against any robber position/movement.

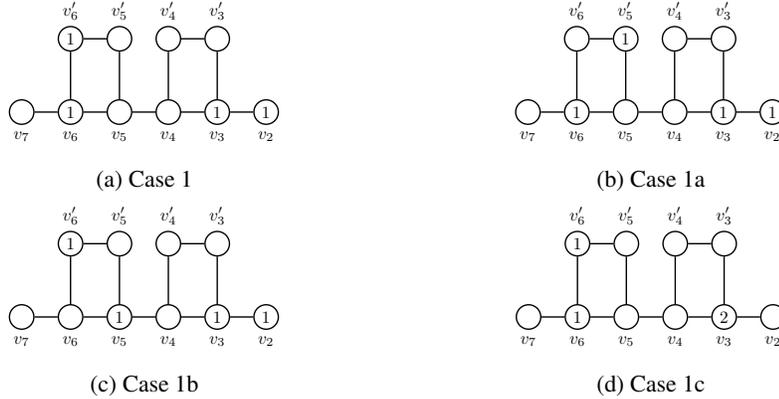


Fig. 5: Case 1 analysis for the proof of Lemma 2

In Case 2, if the robber lies on v'_5 or v'_6 , it is easy to see the cops win in at most two steps, since connectivity to the rest of the graph must be maintained. Instead, if the robber lies in $\{v_4, v'_3, v'_4\}$, then a cop traverses either (v_5, v_4) (Case 2a, Figure 6b) or (v_2, v_3) (Case 2b, Figure 6c). In Case 2a, we arrive to 4-cycle $\{v_3, v_4, v'_3, v'_4\}$ with cops on v_4 and v_3 , and as before, the cops win in at most two steps. In Case 2b, again by preservation of connectivity, at least one edge in $\{(v_5, v_4), (v_3, v_4), (v_3, v'_3)\}$ must be available and one cop moves to either v_4 or v'_3 . In either case, the cops win in at most another two steps.

In Case 3, before they take this new turn, two cops lie on v_6 and another two cops lie on v_3 . By preservation of connectivity, at least one cop can reach a vertex in $\{v'_6, v_5, v_4, v'_3\}$. By symmetry, it suffices to consider the worst-case scenarios where a cop arrives on v'_6 (Case 3a, Figure 7b) and v_5 (Case

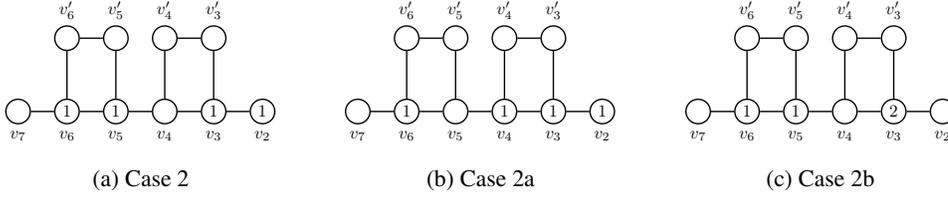


Fig. 6: Case 2 analysis for the proof of Lemma 2

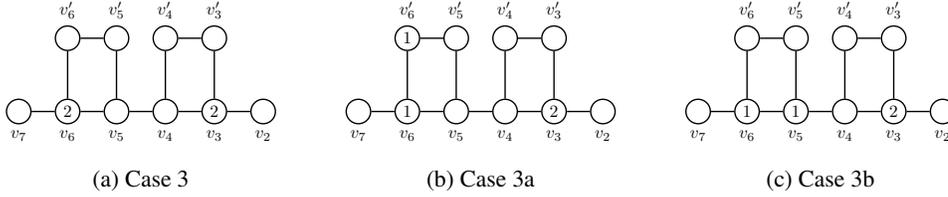


Fig. 7: Case 3 analysis for the proof of Lemma 2

3b, Figure 7c). We notice Case 3a is the same as Case 1c and Case 3b the same as Case 2b. \square

To help us with the matching lower bound to follow, we hereby provide some useful definitions and claims on cop movement restrictions on G_λ incurred by worst-case dynamics. From now on, all vertex indices are assumed to be modulo $2\lambda - 2$ with $v_0 \equiv v_{2\lambda-2}$. For $i = 1, 2, \dots, \lambda - 1$, let *loop* L_i refer to the 4-cycle with $V(L_i) = \{v_{2i-1}, v_{2i}, v'_{2i-1}, v'_{2i}\}$ and let its edge-set be defined as $E(L_i) = \{(v_{2i-1}, v_{2i}), (v'_{2i-1}, v'_{2i}), (v_{2i-1}, v'_{2i-1}), (v_{2i}, v'_{2i})\}$. We say that L_i is *cop-occupied* if at least one cop lies at some vertex in $V(L_i)$, otherwise, L_i is *cop-free*. We say that a cop *crosses* L_i if, starting from vertex v_{2i-1} (*cross-start* vertex), it can eventually arrive to vertex v_{2i} (*cross-end* vertex), or vice versa. We refer to a (counterclockwise-movement) crossing from v_{2i-1} to v_{2i} as a *cc-crossing* and to a (clockwise-movement) crossing from v_{2i} to v_{2i-1} as a *c-crossing*. A cop trivially cc-crosses L_i if it already lies on v_{2i} or v_{2i+1} , i.e., the counterclockwise neighbor of v_{2i} . Respectively, a cop trivially c-crosses L_i if it already lies on v_{2i-1} or v_{2i-2} , i.e., the clockwise neighbor of v_{2i-1} . The intuition behind Proposition 3 is that, while a number of cops crosses a loop, at least one of them must *stay behind*, that is, will not be able to ever cross the loop due to worst-case dynamics.

Proposition 3. *Assume we focus on a given loop $L_i \subset V(G_\lambda)$ and at most one edge in $E(L_i)$ is not present at each time step of evolution. In the worst case, at most ρ cops can cross L_i , if $\rho + 1$ cops are present at the cross-start vertex.*

Proof: Without loss of generality, consider loop $L_1 = \{v_1, v_2, v'_1, v'_2\}$ and suppose $\rho + 1$ cops lie on v_1 and wish to cross to v_2 . The dynamics of the graph evolve as follows: for each t , if at the end of round t there is at least one cop on v_1 , then $(v_1, v_2) \notin E_{t+1}$. Otherwise, $(v'_1, v'_2) \notin E_{t+1}$. In other words, as long as there is a cop on v_1 , the edge to v_2 is blocked. The cops could take advantage of this situation such that at most ρ of them reach v_2 via the available path v_1, v'_1, v'_2, v_2 . If at any time v_1 is cop-free, then the above path is blocked and (v_1, v_2) is available, however no cop is there to traverse it and cross the loop. The last remaining cop cannot cross since it would mean that, at some point in time, either v_1

is cop-occupied and (v_1, v_2) is available or v_1 is cop-free and (v'_1, v'_2) is available, a contradiction to the specified dynamics. \square

Assume strictly fewer than $\lambda - 1$ cops initially place themselves at the vertices of G_λ . Since there are $\lambda - 1$ loops, there exists at least one cop-free loop L_i . In general, after the cops are initially positioned, G_λ can be partitioned into alternating sequences of cop-occupied and cop-free loops L_i . Let O_1, \dots, O_p , respectively F_1, \dots, F_p , stand for the sequences of cop-occupied, respectively cop-free loops, where F_1 is set arbitrarily, and we assume F_i is between O_i (clockwise) and O_{i+1} (counterclockwise). Moreover, for $i = 1, 2, \dots, p$, let $|F_i| = f_i$ and $|O_i| = o_i$. The cardinality p of the two sequence sets is the same, since two non-maximal adjacent cop-occupied subsequences, i.e., with no cop-free loop between them, form one bigger cop-occupied sequence; a similar observation holds for cop-free sequences. By the reasoning above, it holds $p \geq 1$. An example initial placement on G_7 is given in Figure 8: The sequences of cop-occupied and cop-free loops formed are $O_1 = \{L_6, L_1, L_2\}$, $F_1 = \{L_3\}$, $O_2 = \{L_4\}$, and $F_2 = \{L_5\}$.

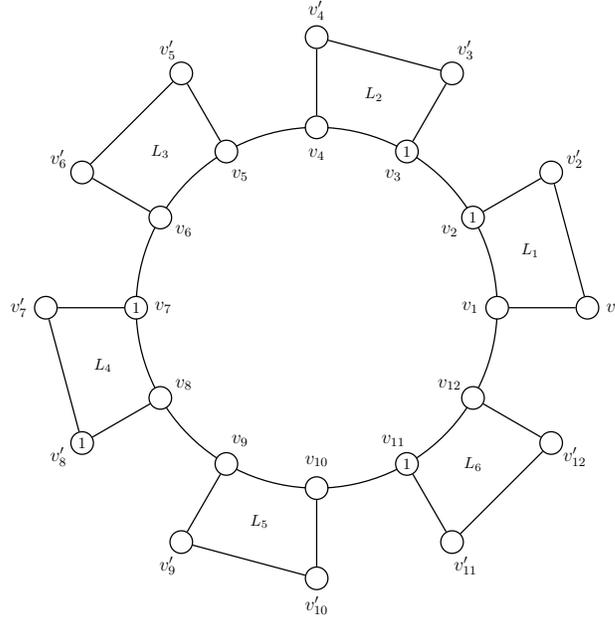


Fig. 8: An example initial placement of $\lambda - 2 = 5$ cops on G_7 .

The following proposition provides us with a necessary condition in order for the cops to win against a robber placed on some cop-free sequence of loops. For a sequence $F = \{L_1, L_2, \dots, L_f\}$, let $V(F) = \cup_{i=1}^f V(L_i)$.

Proposition 4. *Let $F = \{L_1, L_2, \dots, L_f\}$ be a cop-free sequence of loops with the robber lying on some vertex within $V(F)$ not adjacent to a cop. Let L_{cc} , respectively L_c , stand for the cop-occupied loop adjacent to the counter clockwise of F , respectively to the clockwise of F . At least $f + 1$ cops must be able to c -cross L_{cc} , and another $f + 1$ cops to cc -cross L_c , in order for the cops to win.*

Proof: In contradiction, and without loss of generality, assume whenever $f + 1$ cops can cc-cross L_c at most f cops can c-cross L_{cc} . Assume L_f is clockwise adjacent to L_{cc} , L_i is clockwise adjacent to L_{i+1} for $i = 1, 2, \dots, f - 1$, and L_c is clockwise adjacent to L_1 ; see Figure 9. From now on, consider that the dynamics of the graph force the single edge connecting L_c to L_1 to be unavailable in all graph instances. So, no cop may leave L_c and reach L_1 in F by moving counterclockwise and each graph instance remains connected.

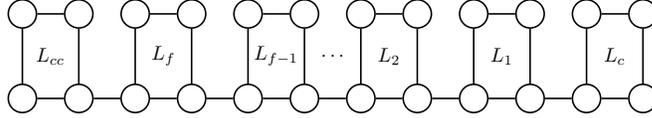


Fig. 9: Loop numbering for a cop-free sequence $F = \{L_1, L_2, \dots, L_f\}$. Left stands for counterclockwise direction in G_λ , respectively right for clockwise direction.

Having crossed L_{cc} , the f cops may all move to L_f . By Proposition 3, at most $f - 1$ cops can c-cross L_f . For some $i \geq 1$, assume $f - i$ cops have c-crossed L_{f+1-i} . Then, by Proposition 3, at most $f - i - 1 = f - (i + 1)$ cops can c-cross $L_{f+1-(i+1)}$. Overall, by induction, at most $f - i$ cops are able to c-cross L_{f+1-i} . For $i = f - 1$, at most $f - (f - 1) = 1$ cop is able to c-cross L_2 .

To win, the robber has a feasible strategy to evade indefinitely, that is, to be placed at the vertex in L_1 connected by the (always unavailable) edge to L_c as discussed above. No cop can arrive to L_1 from L_c due to the missing edge, and since at most 1 cop can c-cross L_2 , no cop can ever c-cross L_1 . \square

Now, we are ready to show, in Proposition 5, how the robber can identify a cop-free sequence to employ the winning strategy demonstrated in Proposition 4.

For integers i, w_i, w'_i , where $1 \leq i \leq \lambda - 1$, $0 \leq w_i \leq p - 1$, $0 \leq w'_i \leq p - 1$, let $F_c(i, w_i) = \{F_i, F_{i-1}, \dots, F_{i-w_i}\}$ stand for the set including F_i and the w_i cop-free sequences nearer to F_i in clockwise fashion, and $F_{cc}(i, w'_i) = \{F_i, F_{i+1}, \dots, F_{i+w'_i}\}$ stand for the set including F_i and the w'_i cop-free sequences nearer to F_i in counterclockwise fashion. In a similar manner, for the cop-occupied sequences, let $O_c(i, w_i) = \{O_i, O_{i-1}, \dots, O_{i-w_i}\}$ and $O_{cc}(i, w'_i) = \{O_i, O_{i+1}, \dots, O_{i+w'_i}\}$. For a cop-occupied sequence O_i , we say that $o_i = |O_i|$ cops (choosing one per loop in O_i) are its *occupant* cops. If strictly more than o_i cops lie at vertices of O_i , then this surplus of cops are referred to as *extra* cops.

Proposition 5. *If there exists a cop-free sequence F_i in G_λ such that at least one of the following holds:*

- (a) *strictly fewer than $\sum_{F_j \in F_c(i, w_i)} f_j$ extra cops lie within vertices in $O_c(i, w_i)$, for all integers w_i , where $0 \leq w_i \leq p - 1$,*
- (b) *strictly fewer than $\sum_{F_j \in F_{cc}(i, w'_i)} f_j$ extra cops lie within vertices in $O_{cc}(i + 1, w'_i)$, for all integers w'_i , where $0 \leq w'_i \leq p - 1$,*

then the robber wins.

Proof: Assume the cops are initially placed and the robber is able to identify a cop-free sequence $F_i = \{L_1, \dots, L_{f_i}\}$, where, for $i = 1, 2, \dots, f_i - 1$, L_i is clockwise adjacent to L_{i+1} , L_c is clockwise adjacent to L_1 and L_{f_i} is clockwise adjacent to L_{cc} , like in Figure 9, such that at least one of (a) and (b) holds.

Without loss of generality, we hereby consider only case (a); the other case follows in similar manner by symmetry.

The robber strategy is simply to place itself at vertex v in L_{f_i} , which is adjacent to vertex u in L_{cc} and, from now on, consider that the dynamics always force edge (v, u) to be unavailable. We now show that no cop can cc-cross L_{f_i} , therefore the robber wins.

For $w_i = 0$, strictly fewer than f_i extra cops lie in $O_c(i, 0) = \{O_i\}$. Hence, by using only cops within $O_c(i, 0)$, at most f_i cops can cross L_c : the extra cops in O_i and, by Proposition 3, at most one occupant cop, e.g., the one occupying L_c by its initial placement. By Proposition 4, the robber wins.

Inductively, assume that, for some w where $0 \leq w < p - 1$, for all $w_i \leq w$ it holds that, by only using cops within $O_c(i, w_i)$, at most f_i cops can cross L_c . If we consider $w + 1$, by assumption, for $w_i = 0, 1, \dots, w + 1$, strictly fewer than $\sum_{F_j \in F_c(i, w_i)} f_j$ extra cops lie within $O_c(i, w_i)$. Let $f' < \sum_{F_j \in F_c(i, w)} f_j$ be the number of extra cops within $O_c(i, w)$ and $f'' < \sum_{F_j \in F_c(i, w+1)} f_j$ be the number of extra cops within $O_c(i, w + 1)$. Then, $f^* = f'' - f' \geq 0$ extra cops lie within $O_c(i, w + 1) \setminus O_c(i, w) = \{O_{i-(w+1)}\}$. We consider two cases:

- If $f^* \geq f_{i-(w+1)}$, by inductively applying Proposition 3, move $f_{i-(w+1)}$ out of the f' extra cops in $O_{i-(w+1)}$ to become occupant cops of all the free loops in $F_{i-(w+1)}$. Since there is no free loop left between the original sequences O_{i-w} and $O_{i-(w+1)}$, we now reset O_{i-w} to a larger sequence O'_{i-w} containing all loops in $O_{i-w}, F_{i-(w+1)}, O_{i-(w+1)}$. The number of extra cops within vertices in $O'_c(i, w) = O_c(i, w) \setminus \{O_{i-w}\} \cup \{O'_{i-w}\}$ is $f'' - f_{i-(w+1)} < \sum_{F_j \in F_c(i, w+1)} f_j - f_{i-(w+1)} = \sum_{F_j \in F_c(i, w)} f_j$. By the inductive assumption for w , by only using cops within $O'_c(i, w)$, at most f_i cops can cross L_c . Since at most f_i cops can cross L_c for all possible $w_i \leq w$, then by Proposition 4, the robber wins.
- If $f^* < f_{i-(w+1)}$, by inductively applying Proposition 3, move all f^* extra cops in $O_{i-(w+1)}$ to become occupant cops of some free loops in $F_{i-(w+1)}$. Let F^* , where $|F^*| = f^*$, denote the formerly free loops which are now occupied. We now reset $O_{i-(w+1)}$ to a larger sequence $O'_{i-(w+1)}$ containing all loops in F^* and $O_{i-(w+1)}$. We wish to use cops only within $O'_c(i, w+1) = O_c(i, w+1) \setminus \{O_{i-(w+1)}\} \cup \{O'_{i-(w+1)}\}$. Since no extra cops remain within $O'_{i-(w+1)}$, we focus only on the number of extra cops within $O_c(i, w)$. By assumption, their number is $f' < \sum_{F_j \in F_c(i, w)} f_j$ and the proof follows as in the first case.

□

Proposition 6. *Assume strictly fewer than $\lambda - 1$ cops are initially placed on G_λ . Then, there exists a cop-free sequence F_i in G_λ such that at least one of conditions (a) and (b) in Proposition 5 holds.*

Proof: Since strictly fewer than $\lambda - 1$ cops are initially placed on G_λ , and G_λ has exactly $\lambda - 1$ loops L_i , then by pigeonhole principle there exists at least one loop with no cop on its vertices, and so at least one cop-free sequence in G_λ .

By contradiction, suppose that for every cop-free sequence F_i in G_λ (i) there exists w_i such that at least $\sum_{F_j \in F_c(i, w_i)} f_j$ extra cops lie within $O_c(i, w_i)$, and (ii) there exists w'_i such that at least $\sum_{F_j \in F_{cc}(i, w'_i)} f_j$ extra cops lie within $O_{cc}(i + 1, w'_i)$. Consider some cop-free sequence, say F_{i_1} , where $i_1 = 1$ without loss of generality. By (ii), there exists some (minimum-value) w_{i_1} such that at least $\sum_{F_j \in F_{cc}(i_1, w_{i_1})} f_j$

extra cops lie within $O_{cc}(i_1 + 1, w_{i_1})$. Let $F_{i_2} = F_{i_1+1+w_{i_1}}$ be the first cop-free sequence to the counterclockwise of $O_{cc}(i_1 + 1, w_{i_1})$. Then, by (ii), there exists some (minimum-value) w_{i_2} such that at least $\sum_{F_j \in F_{cc}(i_2, w_{i_2})} f_j$ extra cops lie within $O_{cc}(i_2 + 1, w_{i_2})$. We proceed with such statements, inductively, until we reach F_{i_l} , for which there exists (minimum-value) w_{i_l} such that at least $\sum_{F_j \in F_{cc}(i_l, w_{i_l})} f_j$ extra cops lie within $O_{cc}(i_l + 1, w_{i_l})$ and $i_l + 1 + w_{i_l} \geq p + i_1$. That is, we have performed a full round on G_λ . There are three cases to consider with respect to the value $i_l + 1 + w_{i_l}$.

- If $i_l + 1 + w_{i_l} = p + i_1$, then, for $i_z = i_1, i_2, \dots, i_l$, sets $O_{cc}(i_z + 1, w_{i_z})$ form a partition of the cop-occupied space in G_λ . By assumption, for each such i_z , at least $\sum_{F_j \in F_{cc}(i_z, w_{i_z})} f_j$ extra cops lie within $O_{cc}(i_z + 1, w_{i_z})$. Summing it all, $\sum_{z=1}^l \sum_{F_j \in F_{cc}(i_z, w_{i_z})} f_j = \sum_{j=1}^p f_j$ extra cops lie within the cop-occupied sequences, since $\cup_{z=1}^l F_{cc}(i_z, w_{i_z})$ contains all cop-free loops in G_λ .
- If $i_l + 1 + w_{i_l} = p + i_y$, for some $y > 1$, then the last interval fully covers some already defined intervals starting at i_1, i_2, \dots, i_{y-1} . In this case, for $i_z = i_y, i_{y+1}, \dots, i_l$, sets $O_{cc}(i_z + 1, w_{i_z})$ form a partition of the cop-occupied space in G_λ . By assumption, for each such i_z , at least $\sum_{F_j \in F_{cc}(i_z, w_{i_z})} f_j$ extra cops lie within $O_{cc}(i_z + 1, w_{i_z})$. Summing it all together as in the previous case, at least $\sum_{z=y}^l \sum_{F_j \in F_{cc}(i_z, w_{i_z})} f_j = \sum_{j=1}^p f_j$ extra cops lie within the cop-occupied sequences.
- If $p + i_y < i_l + 1 + w_{i_l} < p + i_{y+1}$, for some $y > 1$, then the last interval fully contains intervals starting at i_1, i_2, \dots, i_{y-1} and partially overlaps with interval i_y . Let $i_l + 1 + w_{i_l} = p + i_y + x$ for some $1 \leq x \leq i_{y+1} - i_y$. There are strictly fewer than $f_{i_y} + f_{i_{y+1}} + \dots + f_{i_{y+x-1}}$ extra cops within vertices in $O_{i_{y+1}}, \dots, O_{i_{y+x}}$, otherwise the choice of w_{i_y} would not be minimum. As an implication, there are at least $f_{i_{y+x}} + \dots + f_{i_{y+w_{i_y}}}$ extra cops within $O_{i_{y+x+1}}, \dots, O_{i_{y+w_{i_y}+1}}$. Also, by assumption, at least $f_{i_l} + f_{i_{l+1}} + \dots + f_1 + \dots + f_{i_{y+x-1}}$ extra cops lie within vertices in $O_{i_{l+1}}, O_{i_{l+2}}, \dots, O_1, \dots, O_{i_{y+x}}$. Overall, at least $f_{i_l} + f_{i_{l+1}} + \dots + f_1 + \dots + f_{i_{y+x-1}} + f_{i_{y+x}} + \dots + f_{i_{y+w_{i_y}}}$ extra cops lie within $O_{i_{l+1}}, O_{i_{l+2}}, \dots, O_1, \dots, O_{i_{y+x}}, O_{i_{y+x+1}}, \dots, O_{i_{y+w_{i_y}+1}}$. For the rest of the graph, for $z = y + 1, \dots, l - 1$, at least $\sum_{F_j \in F_{cc}(i_z, w_{i_z})} f_j$ extra cops lie within $O_{cc}(i_z + 1, w_{i_z})$. Summing it all together, at least $\sum_{j=1}^p f_j$ extra cops lie within the cop-occupied sequences, since each f_j is considered once in the above calculations.

In all three cases, considering occupant cops and extra cops together, it follows there are at least $\sum_{i=1}^p (o_i + f_i) = \lambda - 1$ cops in G_λ , since the number of loops in all the sequences is exactly the number of loops in G_λ . \square

Lemma 3. For any $G_\lambda \in \mathbb{G}$, it holds $c_t(G_\lambda) \geq \lambda - 1$.

Proof: Follows by the combination of Propositions 5 and 6. \square

5 Conclusions

In this paper, we consider the topic of playing Cops and Robbers games on dynamic graphs. We show how the cop number can be computed in the offline case, where all graph dynamics are known a priori, via a reduction to a reachability game. In the online case with a connectedness restriction, we show a nearly tight bound on the cop number of a family of sparse graphs.

In the future, considering the online case, we would like to tighten the bound for sparse graphs, and also consider dense graphs.

References

- [1] Isolde Adler, Georg Gottlob, and Martin Grohe. Hypertree width and related hypergraph invariants. *European Journal of Combinatorics*, 28(8):2167–2181, 2007.
- [2] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.
- [3] Thomas Andreae. Note on a pursuit game played on graphs. *Discrete Applied Mathematics*, 9(2):111–115, 1984.
- [4] Alessandro Berarducci and Benedetto Intrigila. On the cop number of a graph. *Advances in Applied Mathematics*, 14(4):389–403, 1993.
- [5] Dietmar Berwanger. Graph games with perfect information. *preprint*, 2013.
- [6] Lélia Blin, Pierre Fraigniaud, Nicolas Nisse, and Sandrine Vial. Distributed chasing of network intruders. *Theoretical Computer Science*, 399(1-2):12–37, 2008.
- [7] Béla Bollobás, Gábor Kun, and Imre Leader. Cops and robbers in a random graph. *Journal of Combinatorial Theory, Series B*, 103(2):226–236, 2013.
- [8] Anthony Bonato. *The game of cops and robbers on graphs*. American Mathematical Soc., 2011.
- [9] Anthony Bonato, Ehsan Chiniforooshan, and Paweł Prałat. Cops and robbers from a distance. *Theoretical Computer Science*, 411(43):3834–3844, 2010.
- [10] Anthony Bonato and Gary MacGillivray. Characterizations and algorithms for generalized cops and robbers games. *arXiv preprint arXiv:1704.05655*, 2017.
- [11] Anthony Bonato, Paweł Prałat, and Changping Wang. Pursuit-evasion in models of complex networks. *Internet Mathematics*, 4(4):419–436, 2007.
- [12] Arnaud Casteigts. *A Journey through Dynamic Networks (with Excursions)*. Habilitation à diriger des recherches, Université de Bordeaux, June 2018.
- [13] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.
- [14] Nancy E Clarke and Gary MacGillivray. Characterizations of k-copwin graphs. *Discrete Mathematics*, 312(8):1421–1425, 2012.
- [15] Thomas Erlebach and Jakob T Spooner. A game of cops and robbers on graphs with periodic edge-connectivity. In *International Conference on Current Trends in Theory and Practice of Informatics*, volume 12011, pages 64–75. Springer LNCS, 2020.
- [16] Shannon L Fitzpatrick and Richard J Nowakowski. Copnumber of graphs with strong isometric dimension two. *Ars combinatoria*, 59:65–73, 2001.

- [17] Peter Frankl. Cops and robbers in graphs with large girth and cayley graphs. *Discrete Applied Mathematics*, 17(3):301–305, 1987.
- [18] Geňa Hahn and Gary MacGillivray. A note on k-cop, l-robber games on graphs. *Discrete mathematics*, 306(19-20):2492–2497, 2006.
- [19] Petter Holme and Jari Saramäki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.
- [20] Athanasios Kehagias and Georgios Konstantinidis. Cops and robbers, game theory and Zermelo’s early results. *arXiv preprint arXiv:1407.1647*, 2014.
- [21] Imran Khaliq and Gulshad Imran. Reachability games revisited. *SOFTENG 2016*, page 137, 2016.
- [22] William B Kinnersley. Cops and robbers is exptime-complete. *Journal of Combinatorial Theory, Series B*, 111:201–220, 2015.
- [23] Adrian Kosowski, Bi Li, Nicolas Nisse, and Karol Suchan. k-chordal graphs: From cops and robber to compact routing via treewidth. *Algorithmica*, 72(3):758–777, 2015.
- [24] Linyuan Lu and Xing Peng. On Meyniel’s conjecture of the cop number. *Journal of Graph Theory*, 71(2):192–205, 2012.
- [25] Tomasz Łuczak and Paweł Prałat. Chasing robbers on random graphs: zigzag theorem. *Random Structures & Algorithms*, 37(4):516–524, 2010.
- [26] Malaz Maamoun and Henry Meyniel. On a game of policemen and robber. *Discrete Applied Mathematics*, 17(3):307–309, 1987.
- [27] René Mazala. Infinite games. In *Automata logics, and infinite games*, pages 23–38. Springer, 2002.
- [28] Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016.
- [29] Othon Michail and Paul G Spirakis. Elements of the theory of dynamic networks. *Communications of the ACM*, 61(2):72–72, 2018.
- [30] Nils Morawietz, Carolin Rehs, and Mathias Weller. A Timecop’s Work Is Harder Than You Think. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 71:1–71:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
- [31] Nils Morawietz and Petra Wolf. A Timecop’s Chase Around the Table. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021)*, volume 202 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [32] Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235–239, 1983.

- [33] Paweł Prałat and Nicholas Wormald. Meyniel’s conjecture holds for random graphs. *Random Structures & Algorithms*, 48(2):396–421, 2016.
- [34] A Quillot. *Études de quelques problèmes sur les graphes et hypergraphes et applications à la théorie des jeux à information complète*. PhD thesis, PhD thesis, Université de Paris VI, 1978.
- [35] Alex Scott and Benny Sudakov. A bound for the cops and robbers problem. *SIAM Journal on Discrete Mathematics*, 25(3):1438–1442, 2011.
- [36] Kazuo Sugihara and Ichiro Suzuki. On a pursuit-evasion problem related to motion coordination of mobile robots. In *Proceedings of the Twenty-First Annual Hawaii International Conference on System Sciences, 1988. Vol. IV. Applications Track*, volume 4, pages 218–219. IEEE Computer Society, 1988.