A Cop and Robber Game on Edge-Periodic Temporal Graphs

Thomas Erlebach[a,1], Nils Morawietz[b,2], Jakob T. Spooner[c], Petra Wolf[d,3]

[a]*Department of Computer Science, Durham University, Durham, UK*
[b]*Friedrich Schiller University Jena, Institute of Computer Science, Jena, Germany*
[c]*School of Computing and Mathematical Sciences, University of Leicester, Leicester, UK*
[d]*LaBRI, CNRS, Université de Bordeaux, Bordeaux, France*

Abstract

We introduce a cops and robbers game with one cop and one robber on a special type of time-varying graphs (TVGs), namely edge-periodic graphs. These are TVGs in which, for each edge e, a binary string $\tau(e)$ is given such that the edge e is present in time step t if and only if $\tau(e)$ contains a 1 at position $t \bmod |\tau(e)|$. This periodicity allows for a compact representation of infinite TVGs. We prove that even for very simple underlying graphs, i.e., directed and undirected cycles, the problem of deciding whether a cop-winning strategy exists is NP-hard and W[1]-hard parameterized by the number of vertices. Furthermore, we show that this decision problem can be solved on general edge-periodic graphs in PSPACE. Finally, we present tight bounds on the minimum length of a directed or undirected cycle that guarantees the cycle to be robber-winning.

[⋆]Preliminary versions of different parts of this work have appeared in the proceedings of SOFSEM 2020 [1] and MFCS 2021 [2].

Email addresses: `thomas.erlebach@durham.ac.uk` (Thomas Erlebach), `nils.morawietz@uni-jena.de` (Nils Morawietz), `mail@wolfp.net` (Petra Wolf)

[1]Supported by EPSRC grants EP/S033483/2 and EP/T01461X/1.

[2]The research was performed while the author was associated with Philipps-Universität Marburg, Germany. During this time the author was supported by Deutsche Forschungsgemeinschaft (DFG), project OPERAH, KO 3669/5-1.

[3]Supported by the French ANR, project ANR-22-CE48-0001 (TEMPOGRAL). The research was partially performed while the author was associated with University Trier, Germany. During this time the author was supported by Deutsche Forschungsgemeinschaft (DFG), project FE560/9-1.

Keywords: Time-varying graph, Edge-periodic cycle, Game of cops and robbers, Computational complexity

We dedicate this article to the memory of Rolf Niedermeier.

1. Introduction

Pursuit-evasion games are games played between two teams of players, who take turns moving within the confines of some abstract arena. Typically, one team – the *pursuers* – are tasked with catching the members of the other team – the *evaders* – whose task it is to evade capture indefinitely. The study of such games has led to their application in a number of real-world scenarios, one widely-studied example of which would be their application to the problem of guiding robots through real-world environments [3]. From a theoretical standpoint, other variants of the game have been studied for their intrinsic links to important graph parameters; for example, in one particular variant in which each pursuer can, in a single turn, move to an arbitrary vertex of the given graph G, it is well known that establishing the number of pursuers it takes to catch one evader also establishes the treewidth of G [4].

The variant most closely resembled by the one considered in this paper was first studied separately by Quilliot [5], and by Nowakowski and Winkler [6], as the discrete *cops and robbers* game. In essence, the games these authors considered were the same: one cop (pursuer) and one robber (evader) take turns moving across the edges (or remaining at their current vertex) of a given graph G, with the cop aiming to catch the robber, and the robber attempting to avoid capture. (By 'catching the robber' we mean that the cop is able to occupy the same vertex as the robber within G.) Aigner and Fromme [7] considered a generalized variant of the game, in which k cops attempt to catch a single robber; their paper introduced the notion of the *cop-number* of a graph, i.e., the minimum number of cops required to guarantee that the robber is caught.

Such games have been studied intensively for static graphs [8]. If the game is played with one robber and $k \geq 1$ cops on a given graph, the cops first place themselves on vertices of the graph, before the robber chooses his initial vertex (after observing where the cops have been placed). Then, in each round, the players alternate turns and the cops move first. Here, each cop can move to an adjacent vertex or pass and stay on her vertex. The same holds for the robber. We say that a graph is *k-cop-winning*, if there exists a strategy for the k cops using which they finally catch the robber, i.e., a cop occupies the same vertex as the robber. If the context is clear, we call a 1-cop-winning graph a *cop-winning* graph. If a graph is not cop-winning, we call it *robber-winning*. Special attention has been devoted to the characterization of graphs that are k-cop-winning. While for one cop, the cop-winning graphs where understood in 1978 and independently in 1983 [5, 6] as those featuring a special kind of ordering on the vertex set, called a *cop-win* or *elimination* or *dismantling* order, the case for k cops was long open and solved in 2009 by exploiting a linear structure of a certain power of the graph [9].

In this paper, we introduce a variant of the cops and robbers game with an essentially identical set of rules to the one considered in [5, 6], but broaden the class of viable game arenas to include the *edge-periodic* graphs [10]. As such, we call the game *periodic cop and robber*. Informally, edge-periodic graphs can be thought of as traditional static graphs equipped with an additional function, mapping each edge e to a *label* $\tau(e)$, which is a binary string that dictates in which time steps e is present within each consecutive period of $|\tau(e)|$ time steps. The class of edge-periodic graphs can also be considered a subclass of so-called *time-varying graphs* or *temporal graphs* [11].

In general, a *time-varying graph* (TVG) describes a graph that varies over time. For most applications, this variation is limited to the availability or weight of edges, meaning that edges are only present at certain time steps or the time needed to cross an edge changes over time. TVGs are of great interest in the area of *dynamic networks* [10, 12, 13, 14] such as *mobile ad hoc networks* [15] and *vehicular networks* modeling traffic load factors on a road network [16]. In those networks, the topology naturally changes over time, and TVGs are used to reflect this dynamic behavior. Quite recently, TVGs have attracted interest in the context of graph games such as competitive diffusion games and Voronoi games [17]. There are plenty of representations for TVGs in the literature, which are not equivalent in general. For instance, in [10] a TVG is defined as a tuple $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$ where V is a set of vertices, $E \subseteq V \times V \times L$ is a set of labeled edges (with labels from a set L), $\mathcal{T} \subseteq \mathbb{T}$ is the *lifetime* of the graph, $\mathbb{T}$ is the temporal domain and assumed to be $\mathbb{N}$ for discrete systems and $\mathbb{R}^+$ for continuous systems, $\rho\colon E \times \mathcal{T} \to \{0, 1\}$ is the *presence function* indicating whether an edge e is present in time step t, and $\zeta\colon E \times \mathcal{T} \to \mathbb{T}$ is the latency function indicating the time needed to cross edge e in time step t. We call the graph $G = (V, E)$ the *underlying graph* of $\mathcal{G}$. As of yet, there is no agreement in the literature on how the functions ρ and ζ are given in the input. In the context of computational complexity, this is of significant importance, particularly when ρ exhibits periodicity with respect to single edges. As we are concerned with the computational complexity of determining whether an edge-periodic graph is cop-winning, we now discuss the issue of input representation for temporal graphs with periodicity in more detail. In analogy with class 8 defined in [10], but without requiring the underlying graph G to be connected, we say that a TVG belongs to the class of TVGs featuring *periodicity of edges* if $\forall e \in E, \forall t \in \mathcal{T}, \forall k \in \mathbb{N}, \rho(e,t) = \rho(e, t + kp_e)$ for some $p_e \in \mathbb{T}$ depending on e. For such TVGs with discrete time steps, the function ρ can be represented for each edge $e \in E$ as a binary string of size p_e concatenating the values of $\rho(e,t)$ for $0 \leq t < p_e$. Note that the period of the whole graph $\mathcal{G}$, also called the *global period*, is then the least common multiple (lcm for short) of all string lengths p_e describing edge periods. Therefore, the global period can be exponential in the size of the input, and the underlying graph G of $\mathcal{G}$ can have exponentially many different sub-graphs G_t representing the snapshot of $\mathcal{G}$ at time t. This exponential blow-up is a huge challenge in determining the precise complexity of problems for TVGs featuring periodicity of edges, as discussed in more detail in Section 4 and 6, but it can also lead to more structure that can be exploited algorithmically, as shown in [18]. Often, for general TVGs, a representation containing all sub-graphs representing snapshots over the whole lifetime of the graph is chosen when the complexity of decision problems over TVGs is considered [19, 20]. An approach to unify the representation of TVGs is given in [21], also including the existence of vertices being affected over time. This approach represents $\rho(e,t)$ by enhancing an edge $e = (u,v)$ with the departure time t_d at u and the arrival time t_a at v, where t_a might be smaller than t_d in order to model periodicity. For TVGs with periodicity of edges where ρ is represented as a binary string for each edge, the periodicity of the TVG $\mathcal{G}$ might be exponential in its representation. Therefore, using the approach of [21] would cause an exponential blow-up in the representation of $\mathcal{G}$, as a decrement of the time value could only be used after a whole period of the graph, rather than after the period of one edge. Another class of TVGs based on periodicity was considered in the field of robotics to model motion planning tasks when time dependent obstacles are present [22]. There, the availability of the vertices in the graph changes periodically and each edge needs a constant number of time steps to be crossed. An edge $e = \{u,v\}$ is only present if, in the time span needed to cross e, both endpoints u and v are continuously present. In [22] the periodic function describing the availability of a vertex and the function describing the time needed to cross an edge are represented by an on-line program and can hence handle values exponential in their representation. This is crucial in the PSPACE-hardness proof of the reachability problem for graphs in this class presented in [22]. There, the hardness is obtained by a reduction from the halting problem for linear space-bounded deterministic Turing machines where a configuration of the Turing machine is encoded in the time step. In the reduction, the periodicity of a single vertex as well as the time needed to cross an edge is of value exponential in the tape length-bound. Note that this representation of periodicity is exponentially more compact than in our setting and thus the result of [22] does not translate to our setting.

We will stick in the following to the model describing TVGs featuring periodicity of edges where the function $\rho(e,t)$ is represented as a binary string $\tau(e)$. We refer to such TVGs as *edge-periodic graphs*.

As mentioned earlier, in this paper we introduce and study a cops and robbers game for edge-periodic graphs. After the first announcement of an extended abstract of part of the present work that introduced this game and showed that one can decide if a graph is cop-winning in exponential time via a reduction to a reachability game [1], Balev et al. [23] studied the cop and robber game for TVGs with finite lifetime where each snapshot is given explicitly. They showed that deciding if a game is cop-winning can be done in polynomial time in this case, via reduction to a reachability game similar to the one mentioned above. In their case, contrary to edge-periodic graphs, the reachability game is of polynomial size. They also study the number of cops required to catch the robber in an online variant of the game where the cop does not know the graph of the next time step, while the robber can determine what the graph in the next time step is, the only requirement being that the graph must be connected. Then it was shown by Morawietz, Rehs, and Weller [24] that deciding if an instance of the game with a single cop is cop-winning is NP-hard for edge-periodic graphs whose underlying graph has a constant-size vertex cover or where only two edges have to be removed to obtain a cycle. Moreover, they showed that the problem is W[1]-hard when parameterized by the size of the underlying graph G even in these restricted cases, implying that there is presumably no algorithm solving the problem in time $f(n+m) \cdot |I|^{O(1)}$ for any computable function f, where $|I|$ is the size of the input and n and m represent the number of vertices and edges, respectively, of G. In other words, the exponential growth of the running time of every algorithm solving the problem has to depend on the lengths of the binary strings describing $\rho(e,t)$. Subsequently, in the first announcement of an extended abstract of another part of the present work [2], it was shown that deciding if an instance of the game is cop-winning is NP-hard (and W[1]-hard when parameterized by the size of the underlying graph) already for directed and undirected edge-periodic cycles. In addition, it was also shown in [2] that the upper bounds on the length of undirected edge-periodic cycles that guarantee them to be robber-winning from [1] are tight.

After the two conference versions of this combined journal version appeared, De Carufel et al. studied the game of cops and robbers on periodic temporal graphs where all edge labels have the same length. Note that in their setting, the global period is always linear in the size of the input, while in our setting the global period can be exponential. De Carufel et al. gave a characterization for cop-winning periodic temporal graphs in their setting in the case of a single cop [25]. They also gave an algorithm that decides if a periodic temporal graph with global period p is cop-winning for a single cop with running time $O(pn^2+nm')$ where $m' = \sum_{0\le t<p}^{|E_i|}$ is the total number of edge appearances in the first p snapshots. We remark that our algorithm of Theorem 5 can be shown to have the same time bound, as we discuss briefly after the proof of that theorem. In [26] the same authors relate the cop-number of the periodic temporal graph with the cop-number of the underlying graph and of the individual snapshot graphs.

1.1. Our contribution

In this work, we introduce the periodic cop and robber game (Section 2). Then we show that deciding whether a given edge-periodic graph is cop-winning is NP-hard even for very simple classes of edge-periodic graphs, namely for directed and undirected cycles. Moreover, we show that the problem is W[1]-hard when parameterized by the size of G for these restricted instances (Section 3). Then, we present an algorithm with time and space bound $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$ for deciding whether an edge-periodic graph is cop-winning, where n is the number of vertices of the graph and $\mathrm{lcm}(L_{\mathcal{G}})$ is the least common multiple of the edge periods. Furthermore, we show that the problem is contained in PSPACE (Section 4).

Next, we study the question of how long a directed or undirected edge-periodic cycle must be to guarantee that it is robber-winning. We first show an auxiliary result for infinite directed edge-periodic paths and then obtain tight bounds on the minimum length that guarantees a cycle to be robber-winning, for both directed and undirected edge-periodic cycles (Section 5). Let $L_{\mathcal{G}}$ be the set of edge periods. Let $\ell = 1$ if $\mathrm{lcm}(L_{\mathcal{G}})$ is at least two times the longest edge period and $\ell = 2$, otherwise. Then the minimum length that guarantees a directed edge-periodic cycle to be robber-winning is shown to be $\mathrm{lcm}(L_{\mathcal{G}}) + \ell$. For the undirected case, we show the minimum length that guarantees an edge-periodic cycle to be robber-winning to be $2 \cdot \ell \cdot \mathrm{lcm}(L_{\mathcal{G}})$.

We conclude with a discussion on open questions regarding the precise complexity of the problem of deciding whether an edge-periodic graph is cop-winning. In particular, we discuss why, at least for the special case of directed edge-periodic cycles, standard complexity classes such as NP and PSPACE might not be suitable for precisely characterizing the complexity of the problem (Section 6).

This journal paper presents a unified, full version of the results announced in the extended abstracts [1, 2] together with new results (Section 5.2) on the length of directed edge-periodic cycles that guarantees them to be robber-winning.

1.2. Further related work

In [27, 28, 29], the authors develop reductions from the standard game of cops and robbers to a *directed game graph* and specify algorithms that can decide, for a given graph, whether cop or robber wins. In [30], Kehagias and Konstantinidis note a connection between the formulations of [27, 28, 29] and reachability games. Reachability games are a well-studied class of 2-player *token-pushing* games, in which two players push a token along the edges of a directed graph in turn – one with the aim to push the token to some vertex belonging to a prespecified subset of the graph's vertex set, and the other with the aim to ensure the token never reaches such a vertex [31]. It is well known that the winner of a reachability game played on a given directed graph G can be established in polynomial time [32, 31]. For more information regarding cops and robbers/pursuit-evasion games, as well as their connection to reachability games, we refer the reader to [31, 32, 30, 33, 8, 34, 3].

2. Preliminaries

Throughout the paper, for a set A of integers, we denote by $\mathrm{lcm}(A)$ the least common multiple of the integers in A. For a string $w = w_0 w_1 \dots w_n$ with $w_i \in \{0, 1\}$, for $0 \le i \le n$, we denote by $w[i]$ the symbol w_i at position i in w. We write the concatenation of strings u and v as $u \cdot v$. For non-negative integers $i \le j$ we denote by $[i, j]$ the interval of natural numbers n with $i \le n \le j$.

An *edge-periodic (temporal) graph* $\mathcal{G} = (V, E, \tau)$ consists of a graph $G = (V, E)$ (called the *underlying graph*) and a function $\tau : E \to \{0,1\}^*$ where τ maps each edge e to a label $\tau(e) \in \{0,1\}^*$ such that e exists in a time step $t \geq 0$ if and only if $\tau(e)[t]^\circ = 1$, where $\tau(e)[t]^\circ := \tau(e)[t \bmod |\tau(e)|]$. For an edge e and non-negative integers $i \leq j$ we inductively define $\tau(e)[[i,j]]^\circ = \tau(e)[i]^\circ \cdot \tau(e)[[i+1,j]]^\circ$ and $\tau(e)[[j,j]]^\circ = \tau(e)[j]^\circ$. If $\tau(e) = 1$, we call e a 1-*edge*. We assume that every edge e exists in at least one time step, that is, for each edge e there is some $t_e \in [0, |\tau(e)|-1]$ with $\tau(e)[t_e] = 1$. We might abbreviate i repetitions of the same symbol σ in $\tau(e)$ as σ^i. We denote by $L_{\mathcal{G}} = \{|\tau(e)| \mid e \in E\}$ the set of all *edge periods* of some edge-periodic graph $\mathcal{G} = (V, E, \tau)$ and by $\mathrm{lcm}(L_{\mathcal{G}})$ the least common multiple of all periods in $L_{\mathcal{G}}$. We call an edge-periodic graph $\mathcal{G}$ with an underlying graph consisting of a single cycle an *edge-periodic cycle*. We denote by $\mathcal{G}(t)$ the sub-graph of G present in time step t. We do not make any connectivity assumptions on $\mathcal{G}$ or its snapshot graphs, but we note that $\mathcal{G}$ is trivially robber-winning if it has more temporally connected components than there are cops; note that the temporal reachability relation in periodic temporal graphs is symmetric and transitive, and hence the vertex set can be partitioned into maximal components that are temporally connected. We will discuss directed and undirected edge-periodic graphs. If not stated otherwise, we assume an edge-periodic graph to be undirected. We illustrate the notion of edge-periodic cycles in Figure 1, which shows an edge-periodic cycle $\mathcal{G}$ together with $\mathcal{G}(t)$ for the first five time steps.

2.1. The periodic cop and robber game

We now define the variant of the cops and robbers game with a single cop on edge-periodic graphs. Here, first the cop chooses her start vertex in $\mathcal{G}(0)$, then the robber chooses his start vertex in $\mathcal{G}(0)$. Then, in each time step $t \geq 0$, the cop and robber move to an adjacent vertex over an edge which is present in $\mathcal{G}(t)$ or pass and stay on their vertex. In each time step, the cop moves first, followed by the robber. We say that the cop *catches* the robber if there is some time step in which the cop and the robber are on the same vertex after the cop has moved or after the robber has moved. If the cop has a strategy to catch the robber regardless of which start vertex the robber chooses, we say that $\mathcal{G}$ is *cop-winning* and call the strategy implemented by the cop a *cop-winning strategy*. If for all cop start vertices there exists a start vertex and strategy for the robber to elude the cop indefinitely, we call

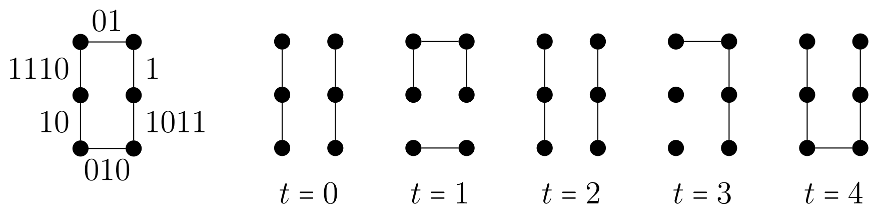


Figure 1: Edge-periodic cycle $\mathcal{G}$ (left) together with snapshots $\mathcal{G}(t)$ for $0 \le t \le 4$.

$\mathcal{G}$ *robber-winning*. The described game is a zero-sum game, i.e., $\mathcal{G}$ is either cop-winning or robber-winning.

We are interested in the computational complexity of the following problem:

> Periodic Cop & Robber
> **Input:** An edge-periodic graph $\mathcal{G} = (V, E, \tau)$.
> **Question:** Is $\mathcal{G}$ cop-winning?

A generalization with $k \ge 2$ cops instead of a single cop can be defined analogously. Initially, each of the k cops chooses her start vertex, where it is allowed that several cops choose the same start vertex. In each time step, first each of the k cops makes her move (i.e., moves to an adjacent vertex or remains where she is), then the robber. The cops catch the robber if there is some time step in which at least one of the cops is located on the same vertex as the robber after the cops have moved or after the robber has moved. For this generalization, the notions of being *k-cop-winning* and *robber-winning against k cops* are defined analogously.

3. It's hard to run around a table

In this section, we show that the NP-hardness of Periodic Cop & Robber already holds if the input graphs are very restricted. More precisely, we show that Periodic Cop & Robber is NP-hard and W[1]-hard when parameterized by the size of G, even for directed and undirected edge-periodic cycles $\mathcal{G}$.

Theorem 1. Periodic Cop & Robber *on directed or undirected edge-periodic cycles is* NP-*hard, and* W[1]-*hard parameterized by the size of the underlying graph G.*

Both the undirected and directed case of Theorem 1 are shown by a reduction from the Periodic Character Alignment problem, which was shown to be both NP-hard and W[1]-hard when parameterized by $|X|$ in [24].

> Periodic Character Alignment
> **Input:** A finite set $X \subseteq \{0,1\}^*$ of binary strings.
> **Question:** Is there a position i such that $x[i]^\circ = 1$ for all $x \in X$, where $x[i]^\circ := x[i \bmod |x|]$?

We begin with considering the case of undirected edge-periodic cycles and then proceed by adapting the obtained construction for directed edge-periodic cycles.

Lemma 2. Periodic Cop & Robber *on undirected edge-periodic cycles is* NP*-hard, and* W[1]*-hard parameterized by the size of the underlying graph* G.

Proof. We first sketch the idea of the construction. It is helpful to consider Figure 2 in the following. We represent each string in X by an edge label. The constructed cycle will consist of two chains connected by two special edges. In the first chain, the elements in X are used in some fixed order to determine an individual edge label each. In the second chain, the same edge labels are used in reverse order. This will allow the cop and the robber to occupy antipodal vertices with the same edge labels on incident edges. Hence, while the cop is on one chain and the robber on the other chain, the robber can mimic the movements of the cop. The two chains are connected by two special edges for which the edge labels are complementary in one position of the labels and identical in all other positions. This will allow the cop to switch between the chains in a certain time step while the robber is trapped on his chain. In this situation, the cop will be able to catch the robber if and only if there is a position i, such that $x[i]^\circ = 1$ for all $x \in X$, in which case all edges of the chains will be present for some period.

We now proceed with the formal proof. Let X be an instance of Periodic Character Alignment. We describe how to construct in polynomial time an instance $\mathcal{G} = (V, E, \tau)$ of Periodic Cop & Robber, where $\mathcal{G}$ is an undirected edge-periodic cycle, such that X is a yes-instance of Periodic Character Alignment if and only if $\mathcal{G}$ is a yes-instance of Periodic Cop & Robber.

Let $|X| = m$ and $\{x_1, \ldots, x_m\}$ be the elements of X. We set $V := \{\ell_j, r_j \mid 0 \le j \le m\}$ and $E := \{\{\ell_{j-1}, \ell_j\}, \{r_{j-1}, r_j\} \mid 1 \le j \le m\} \cup \{\{\ell_0, r_m\}, \{\ell_m, r_0\}\}$.

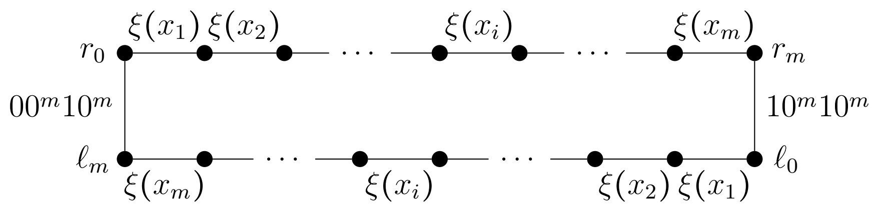


Figure 2: Periodic Cop & Robber instance constructed from a Periodic Character Alignment instance with set of strings $X = \{x_1, \ldots, x_m\}$ in the proof of Lemma 2. For $x_j \in X$ the edge labels are defined as $\xi(x_j) \coloneqq \xi(x_j[0]) \cdot \xi(x_j[1]) \cdot \ldots \cdot \xi(x_j[|x_j|-1])$, with $\xi(c) \coloneqq 0c^m01^m$ for $c \in \{0,1\}$. The upper chain corresponds to the vertices r_j and the lower chain to the vertices ℓ_j.

Next, we set $\tau(\{\ell_0, r_m\}) \coloneqq 10^m10^m$ and $\tau(\{\ell_m, r_0\}) \coloneqq 00^m10^m$. Let $\xi(c) \coloneqq 0c^m01^m$ for all $c \in \{0,1\}$. Finally, we set $\tau(\{\ell_{j-1}, \ell_j\}) \coloneqq \tau(\{r_{j-1}, r_j\}) \coloneqq \xi(x_j[0]) \cdot \xi(x_j[1]) \cdot \ldots \cdot \xi(x_j[|x_j|-1])$ for each $x_j \in X$. Note that the length of each edge label is divisible by $2m+2$. For $i \geq 0$, let $T_i \coloneqq [(2m+2)\cdot i, (2m+2)\cdot(i+1)-1]$ denote the *i-th time block*, that is, the $2m+2$ consecutive time steps starting from $(2m+2)\cdot i$. Note that the j-th edge label limited to the i-th time block $\tau(\{\ell_{j-1}, \ell_j\})[T_i]^\circ = \tau(\{r_{j-1}, r_j\})[T_i]^\circ$ is exactly $\xi(x_j[i]^\circ)$.

Next, we show that X is a yes-instance of Periodic Character Alignment if and only if $\mathcal{G}$ is a yes-instance of Periodic Cop & Robber.

($\Rightarrow$) Let i be a position such that $x[i]^\circ = 1$ for all $x \in X$. We describe the winning strategy for the cop. She should choose the vertex ℓ_0 as her start vertex and should never move until the beginning t of the i-th time block T_i. Since $x[i]^\circ = 1$ for all $x \in X$, $\tau(\{\ell_{j-1}, \ell_j\})[T_i]^\circ = \tau(\{r_{j-1}, r_j\})[T_i]^\circ = \xi(1) = 01^m01^m$. Consequently, in time step t only the edge $\{\ell_0, r_m\}$ exists and in the following m time steps, all edges except $\{\ell_0, r_m\}$ and $\{\ell_m, r_0\}$ exist.

If the robber is currently on some vertex r_j, then the cop should move to r_m in time step t. Otherwise, the cop should stay on ℓ_0 in this time step. By the fact that the edge $\{\ell_m, r_0\}$ does not exist in time step t, we obtain that, at the beginning of time step $t+1$, both players are either on vertices labeled with r or labeled with ℓ. Since all edges of the two paths $(\ell_0, \ldots, \ell_m)$ and $(r_0, \ldots, r_m)$ exist in the time steps $[t+1, t+m]$, the cop can catch the robber in at most m time steps, since neither $\{\ell_0, r_m\}$ nor $\{\ell_m, r_0\}$ exists in any of the time steps $[t+1, t+m]$. Consequently, $\mathcal{G}$ is a yes-instance of Periodic Cop & Robber.

($\Leftarrow$) Suppose that X is a no-instance of Periodic Character Alignment. We describe a winning strategy for the robber. In the following, we say that the vertex ℓ_j is the *mirror vertex* of r_j and vice versa. Moreover, we say that the robber *mirrors the move* of the cop at some time step t, if the cop is on the mirror vertex of the robber at the beginning of time step t and the robber moves to the mirror vertex of the vertex the cop ends on in time step t.

The start vertex of the robber should be the mirror vertex of the start vertex of the cop. If it is possible, then the robber should always mirror the moves of the cop.

Note that the only move the robber *cannot* mirror is if the cop traverses the edge $\{\ell_0, r_m\}$ at the beginning of some i-th time block.

We show that the robber has a strategy to end on the mirror vertex during the i-th time block and evade the cop until then.

Due to symmetry assume that the cop moves from r_m to ℓ_0 and, thus, the robber is currently on ℓ_m. Since X is a no-instance of Periodic Character Alignment, there is at least one $x_j \in X$ with $x_j[i]^\circ = 0$. Hence, $\tau(\{\ell_{j-1}, \ell_j\})[T_i]^\circ = \tau(\{r_{j-1}, r_j\})[T_i]^\circ = \xi(0) = 00^m01^m$. Consequently, the cop cannot catch the robber at ℓ_m in the first $m+1$ time steps of the i-th time block. Hence, the robber should stay on this vertex until the beginning of time step $(2m+2)\cdot i + m + 1$.

If the cop moves from ℓ_0 to r_m in time step $(2m+2)\cdot i+m+1$, the robber is again on the mirror vertex of the cop and is able to mirror all of the cop's moves in the remaining time steps of this time block. Otherwise, the cop is on some vertex ℓ_p at the end of time step $(2m+2)\cdot i + m + 1$. In this case, the robber should move to r_0 in that time step. Since the edges $\{\ell_0, r_m\}$ and $\{\ell_m, r_0\}$ do not exist in the remaining time steps of this time block, the cop cannot catch the robber in this time block. Moreover, since all edges of the path $(r_0, \dots, r_m)$ exist in all of the last m time steps of the i-th time block, the robber can move along the path $(r_0, \dots, r_m)$ and reach the mirror vertex of the cop in at most m time steps. Consequently, we can show via induction that the robber has an infinite evasive strategy and, thus, $\mathcal{G}$ is a no-instance of Periodic Cop & Robber. Since Periodic Character Alignment is W[1]-hard when parameterized by $|X|$ and $|V| = |E| = 2\cdot|X|+2$, we obtain that Periodic Cop & Robber is W[1]-hard when parameterized by the size of the underlying graph of $\mathcal{G}$ even on undirected edge-periodic cycles. □

Next, we adapt the previous construction for *directed* edge-periodic cycles. It is helpful to consider Figure 3 in the following. In the adaption, we only have one chain with edge labels determined by the elements of X. The end vertex of this chain is connected to a new vertex s which is again connected to the start vertex of the chain. The edges incident with s will act as the two edges connecting the two chains in the previous construction by delaying the robber, such that the cop can catch him if all edges corresponding to X are present in some time period.

Lemma 3. Periodic Cop & Robber *on* directed *edge-periodic cycles is* NP*-hard, and* W[1]*-hard parameterized by the size of the underlying graph.*

Proof. Again, we reduce from Periodic Character Alignment. Let X be an instance of Periodic Character Alignment. We describe how to construct an instance $\mathcal{G} = (V, E, \tau)$ of Periodic Cop & Robber, where $\mathcal{G}$ is a directed edge-periodic cycle. Let $|X| = m$ and $\{x_1, \dots, x_m\}$ be the elements of X. We set $V := \{v_j \mid 0 \le j \le m\} \cup \{s\}$ and $E := \{(v_{j-1}, v_j) \mid 1 \le j \le m\} \cup \{(v_m, s), (s, v_0)\}$. Next, we set $\tau((v_m, s)) := 0^m10^m0$ and $\tau((s, v_0)) := 0^m00^m1$. Let $\xi(c) := c^m01^m1$ for all $c \in \{0, 1\}$. Finally, we set $\tau((v_{j-1}, v_j)) := \xi(x_j[0]) \cdot \xi(x_j[1]) \cdot \ldots \cdot \xi(x_j[|x_j| - 1])$ for each $x_j \in X$.

Note that the length of each edge label is divisible by $2m + 2$. For $t \ge 0$, let $T_t := [(2m + 2) \cdot t, (2m + 2) \cdot (t + 1) - 1]$ denote the *t-th time block*, that is, the $2m + 2$ consecutive time steps starting from $(2m + 2) \cdot t$. Note that the j-th edge label limited to the t-th time block $\tau((v_{j-1}, v_j\})[T_t]^\circ$ is exactly $\xi(x_j[t]^\circ)$. Next, we show that X is a yes-instance of Periodic Character Alignment if and only if $\mathcal{G}$ is a yes-instance of Periodic Cop & Robber.

($\Rightarrow$) Let t be a position such that $x[t]^\circ = 1$ for all $x \in X$. We describe the winning strategy for the cop. The cop should choose the vertex v_0 as her start vertex and should never move until the beginning $t^* := (2m + 2) \cdot t$ of the t-th time block. By construction and the fact that $x_i[t]^\circ = 1$ for each $x_i \in X$, $\tau((v_{i-1}, v_i))[T_t]^\circ = \xi(1) = 1^m01^m1$. Hence, the cop can move from vertex v_i to vertex v_{i+1} in time step $t^* + i$ for each $i \in [0, m-1]$ and, thus, reach the vertex v_m in time step $t^* + m - 1$. Moreover, the cop can then move to the vertex s in time step $t^* + m$. By construction, $\tau((s, v_0))[t^* + j]^\circ = 0$ for each $j \in [0, m]$. Hence, the cop has a winning strategy since she started at vertex v_0 and moved over every vertex of V while the robber was not able to traverse the edge (s, v_0).

($\Leftarrow$) Suppose that for every position t, there is some $x_j \in X$ with $x_j[t]^\circ = 0$. We show that the robber has a winning strategy. For some time step, let w_C and w_R denote the vertex of the cop and robber, respectively, in this time step. We call the vertex v_0 *safe* for all vertices of $V \setminus \{v_0, s\}$, we call v_m *safe*

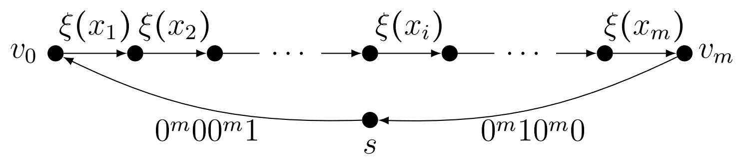


Figure 3: Periodic Cop & Robber instance constructed from a Periodic Character Alignment instance with set of strings $X = \{x_1, \dots, x_m\}$ in the proof of Lemma 3. For $x_j \in X$ the edge labels are defined by $\xi(x_j) := \xi(x_j[0]) \cdot \xi(x_j[1]) \cdot \dots \cdot \xi(x_j[|x_j| - 1])$, with $\xi(c) := c^m 0 1^m 1$ for $c \in \{0, 1\}$.

for v_0 and s, and we call s *safe* for v_0. Let u_C be the start vertex of the cop, then the robber should choose a vertex which is safe for u_C as his start vertex.

Claim 4. *Let $t^* = t \cdot (2m+2)$ be the beginning of the t-th time block for some $t \geq 0$, let u_C be the vertex of the cop at time step t^* and u_R be the vertex of the robber at time step t^*. If u_R is safe for u_C, then the robber has a strategy such that the cop cannot catch him in the t-th time block and the robber ends on a vertex that is safe for the vertex of the cop at the end of the t-th time block.*

Proof. **Case 1:** $u_C \in V \setminus \{s, v_0\}$ **and** $u_R = v_0$**.** The robber should wait on vertex v_0 until the beginning of time step $t^* + m$. Since the edge (s, v_0) only exists in the last time step of the t-th time block, the cop cannot catch the robber in any of these time steps. If the cop does not traverse the edge (v_m, s) in time step $t^* + m$, then the robber should stay on vertex v_0 until the beginning of the next time block. Since the edge (v_m, s) only exists in time steps t' with $t' \bmod (2m + 2) = m$, it follows that the cop ends on some vertex of $V \setminus \{s, v_0\}$ at the end of the t-th time block. Thus, at the beginning of the $(t+1)$-th time block, the vertex of the robber is safe for the vertex of the cop.

Otherwise, the cop traverses the edge (v_m, s) in time step $t^* + m$. Then, the robber should traverse the edge (v_{i-1}, v_i) in time step $t^* + m + i$ for each $i \in [1, m]$, while the cop has to wait on s. Hence, the robber reaches v_m in time step $t^* + (2m + 2) - 2$. In time step $t^* + (2m + 2) - 1$, the cop can either stay on s or move to v_0. In both cases the robber should stay on v_m which is safe for both s and v_0.

Case 2: $u_C = s$ **and** $u_R = v_m$**.** Since the edge (s, v_0) only exists in the last time step of the t-th time block, the cop has to stay on s until the beginning of time step $t^* + (2m+2) - 1$. In time step $t^* + (2m+2) - 1$, the cop can either stay on s or move to v_0. In both cases the robber stays on v_m which is safe for both s and v_0.

Case 3: $u_C = v_0$ **and** $u_R \in \{v_m, s\}$**.** Let $j \in [1, m]$ such that $x_j[t]^\circ = 0$, and recall that by definition of τ it follows that $\tau((v_{j-1}, v_j))[T_t]^\circ = 0^m 0 1^m 1$. Thus, the cop cannot reach the vertex v_m in the first $m + 1$ time steps of the t-th time block. In time step $t^* + m$, the robber should stay on s if s is his current vertex or traverse the edge (v_m, s), otherwise. Since this is the only time step in which this edge exists in the t-th time block, the cop cannot catch the robber in this time block. Until the beginning of time step $t^* + (2m + 2) - 1$, the robber should stay on s. If the cop ends her turn on vertex v_0, then the robber should stay on s. Otherwise, the robber should traverse the edge (s, v_0) in time step $t^* + (2m + 2) - 1$. In both cases, the vertex of the robber is safe for the vertex of the cop at the beginning of the $(t + 1)$-th time block. $\lhd$

By using Claim 4, one can show via induction that the robber has an infinite evasive strategy and, thus, $\mathcal{G}$ is a no-instance of PERIODIC COP & ROBBER. $\square$

4. Complexity upper bounds

In this section, we show that PERIODIC COP & ROBBER can be solved for *general* edge-periodic graphs by translating the periodic cop and robber game into a reachability game. First, we show that constructing the reachability game explicitly yields an algorithm with time and space bound $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$ for one cop or $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot k \cdot n^{k+2})$ for k cops, where n is the number of vertices of $\mathcal{G}$. Within the same time and space bounds, one can also determine a winning strategy for the winning player. As $\mathrm{lcm}(\mathcal{G})$ can be exponential in the size of the representation of the given periodic cop and robber game, these algorithms may require exponential time and space.

Our approach is essentially to convert the given edge-periodic graph into a temporal graph with global period $\mathrm{lcm}(L_{\mathcal{G}})$ and then translate the problem of deciding if that graph is cop-winning into a reachability game. Applying the more recent algorithm by De Carufel et al. [25] with running-time $O(pn^2 + nm')$ to that temporal graph with global period $p = \mathrm{lcm}(L_{\mathcal{G}})$ would yield the same worst-case running-time bound of $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$ for one cop, as the total number m' of edge appearances in the first p snapshots can be as large as $\Theta(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^2)$.

Afterwards, we show that the Periodic Cop & Robber problem can be solved in polynomial space, both for the case of one cop and the case of k cops provided that k is bounded by a constant. Note that two-player games may take exponentially many turns, and hence containment in PSPACE is not obvious. In our case, already the period of graphs on which the game is played is exponential in general. This prohibits a standard incremental PSPACE algorithm approach. We show that, despite the potentially exponential period of the sequence of graphs $\mathcal{G}(t)$, we can determine whether the cop has a winning strategy by sweeping through the configuration space in such a way that we need to consider only polynomially many vertices at any time. The fact that we only consider one cop (or a number of cops bounded by a fixed constant) and one robber here is crucial for the polynomial bound.

4.1. Solving Periodic Cop & Robber *via reachability games*

In the following, we establish this theorem:

Theorem 5. *Let $\mathcal{G}$ be an edge-periodic graph of order n, and let $L_{\mathcal{G}} = \{|\tau(e)| : e \in E(\mathcal{G})\}$. Then,* Periodic Cop & Robber *can be decided in $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$ time.*

The proof relies primarily on a transformation from a given edge-periodic graph $\mathcal{G}$ to a finite *directed game graph G'*. The transformation is such that playing an instance of the periodic cop and robber game on $\mathcal{G}$ is equivalent to playing a 2-player token-pushing game (specifically, a *reachability game*) on G'. To establish this equivalence, we need a way of translating a particular state of the cop and robber game played on $\mathcal{G}$ to a corresponding state in the reachability game played on G'. To this end, the following definition properly introduces the notion of a *configuration* in a cop and robber game played on an edge-periodic graph $\mathcal{G}$.

Definition 6 (Configuration in $\mathcal{G}$)**.** *The current state of a cop and robber game played on an edge-periodic graph $\mathcal{G}$ is determined by four individual pieces of information: (1) the vertex currently occupied by the cop; (2) the vertex currently occupied by the robber; (3) the player whose turn it is to move; and (4) the current time step t. We define a configuration in $\mathcal{G}$ to be a 4-tuple, (u_c, u_r, s, t), where $u_c \in V(\mathcal{G})$ is the cop's current vertex, $u_r \in V(\mathcal{G})$ is the robber's current vertex, $s \in \{c, r\}$ is the player whose turn it is to move next (where c stands for the cop and r for the robber), and t is the current time step.*

We call any configuration (u_c, u_r, s, t) such that $u_c = u_r$ a *terminating configuration*, since this indicates that both players are situated on the same vertex and hence the cop has won. We now formally introduce the notion of *reachability games* [31].

Definition 7 (Reachability game G')**.** *A reachability game is a directed graph G', given as a 3-tuple:*

$$G' = (V_0 \cup V_1, E', F),$$

where $V_0 \cup V_1$ is a partition of the state set V'; $E' \subseteq V' \times V'$ is a set of directed edges; and $F \subseteq V'$ is a set of final states.

The game is played by two opposing players, Player 0 and Player 1; V_0 and V_1 are the (disjoint) sets of nodes owned by Player 0 and Player 1, respectively. One can imagine a token being placed at some initial vertex (call it v_0) at the start of the game. Depending on whether $v_0 \in V_0$ or $v_0 \in V_1$, we can then imagine the corresponding player selecting one of the outgoing edges of v_0, and pushing the token along that edge. When the token arrives at the next vertex, the corresponding player then selects an outgoing edge and pushes the token along it. This process then continues – such a sequence of moves constitutes a *play* of the reachability game on G'. Formally, a play $\phi = v_0, v_1, \ldots$ is a (possibly infinite) sequence of vertices in V', such that $(v_i, v_{i+1}) \in E'$ for all $i \geq 0$. We say that a play ϕ is *won* by Player 0 if there exists some i such that $v_i \in F$. Otherwise, ϕ is of infinite length and for no i is $v_i \in F$; in this latter case, we say that ϕ is won by Player 1.

Reachability games of this type are also sometimes called *turn-based reachability games*, as opposed to concurrent reachability games [35]. In the case where F contains only a single vertex, the problem of determining whether Player 0 has a winning strategy from a given start vertex v_0 in a turn-based reachability game is known to be equivalent to the AND-OR GRAPH REACHABILITY problem, which is PTIME-complete [36]. In our transformation of a cop and robber game into a reachability game, the size of the resulting directed graph for the reachability game may be exponential in the size of the edge-periodic graph of the cop and robber game.

4.1.1. Transformation

We now detail our transformation from a given edge-periodic graph $\mathcal{G}$ to a reachability game $\beta(\mathcal{G})$: let β be a transformation function that takes as argument a given edge-periodic graph $\mathcal{G}$, so that the notation $\beta(\mathcal{G})$ denotes the game graph G' on which each play of a reachability game corresponds to a sequence of moves performed by the cop and the robber in a game of cop and robber on $\mathcal{G}$. Further, let $L_{\mathcal{G}} = \{|\tau(e)| : e \in E(\mathcal{G})\}$. We let $\beta(\mathcal{G}) := G' = (V', E', F)$, and go on to define its individual components below:

State set V'. We define the state set (i.e., vertex set) of our directed game graph $\beta(\mathcal{G})$ to be a set of 4-tuples, each corresponding to a configuration in the game of cop and robber on $\mathcal{G}$, as follows:

$$V' = \{(u_c, u_r, s, t) : u_c, u_r \in V(\mathcal{G}), s \in \{c, r\}, \text{ and } t \in [0, \mathrm{lcm}(L_{\mathcal{G}}) - 1]\}.$$

Keeping in line with Definition 7, we also let $V_0 := \{(u_c, u_r, s, t) \in V' : s = c\}$ and $V_1 := \{(u_c, u_r, s, t) \in V' : s = r\}$ be the sets of $\mathsf{Player\ 0}$ (or cop) owned nodes, and $\mathsf{Player\ 1}$ (or robber) owned nodes, respectively. We wish to capture with the finite directed game graph G' all possible configurations P of the cop and robber game on $\mathcal{G}$. Since the lifetime of $\mathcal{G}$ is infinite, we cannot simply create a state $S \in V(G')$ corresponding to each possible configuration P, however; the infinite number of time steps would result in an infinite game graph G'. It is not hard to see that in time step $\mathrm{lcm}(L_{\mathcal{G}}) - 1$ the presence of each edge is determined by the final bit of its label (since $|\tau(e)|$ divides $\mathrm{lcm}(L_{\mathcal{G}})$ for all $e \in E(\mathcal{G})$). In the next time step, all labels will restart, i.e., the presence of each edge is determined by the first bit of its label. As such, we can view the temporal structure of our edge set as an infinitely repeating pattern, and by letting t range over the integers in $[0, \mathrm{lcm}(L_{\mathcal{G}}) - 1]$ we are able to properly capture this structure using only a finite number of states.

Edge set E'. In order to construct the edge set $E' \subseteq (V_0 \times V_1) \cup (V_1 \times V_0)$, we consider all pairs of states $S = (u_c, u_r, s, t)$ and $S' = (u'_c, u'_r, s', t')$ such that $S \neq S'$ and $S, S' \in V'$. We then let E' be the set of edges such that $(S, S') \in E'$ if and only if the states S and S' satisfy all of the conditions below:

(1) $\big(s = c \wedge s' = r\big) \vee \big(s = r \wedge s' = c\big)$,

(2) $s = c \implies \big(u_c = u'_c \vee \{u_c, u'_c\} \in E(\mathcal{G})\big) \wedge (u_r = u'_r) \wedge (t' = t)$,

(3) $s = r \implies \big(u_r = u'_r \vee \{u_r, u'_r\} \in E(\mathcal{G})\big) \wedge \big(u_c = u'_c\big)$
$\wedge \big(t' \in [0, \mathrm{lcm}(L_{\mathcal{G}}) - 1] \text{ satisfies } t' = (t+1) \bmod \mathrm{lcm}(L_{\mathcal{G}})\big)$,

(4) $s = c \wedge u_c \neq u'_c \implies \tau(\{u_c, u'_c\})[t]^{\circ} = 1$,

(5) $s = r \wedge u_r \neq u'_r \implies \tau(\{u_r, u'_r\})[t]^{\circ} = 1$.

Condition (1) ensures that the moves in any sequence of moves constituting a play in G' alternate between cop and robber. Condition (2) ensures that any state S', reachable in one move from a cop-owned state S, is such that u'_c is adjacent to u_c in $\mathcal{G}$ (or is in fact u_c, indicating that the cop has waited at the current vertex); that the robber's vertex u_r does not change; and, that $t' = t$, satisfying the rule stating that the cop moves first in any given time step, followed by the robber who must also make a move in time step t. On the other hand, Condition (3) ensures that, once the robber pushes the token from some robber-owned state S, his new vertex u'_r is adjacent to u_r in $\mathcal{G}$ (or equal to u_r); that the cop's vertex remains the same, and that the state S' to which the token is pushed is a state in which the current time step is advanced by one. Conditions (4) and (5) ensure that both players can only make moves across edges that are incident to their current vertex if they are present in the current time step; on the other hand, they also ensure that either player always has the ability to remain at their current vertex in any time step t if they should choose to do so.

Set of final states F. Let $F = \{(u_c, u_r, s, t) \in V' : u_c = u_r\}$, so that the set of final states consists of all those states that correspond to a configuration in $\mathcal{G}$ such that the cop is positioned on the same vertex as the robber. This models the fact that the game terminates only when this condition is met by the current configuration.

4.1.2. Proof of Theorem 5

We first introduce the elements of the theory of reachability games that are required for the proof of Theorem 5, starting with the definition of the *attractor set*:

Definition 8 (Attractor set $\mathrm{Attr}(F)$ [32]). *The sequence $(\mathrm{Attr}_i(F))_{i\geq 0}$ is recursively defined as follows:*

$$\begin{aligned}\mathrm{Attr}_0(F) &= F\\ \mathrm{Attr}_{i+1}(F) &= \mathrm{Attr}_i(F) \cup \{v \in V_0 \mid \exists (v,u) \in E' : u \in \mathrm{Attr}_i(F)\} \cup\\ &\quad\ \{v \in V_1 \mid \forall (v,u) \in E' : u \in \mathrm{Attr}_i(F)\}\end{aligned}$$

We can see that the sets $\mathrm{Attr}_i(F)$, as defined above, are a sequence of subsets of V' that are monotone with respect to set-inclusion. We then let

$$\mathrm{Attr}(F) = \bigcup_{i\geq 0} \mathrm{Attr}_i(F).$$

Since G' is finite, we are able to view the set $\mathrm{Attr}(F)$ *as the least fixed point of the sequence* $(\mathrm{Attr}_i(F))_{i\geq 0}$.

From Definition 8 it follows by induction that, from those states $S \in \mathrm{Attr}_i(F) \cap V_0$ such that $i \geq 1$ and $S \notin \mathrm{Attr}_j(F)$ for any $j < i$, Player 0 is able to force the sequence of play into some state $S_F \in F$ within i moves, by selecting for each such S a successor state S' such that $(S, S') \in E'$ and $S' \in \mathrm{Attr}_{i-c}$ for some $c \geq 1$. On the other hand we have that, from any state $S \in \mathrm{Attr}_i(F) \cap V_1$ (again, let $i \geq 1$ and $S \notin \mathrm{Attr}_j(F)$ for any $j < i$), Player 1 cannot avoid forcing the sequence of play into a state $S' \in \mathrm{Attr}_{i-c}$ (for some $c \geq 1$); from the definition of Attr_i ($i \geq 0$), it again follows by induction that the play will be forced into some state $S_F \in F$ in at most i time steps. This brings us to the following well-known result from the reachability games literature, which will be useful in proving Theorem 5:

Theorem 9 (Berwanger [32])**.** *In a given reachability game $G' = (V', E', F)$, Player 0 has a winning strategy from any state $S \in \mathrm{Attr}(F)$, and Player 1 has a winning strategy from any state $S \in (V_0 \cup V_1) \setminus \mathrm{Attr}(F)$.*

Recall now that the transformation β produces, from a given edge-periodic graph $\mathcal{G}$, a directed game graph $\beta(\mathcal{G}) = (V', E', F)$ such that there is a correspondence between every possible configuration in the game of cop and robber on $\mathcal{G}$ with some state in V', and vice versa. Using the notation S_P to refer to the state in V' that corresponds to the configuration P in the game of cop and robber on $\mathcal{G}$, we can compute the set $\mathrm{Attr}(F)$ for the game graph $\beta(\mathcal{G})$ and thus, on invocation of Theorem 9, state the following lemma:

Lemma 10. *The cop can force a win from a configuration P if and only if the state $S_P \in V(\beta(\mathcal{G}))$ satisfies $S_P \in \mathrm{Attr}(F)$.*

Note that one consequence of Lemma 10 is the following: In a game of cop and robber on $\mathcal{G}$ starting from a configuration P such that $S_P \notin \mathrm{Attr}(F)$, the robber can force the sequence of moves to never reach any state $S \in F$, and, as such, the game can be won by the robber.

Lemma 11. *An edge-periodic graph $\mathcal{G}$ is cop-winning if and only if there exists a vertex $v \in V(\mathcal{G})$ such that $(v, u, c, 0) \in \mathrm{Attr}(F)$ for all $u \in V(\mathcal{G})$.*

Proof. ($\Rightarrow$) Assume not, so that $\mathcal{G}$ is cop-winning but there exists no vertex $v \in V(\mathcal{G})$ such that $(v, u, c, 0) \in \mathrm{Attr}(F)$ for all $u \in V(\mathcal{G})$. Then, for every v, there exists at least one vertex u_v such that the state $(v, u_v, c, 0) \notin \mathrm{Attr}(F)$. Assume that the cop chooses some start vertex v. Then the robber chooses start vertex u_v. It follows that the robber can force the equivalent reachability game on $\beta(\mathcal{G})$ to begin from a state $S_{(v,u_v,c,0)} \notin \mathrm{Attr}(F)$, hence winning the reachability game regardless of the cop's choice of v. Notice that this implies that there exists a winning strategy for the robber in the game of cop and robber on $\mathcal{G}$; this is a contradiction since, by assumption, $\mathcal{G}$ is cop-winning.

($\Leftarrow$) Assume the cop chooses v as her start vertex. By doing so, the equivalent reachability game on $\beta(\mathcal{G})$ starts at a state $(v, u_r, c, 0) \in \mathrm{Attr}(F)$ regardless of the robber's choice of u_r, since $(v, u_r, c, 0) \in \mathrm{Attr}(F)$ for all $u_r \in V(\mathcal{G})$. Hence, regardless of the robber's choice of u_r, the cop wins the reachability game on $\beta(\mathcal{G})$ and, as a result, can win the game of cop and robber on $\mathcal{G}$ by picking start vertex v; the lemma follows. □

The proof of the main theorem will also make use of a further known result from the reachability games literature; for the following, let $G' = (V', E', F)$ be a given directed game graph.

Theorem 12 (Grädel et al. [31]). *There exists an algorithm that computes the set* $\mathrm{Attr}(F)$ *in time* $O(|V'| + |E'|)$.

Given the above, all is in place for the proof of Theorem 5:

Proof of Theorem 5. Since $n = |V(\mathcal{G})|$, β produces, given an edge-periodic graph $\mathcal{G}$, a directed game graph $\beta(\mathcal{G}) = (V', E', F)$ such that $|V'| \in O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^2)$. To see this, observe first that for a configuration $P = (u_c, u_r, s, t)$ in a game of cop and robber on $\mathcal{G}$, there are n ways to choose $u_c \in V(\mathcal{G})$, n ways to choose $u_r \in V(\mathcal{G})$, and a further 2 ways to choose $s \in \{c, r\}$. By definition of the transformation function, G' has states for time steps $t \in [0, \mathrm{lcm}(L_{\mathcal{G}}) - 1]$ only, and so in total we have that $|V'| = 2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) \cdot n^2 = O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^2)$, as claimed. Next, note that each state $S_P \in V'$ has at most n edges leading away from it to other states. This is because, in the corresponding configuration P in the game of cop and robber on $\mathcal{G}$, the player whose turn it currently is has at most n choices of moves across edges – at most $n-1$ edges leading to other vertices plus the choice of remaining at the current vertex. Since there are $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^2)$ states $S \in V'$, it follows that $|E'| \in O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$.

Combining the above with the result of Theorem 12, we can conclude that the attractor set $\mathrm{Attr}(F)$ (that is, the set of all states from which Player 0, i.e., the cop, has a winning strategy) of $\beta(\mathcal{G})$ can be computed in time

$O(\mathrm{lcm}(L_{\mathcal{G}})\cdot n^3)$. By Lemma 11, we can then verify whether $\mathcal{G}$ is cop-winning by checking if there exists at least one vertex $v \in V(\mathcal{G})$ such that $(v,u,c,0) \in \mathrm{Attr}(F)$ for all $u \in V(\mathcal{G})$; if such a vertex v exists, the algorithm will return YES, otherwise the algorithm will return NO. Carrying out this check can clearly take at most $O(n^2)$ time, and the theorem follows. □

We remark that the time bound of Theorem 5 can also be stated as $O(\mathrm{lcm}(L_{\mathcal{G}})\cdot n^2 + nm')$, where m' is the total number of edge appearances in the first $\mathrm{lcm}(L_{\mathcal{G}})$ snapshots of $\mathcal{G}$. This can be shown as follows. For the game graph $G' = (V', E')$, we have already bounded $|V'| = O(\mathrm{lcm}(L_{\mathcal{G}})\cdot n^2)$ in the proof of Theorem 5. To bound $|E'|$, it suffices to observe that the $2n^2$ states of the form (u,v,s,t) for fixed t have at most $2n^2 + 2n|E(\mathcal{G}(t))|$ outgoing edges in total, where $\mathcal{G}(t)$ is the snapshot of all edges that are present at time t in $\mathcal{G}$. This holds because, for each vertex $u \in V$, there are n states (u,v,c,t) and n states (v,u,r,t) in V', and each of these has an outgoing edge for each edge incident with u in $\mathcal{G}(t)$ plus one additional outgoing edge corresponding to the cop or robber deciding to stay at the current vertex u. The number of edges of the former type is $2n|E(\mathcal{G}(t))|$, and the number of edges of the latter type is $2n^2$. Adding these bounds over all $\mathrm{lcm}(L_{\mathcal{G}})$ time steps, we get $|E'| = O(\mathrm{lcm}(L_{\mathcal{G}})\cdot n^2 + nm')$. Thus, we have $|V'| + |E'| = O(\mathrm{lcm}(L_{\mathcal{G}})\cdot n^2 + nm')$, and by Theorem 12 the running-time of our algorithm satisfies the same time bound as the more recent algorithm with running-time $O(pn^2 + nm')$ for temporal graphs with global period p by De Carufel et al. [25].

We also note that, as a direct consequence of Theorem 5, as long as $\mathrm{lcm}(L_{\mathcal{G}})$ is polynomial in n and $\max L_{\mathcal{G}}$, the winner of a given graph $\mathcal{G}$ can be decided in polynomial time. Furthermore, if the label lengths $|\tau(e)|$ are bounded by some constant for all $e \in E(\mathcal{G})$, then the winner can be decided in $O(n^3)$ time.

As well as being able to decide whether a given edge-periodic graph is cop-winning, we would like to be able to compute a strategy for the winning player of the game of cop and robber on a given graph $\mathcal{G}$. One common way to view a strategy for Player i ($i \in \{0,1\}$), in a general infinite game played on a game graph $G = (V,E,F)$ (where $V := V_0 \cup V_1$), is as a partial function $\sigma : V^* \cdot V_i \to V$. Here, $V^* \cdot V_i$ can be seen as the set of all prefixes (of any play ϕ in G) that end in a state $S \in V_i$, with σ dictating to Player i the appropriate move to play, based on the history of these prefixes.

On the other hand, a *memoryless* strategy can be viewed more simply

– as a partial function $\sigma : V_i \to V'$. Such a strategy σ can be employed in games where a correct move for a player does not depend on the entire state history of some play (or a prefix of) ϕ, but only on the current state. It is well known that reachability games fall into this category [31]; since the cop and robber game reduces to a reachability game, we are thus able to make use of the following result from the literature:

Theorem 13 (Berwanger [32]). *Given a reachability game $G' = (V', E', F)$, one can compute in $O(|V'| + |E'|)$ time a memoryless winning strategy for Player 0 from any state $S \in \mathrm{Attr}(F)$, and a memoryless winning-strategy for Player 1 from any state $S \in (V_0 \cup V_1) \setminus \mathrm{Attr}(F)$.*

As such, given a directed game graph $\beta(\mathcal{G}) = (V', E', F)$ (with $V' := V_0 \cup V_1$), Theorem 13 tells us that it suffices to compute, for the winning player, a memoryless winning strategy $\sigma_i : V_i \to V'$, with the value of $i \in \{0, 1\}$ depending on the winner of the reachability game on $\beta(\mathcal{G})$. The following theorem shows that it is possible to interpret any such σ as a strategy for the winning player in the corresponding game of cop and robber on $\mathcal{G}$:

Theorem 14. *Let $\mathcal{G}$ be an arbitrary edge-periodic graph and $L_{\mathcal{G}} = \{|\tau(e)| : e \in E(\mathcal{G})\}$. Then, depending on whether $\mathcal{G}$ is cop-winning or not, one can compute in $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$ time either a memoryless winning strategy enabling the cop to capture the robber, or a memoryless winning strategy enabling the robber to evade capture indefinitely.*

Proof. Let $b \in \{\mathsf{YES}, \mathsf{NO}\}$ be the return value of the algorithm from Theorem 5 when provided $\mathcal{G}$ as input. First, we construct a strategy for the winning player of the equivalent reachability game $\beta(\mathcal{G}) = (V', E', F)$, and then we go on to show how such a strategy can be interpreted as a strategy for the corresponding game of cop and robber on $\mathcal{G}$.

First, consider the case $b = \mathsf{YES}$. Then we know $\mathcal{G}$ is cop-winning and, by Lemma 11, we know that there exists some vertex v such that $(v, u, c, 0) \in \mathrm{Attr}(F)$ for all $u \in V(\mathcal{G})$. As such, the initial stage of our strategy for the cop consists of computing such a vertex v and setting the cop start vertex in $\mathcal{G}$ to v. Now, using Theorem 13, we can compute a memoryless winning-strategy σ_c. The initial stage of identifying some vertex v such that $(v, u, c, 0) \in \mathrm{Attr}(F)$ for all $u \in V(\mathcal{G})$ takes $O(n^2)$ time, and the algorithm of Theorem 13 takes time at most $O(|V'| + |E'|) = O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^3)$; it follows that the overall construction of a cop-strategy for the reachability game

$\beta(\mathcal{G})$ takes $O(\text{lcm}(L_{\mathcal{G}}) \cdot n^3)$ time. Such a strategy for $\beta(\mathcal{G})$ can then be interpreted as strategy for the cop in the game of cop and robber on $\mathcal{G}$ by first selecting start vertex v. From then onward, whenever it is the cop's turn, in order to establish the appropriate move to play given a current configuration $P = (u_c, u_r, c, t)$, the cop constructs from it a state S_P, and checks the u'_c component of the state $\sigma_c(S_P) = (u'_c, u'_r, r, t)$. It is guaranteed that u'_c is adjacent to u_c (or possibly $u'_c = u_c$) during t, due to the way the transformation from $\mathcal{G}$ to $\beta(\mathcal{G})$ has been defined.

In the situation in which $b = \mathsf{NO}$ we know that $\mathcal{G}$ is robber-win, and thus by Lemma 11 for every $v \in V(\mathcal{G})$, there exists at least one vertex u_v such that the state $(v, u_v, c, 0)$ is not in $\text{Attr}(F)$. Thus, the initial stage of our strategy for the robber involves the construction of a mapping $\sigma_r^0 : V(\mathcal{G}) \to V(\mathcal{G})$ from each possible $v \in V(\mathcal{G})$ that the cop might choose as its start vertex to a vertex u_v satisfying the aforementioned non-membership condition. Application of Theorem 13 then allows us to construct a memoryless winning strategy, σ_r, from all states $S \in (V' \setminus \text{Attr}(F)) \cap V_0$. The construction of σ_r^0 involves checking, for each of n possible start vertices v for the cop, at most n vertices u in order to identify which combination of v and u satisfies $(v, u, c, 0) \notin \text{Attr}(F)$. Hence, this initial phase can take at most $O(n^2)$ time. Similar to before, the algorithm of Theorem 13 can take at most $O(|V'| + |E'|) = O(\text{lcm}(L_{\mathcal{G}}) \cdot n^3)$ time, and hence we have that the overall construction of a strategy for the robber can take at most $O(\text{lcm}(L_{\mathcal{G}}) \cdot n^3)$ time, as claimed. Finally, interpreting σ_r as a strategy for the robber in the game of cop and robber on $\mathcal{G}$ is the same as for the cop above; the only difference is the way in which the start vertex is selected – the robber waits until the cop has selected a start vertex v and then chooses its own start vertex as $\sigma_r^0(v)$. □

Finally, we remark that Theorems 5 and 14 can be generalized to the setting with k cops at the expense of increasing the algorithm's running time (and space usage) to $O(\text{lcm}(L_{\mathcal{G}}) \cdot k \cdot n^{k+2})$. We fix an arbitrary ordering of the cops and create $k+1$ layers of states during every time step $t \in [0, \text{lcm}(L_{\mathcal{G}}) - 1]$: one for each of the k cops' moves, followed finally by the robber's move. By allowing in each time step for the players to play their moves in this serialized fashion, the resulting game graph requires $O(\text{lcm}(L_{\mathcal{G}}) \cdot k)$ layers with n^{k+1} states in each, with at most n edges leading from every state to states in the following layer.

4.2. A PSPACE *algorithm for* Periodic Cop & Robber

The algorithm presented in the previous subsection can use exponential time and space, as $\mathrm{lcm}(L_{\mathcal{G}})$ can be exponential in the size of the representation of the given periodic cop and robber game. Therefore, Theorem 5 only shows that Periodic Cop & Robber is contained in EXPTIME. In the following, we show that it is possible to solve the Periodic Cop & Robber problem using only polynomial space.

Theorem 15. Periodic Cop & Robber *is contained in* PSPACE*.*

For a given edge-periodic graph $\mathcal{G} = (V, E, \tau)$, the algorithm in Section 4.1 in some sense 'unrolls' $\mathcal{G}$ into a directed game graph $G' = \beta(\mathcal{G})$ that contains states for all time steps in $[0, \mathrm{lcm}(L_{\mathcal{G}})-1]$. Storing G' in memory may require exponential space. Note that this exponential blow-up comes from unrolling the edge-periodic graph into a TVG with global periodicity $\mathrm{lcm}(L_{\mathcal{G}})$. In a different framework of TVGs that allows only global periodicity, such as the setting in [21, 25], the input would already consist of $\mathrm{lcm}(L_{\mathcal{G}})$ snapshots and thus the algorithm from Section 4.1 (or the more recent algorithm by De Carufel et al. [25], with essentially the same running-time) would actually run in polynomial time and space. In our framework, however, the periodicity is specified on a per-edge basis, and hence further work is needed to obtain a PSPACE algorithm. The idea of our approach is to prove that it is enough to consider $n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$ time steps, to unroll $\mathcal{G}$ into a directed game graph with states for all time steps in $[0, n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1]$, and to show that we can identify the states in the attractor by a backward computation that holds in memory only the states for two consecutive time steps.

We begin by giving an upper bound on the maximum number of time steps (or rounds) that may be necessary for the cop to catch the robber in an edge-periodic graph $\mathcal{G}$ that is cop-winning.

Lemma 16. *Let $\mathcal{G} = (V, E, \tau)$ be an edge-periodic graph. If $\mathcal{G}$ is cop-winning, then the robber can be caught within at most $n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$ rounds.*

Proof. Assume that the cop uses a deterministic winning strategy that minimizes the latest possible time when the robber is caught. Consider a play ϕ in which the robber can evade the cop for as long as possible, and the configurations (u, v, s, t) (cf. Definition 6) that arise during that play. If the play consists of more than $n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$ rounds, then there must be two configurations (u_1, v_1, c, t_1) and (u_2, v_2, c, t_2) with $u_1 = u_2$, $v_1 = v_2$ and $t_2 = t_1 + \ell\, \mathrm{lcm}(L_{\mathcal{G}})$

for some integer $\ell \geq 1$. These two configurations are equivalent, and the cop has made no progress towards capturing the robber in between these configurations. This means that the configurations in time steps $t_1, t_1+1, \ldots, t_2-1$ could be removed from the play, yielding a shorter play that ends in a cop-winning configuration, a contradiction. □

Proof of Theorem 15. As in the proof of Theorem 5, we reduce Periodic Cop & Robber to the problem of computing the attractor set in a reachability game, but this time we do not build the whole game graph for the reachability game explicitly. Let $\mathcal{G} = (V, E, \tau)$ be the input edge-periodic graph. By Lemma 16, we know that if $\mathcal{G}$ is cop-winning, then it is sufficient to consider plays consisting of at most $2n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$ moves (the factor 2 is due to the alternation of players).

Construct a directed game graph $G'' = (V'', E'', F)$ as follows: V'' consists of all configurations (u, v, s, t) with $u, v \in V(\mathcal{G})$, $s \in \{c, r\}$, $t \in [0, n^2 \mathrm{lcm}(L_{\mathcal{G}}) - 1]$. The set $F \subseteq V''$ of target configurations consists of all states $(u, v, s, t) \in V''$ such that $u = v$. The outgoing edges of any state (u, v, s, t) are defined as in Section 4.1.1, except that in condition (3) we remove the modulo operation for the time steps, i.e., we replace the condition $t' = (t+1) \bmod \mathrm{lcm}(L_{\mathcal{G}})$ by $t' = t + 1$. The states $(u, v, s, t) \in V''$ with $s = c$ form V_0 and the remaining states form V_1. We can view G'' as a directed acyclic graph with $2n^2 \mathrm{lcm}(L_{\mathcal{G}})$ levels: For each time step t, the states (u, v, s, t) with $s = c$ form the first level, and the states (u, v, s, t) with $s = r$ form the second level of that time step. Each edge is directed from a state (u, v, c, t) to a state (u', v', r, t) or from a state (u, v, r, t) to a state $(u', v', c, t+1)$. Hence, G' is a graph with $2n^2 \mathrm{lcm}(L_{\mathcal{G}})$ levels, and each edge connects a state in one level to a state in the next level. Furthermore, each level contains only n^2 states. The number of levels of the graph G'' is large enough to contain any path that corresponds to a play of the game with at most $n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$ rounds. By Lemma 16, if $\mathcal{G}$ is cop-winning then this is sufficient for containing all plays that result from a cop-winning strategy that minimizes the number of time steps until the robber is caught.

The Periodic Cop & Robber problem can be solved by checking if there is some vertex u_c such that for all vertices $u_r \in V$, the node $(u_c, u_r, c, 0) \in V''$ is in the attractor set $\mathrm{Attr}(F)$ in G''. Note that the attractor set of G'' corresponds to the set of configurations from which the cop has a winning strategy. We will now prove that this check can be implemented in polynomial space. Note that in G'' only states with identical time steps or with consecutive time steps t and $t+1$ are connected. Hence, in order to compute which nodes with time step t belong to the attractor set, we need to know only which nodes with time step $t+1$ belong to the attractor set. Since G'' is a directed acyclic graph, we can start the computation of the attractor set in the level with $t = n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1$ and $s = r$:

$$\mathrm{Attr}_r^{n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})-1} := \{(u_c, u_r, r, n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1) \mid u_c = u_r\}$$

Once Attr_r^t has been computed for some $t \in [0, n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1]$, we can compute Attr_c^t, and Attr_r^{t-1} if $t > 0$, as follows:

$$\begin{aligned}
\mathrm{Attr}_c^t &:= \{(u_c, u_r, c, t) \mid \exists((u_c, u_r, c, t), (u_c', u_r, r, t)) \in E'' \colon (u_c', u_r, r, t) \in \mathrm{Attr}_r^t\} \\
&\quad \cup \{(u_c, u_r, c, t) \mid u_c = u_r\}, \text{ for } n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1 \geq t \geq 0, \\
\mathrm{Attr}_r^{t-1} &:= \{(u_c, u_r, r, t-1) \mid \forall((u_c, u_r, r, t-1), (u_c, u_r', c, t)) \in E'' \colon (u_c, u_r', c, t) \\
&\quad \in \mathrm{Attr}_c^t\} \cup \{(u_c, u_r, r, t-1) \mid u_c = u_r\}, \text{for } n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1 \geq t \geq 1.
\end{aligned}$$

For each time step t, $n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}}) - 1 \geq t \geq 0$, we only need to keep the previously handled time step $t+1$ (if existent)[4] of $\mathcal{G}$ in memory in order to compute the corresponding levels of G'' and the sets Attr_c^t and Attr_r^t of nodes (u_c, u_r, s, t) in G'' from which the cop has a winning strategy. In particular, at any time we only need to keep the sets Attr_c^t, Attr_r^t, Attr_c^{t+1}, Attr_r^{t+1} in memory for some value of t, yielding a polynomial space algorithm. Note that $\bigcup_{0 \leq t \leq n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})-1} \mathrm{Attr}_c^t \cup \mathrm{Attr}_r^t = \mathrm{Attr}(F)$. In order to check if there is some vertex u_c such that for all vertices $u_r \in V$, the node $(u_c, u_r, c, 0) \in V''$ is in $\mathrm{Attr}(F)$, we only need to consider the set Attr_c^0. □

We remark that the running-time of the algorithm from Theorem 15 is $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^5)$, as it constructs the graph with $O(\mathrm{lcm}(L_{\mathcal{G}}) \cdot n^2)$ levels on-the-fly while only keeping a constant number of levels in memory. Each level contains $O(n^2)$ nodes. Hence, processing the level takes time $O(n^3)$ as each vertex in a level has at most n outgoing edges.

Finally, we observe that Theorem 15 can be generalized to the case of $k \geq 2$ cops using similar ideas as those discussed at the end of Section 4.1.

[4]Note that we can easily compute the snapshot $\mathcal{G}(n^2 \cdot \mathrm{lcm}(L_{\mathcal{G}})) = \mathcal{G}(0)$ by including all edges with $\tau(e)[0]^{\circ} = 1$; and from $\mathcal{G}(t)$ for some time step t, the snapshot $\mathcal{G}(t-1)$ by shifting the pointer in each $\tau(e)$ one step to the left. Therefore, we can compute from each snapshot $\mathcal{G}(t+1)$ the snapshot $\mathcal{G}(t)$ in polynomial time and space.

For each time step t, the directed game graph G'' now has $k+1$ levels, each with n^{k+1} vertices. Furthermore, it is sufficient to build G' for $n^{k+1}\operatorname{lcm}(L_{\mathcal{G}})$ time steps, as can be shown by a suitable adaptation of Lemma 16. It still suffices to keep in memory only the snapshots of $\mathcal{G}$ from two consecutive time steps, and two consecutive levels of G''. The running-time of this algorithm is $O(\operatorname{lcm}(L_{\mathcal{G}}) \cdot kn^{2k+3})$, as the game graph G'' has $(k+1)n^{k+1}\operatorname{lcm}(L_{\mathcal{G}})$ levels, and each level has $O(n^{k+1})$ nodes with at most n outgoing edges each. The space usage is $O(n^{k+2})$ as we build the graph on-the-fly. If k is bounded by a constant, this yields a PSPACE algorithm for the setting with k cops.

5. What length makes an edge-periodic cycle robber-winning?

In this section, we consider restricted subclasses of edge-periodic graphs, namely directed and undirected edge-periodic cycles. For edge-periodic cycles $\mathcal{G}$ with n vertices, we study the question of how large n must be at least, in dependence on $\operatorname{lcm}(L_{\mathcal{G}})$ and $\max L_{\mathcal{G}}$, to guarantee that $\mathcal{G}$ is robber-winning. Note that $\operatorname{lcm}(L_{\mathcal{G}})$ represents the global period of $\mathcal{G}$ and $\max L_{\mathcal{G}}$ the longest edge period.

5.1. Infinite edge-periodic paths

First, as an auxiliary result, we show that any edge-periodic infinite path whose edge periods originate from a set of integers $L_{\mathcal{G}}$ of finite size is robber-winning. In particular, we show that it suffices for the robber to place himself a certain number of edges ahead of the cop initially. This auxiliary result will allow us later to also handle the case in which the cop chases the robber around a cycle in a fixed direction.

Lemma 17. *Let $\mathcal{G}$ be an infinite edge-periodic path, $L_{\mathcal{G}} = \{|\tau(e)| \mid e \in E(\mathcal{G})\}$, and assume that $|L_{\mathcal{G}}|$ is finite. Then, starting from any time step t, there exists a winning strategy for the robber from any vertex with distance at least $2 \cdot \operatorname{lcm}(L_{\mathcal{G}})$ from the cop's start vertex if $\operatorname{lcm}(L_{\mathcal{G}}) = \max L_{\mathcal{G}}$, and from any vertex with distance at least $\operatorname{lcm}(L_{\mathcal{G}})$ from the cop's start vertex otherwise.*

Proof. First, notice that since we assume that $|L_{\mathcal{G}}|$ is finite, so must be $\operatorname{lcm}(L_{\mathcal{G}})$. Let the cop pick its initial vertex $c_t \in V(\mathcal{G})$. Let the robber's initial vertex be denoted by r_t, and assume without loss of generality that r_t will be some vertex that lies to the right of c_t in the underlying graph $P = (V, E)$ of $\mathcal{G}$, which we imagine as a directed path that extends infinitely towards the right. As vertices to the left of c_t are irrelevant, we will from here onward denote by P the path starting at c_t and extending infinitely to the right.

Consider the set $L_\mathcal{G}$ and its constituent elements. There are two cases – either (1) there exists $x \in L_\mathcal{G}$ such that $\max L_\mathcal{G}$ is not a multiple of x – then, $\mathrm{lcm}(L_\mathcal{G}) \geq 2 \cdot \max L_\mathcal{G}$, since it cannot be the case that $\mathrm{lcm}(L_\mathcal{G}) = j \cdot \max L_\mathcal{G}$ for any $j < 2$; or (2) for every $x \in L_\mathcal{G}$, $\max L_\mathcal{G} = x \cdot i$ for some integer $i \geq 1$; then, $\mathrm{lcm}(L_\mathcal{G}) = \max L_\mathcal{G}$. With this in mind, define $B = \mathrm{lcm}(L_\mathcal{G})$ if (1) holds and $B = 2 \cdot \mathrm{lcm}(L_\mathcal{G})$ if (2) holds. Now, let us define the *strips* S_i $(i \geq 1)$ to be finite subpaths of P, such that for all edges $e \in S_i$, e can first be traversed by the cop in some time step $t_e \in [t+(i-1)B, t+iB-1]$. Note that $B \geq 2 \cdot \max L_\mathcal{G}$ and hence each S_i must contain at least two edges. By convention, we call the leftmost and rightmost edges (vertices) of any S_i its *first* and *last* edges (vertices), respectively. Note also that the last vertex of S_i and the first vertex of S_{i+1} are one and the same, for all $i \geq 1$.

Assume from now on that the cop moves right whenever possible. It is safe to do so since, otherwise, the cop may only be positioned at the same vertex or further left than when following this strategy. The strategy for the robber is as follows: pick r_t to be the first vertex of S_2 and move right (i.e., away from the cop) whenever possible.

We now demonstrate that the robber's strategy is a winning one. Let $T_x^F(i)$ and $T_x^L(i)$ denote the first time step in which player $x \in \{c, r\}$ is able to traverse the first/last edge of S_i, respectively. Note that $T_c^F(i) \geq t+(i-1)B$ and that $T_r^L(i) \leq t+(i-1)B-1$. Combining the two gives that $T_r^L(i) < T_c^F(i)$, which implies that the cop can never catch the robber in any time step in which the edge leading to both player's right belongs to S_i.

We next show that the robber cannot be caught when the edge leading to the cop's right belongs to S_i and the edge leading to the robber's right belongs to S_{i+1}. Let $M = \max L_\mathcal{G}$ and recall that $T_c^F(i) \geq t + (i-1)B$. Since the strips S_i are defined to consist of all edges crossed in the period $[t+(i-1)B, t+iB-1]$, and since $B \geq 2M$, it follows that at time $T_r^F(i+1) \leq t+(i-1)B+M-1$, there is at least one more edge of S_i that remains to be crossed by the cop. This gives that $T_c^L(i) > T_r^F(i+1)$ and yields the claim. Combining this with the earlier observation that $T_r^L(i) < T_c^F(i)$, it follows that there exists a winning strategy for the robber starting from the first vertex of S_2.

Finally, recall that when $\mathrm{lcm}(L_\mathcal{G}) = \max L_\mathcal{G}$, we have that each S_i consists of at most $2 \cdot \mathrm{lcm}(L_\mathcal{G})$ edges, and otherwise it consists of at most $\mathrm{lcm}(L_\mathcal{G})$ edges. Hence, there exists a winning strategy for the robber starting from some vertex $r_t \in P$ with distance at most $2 \cdot \mathrm{lcm}(L_\mathcal{G})$ or $\mathrm{lcm}(L_\mathcal{G})$ from c_t, depending on the condition satisfied by $\operatorname{lcm}(L_{\mathcal{G}})$. The lemma follows by noticing that the above strategy also works when the robber is initially positioned at any vertex further to the right than the first vertex of S_2. $\square$

5.2. Directed edge-periodic cycles

In the following, assume that we are given a directed edge-periodic cycle $\mathcal{G} = (V, E, \tau)$, i.e., an edge-periodic graph whose underlying graph is a directed cycle.

Theorem 18. *Let $\mathcal{G} = (V, E, \tau)$ be a directed edge-periodic cycle with n vertices. Then, $\mathcal{G}$ is robber-winning if $n > \operatorname{lcm}(L_{\mathcal{G}})$ in case $\max L_{\mathcal{G}} < \operatorname{lcm}(L_{\mathcal{G}})$, and, if $n > \operatorname{lcm}(L_{\mathcal{G}}) + 1$ in case $\max L_{\mathcal{G}} = \operatorname{lcm}(L_{\mathcal{G}})$. These bounds are best possible.*

Note that the case $\max L_{\mathcal{G}} < \operatorname{lcm}(L_{\mathcal{G}})$ is only possible if $k = \operatorname{lcm}(L_{\mathcal{G}})$ contains at least two distinct prime factors. Therefore, the smallest k for which the case can arise is $k = 6$.

First, we observe that the case where $\operatorname{lcm}(L_{\mathcal{G}}) = 1$ is trivial: All edge periods must be equal to 1 in this case, so every edge is present in every time step. Then, it is easy to see that the cycle is cop-winning if $n \leq 2$ and robber-winning if $n \geq 3$. This proves Theorem 18 for $\operatorname{lcm}(L_{\mathcal{G}}) = 1$. Therefore, we only consider the case where $\operatorname{lcm}(L_{\mathcal{G}}) \geq 2$ in the following.

Theorem 18 then follows from the following four lemmas. The first two lemmas show that directed cycles are robber-winning if n is large enough. First, we show that any edge-periodic cycle $\mathcal{G}$ with $n > \operatorname{lcm}(L_{\mathcal{G}}) + 1$ is robber-winning. This results holds no matter whether $\max L_{\mathcal{G}} = \operatorname{lcm}(L_{\mathcal{G}})$ or $\max L_{\mathcal{G}} < \operatorname{lcm}(L_{\mathcal{G}})$. After this, in Lemma 21, we will show a slightly improved bound for the latter case.

Lemma 19. *Let $k \geq 2$ and let $\mathcal{G} = (V, E, \tau)$ be a directed edge-periodic cycle with $\operatorname{lcm}(L_{\mathcal{G}}) = k$ and n vertices. The directed edge-periodic cycle $\mathcal{G}$ is robber-winning if $n > k + 1$.*

Proof. Let $k \geq 2$ and let $\mathcal{G} = (V, E, \tau)$ be a directed edge-periodic cycle with $\operatorname{lcm}(L_{\mathcal{G}}) = k$ and $n > \operatorname{lcm}(L_{\mathcal{G}}) + 1$ vertices. We show that $\mathcal{G}$ is robber-winning. The robber should choose his starting vertex as the vertex directly behind the cop. Assume towards a contradiction that $\mathcal{G}$ is cop-winning. Then, there is some strategy for the cop, such that there is a latest time step t_0, where the robber is on the vertex r_0 directly behind the vertex c_0 of the cop, that is, where $(r_0, c_0) \in E$. Hence, the cop traverses the unique outgoing edge from c_0 in time step t_0.

Next, we contradict the cop-winning strategy by using the following claim.

Claim 20. *For each $i \geq 0$, let $t_i := t_0 + i \cdot \mathrm{lcm}(L_{\mathcal{G}})$, and let c_i denote the position of the cop at time step t_i. Assume that, at time step t_i, the position of the robber r_i is equal to the position of the cop at time step t_{i-1} (i.e., $r_i = c_{i-1}$) if $i > 0$ or that $(r_i, c_i) \in E$ if $i = 0$. Then the robber has a strategy to a) end his turn on vertex c_i at the end of time step $t_i + \mathrm{lcm}(L_{\mathcal{G}}) - 1 = t_{i+1} - 1$ without getting caught by the cop or b) reach the vertex directly behind the cop at some time step between t_i and t_{i+1}.*

Proof. We show this statement via induction over i. If the cop does not traverse the outgoing edge from c_0 at time step t_0, b) is satisfied directly. Hence, assume in the following that the cop traverses the outgoing edge from c_0 at time step t_0.

Recall that $n > \mathrm{lcm}(L_{\mathcal{G}}) + 1$. Hence, the cop cannot reach vertex r_0 within the next $\mathrm{lcm}(L_{\mathcal{G}})$ time steps since she has to traverse at least $\mathrm{lcm}(L_{\mathcal{G}}) + 1$ edges. Moreover, since we assume that each edge label contains at least one 1 and the label of $e := (r_0, c_0)$ has length at most $\mathrm{lcm}(L_{\mathcal{G}})$, there is a time step $j \in [0, \mathrm{lcm}(L_{\mathcal{G}}) - 1]$ such that $\tau(e)[t_0 + j]^\circ = 1$ and thus the robber can reach vertex c_0 at the end of time step $t_0 + j$ and wait there until the beginning of time step t_1, which fulfills a).

Next, we show the inductive step. Let $i > 0$. We show that, if the robber repeats the moves of the cop from $\mathrm{lcm}(L_{\mathcal{G}})$ time steps earlier, the robber achieves a) or b). Let $P = (x_0, \dots, x_{d_i})$ denote the unique path from $c_{i-1} = r_i$ to c_i, where $x_0 = c_{i-1}$ and $x_{d_i} = c_i$. Let d_i denote the number of edges of P. Note that d_i equals the number of edges that the cop traversed in the preceding $\mathrm{lcm}(L_{\mathcal{G}})$ time steps. Since $n > \mathrm{lcm}(L_{\mathcal{G}}) + 1$, for each $j \in [0, d_i]$, the cop has to traverse at least $\mathrm{lcm}(L_{\mathcal{G}}) + 2 - d_i + j$ edges to reach vertex x_j starting from time t_i. Hence, for each $j \in [0, d_i]$, the earliest time step in which the cop can reach vertex x_j is time step $t_i + \mathrm{lcm}(L_{\mathcal{G}}) - d_i + j$. Since $r_i = c_{i-1}$ and the cop moved from c_{i-1} to c_i between time step t_{i-1} and $t_i - 1$, the robber can also traverse any edge $e_j = (x_{j-1}, x_j)$ in time step $t_i + \ell$ if the cop traversed the edge e_j in time step $t_{i-1} + \ell = t_i + \ell - \mathrm{lcm}(L_{\mathcal{G}})$, for any $j \in [1, d_i]$ and $\ell \in [0, \mathrm{lcm}(L_{\mathcal{G}}) - 1]$. Since the cop was able to reach vertex c_i at the latest at time step $t_i - 1$, the latest possible time step in which the cop traversed the edge (x_{j-1}, x_j) was time step $t_{i-1} + \mathrm{lcm}(L_{\mathcal{G}}) - d_i + j - 1$ for each $j \in [1, d_i]$. Hence, for each $j \in [1, d_i]$, the robber can reach vertex x_j at the latest at time step $t_i + \mathrm{lcm}(L_{\mathcal{G}}) - d_i + j - 1$, while the earliest possible time step the cop can reach vertex x_{j-1} is time step $t_i + \mathrm{lcm}(L_{\mathcal{G}}) - d_i + j - 1 + 1$. Thus, with this strategy the robber can reach vertex c_i and the cop cannot catch the robber in any time step between t_i and $t_{i+1} - 1$ which fulfills a), except if the robber would run into the cop, that is, if the robber would traverse the edge (u, v), where the cop is currently at vertex v. In this case, the robber is directly behind the cop, which satisfies b). $\triangleleft$

Hence, the robber can reach the vertex behind the cop at time step $t^* > t_0$ or the robber can evade the cop indefinitely. In both cases, this contradicts the existence of the described winning strategy of the cop. Thus, $\mathcal{G}$ is robber-winning. $\square$

Lemma 21. *Let $k \geq 6$ be a number with at least two distinct prime factors. Let $\mathcal{G} = (V, E, \tau)$ be a directed edge-periodic cycle with $\max L_{\mathcal{G}} < \mathrm{lcm}(L_{\mathcal{G}}) = k$ and n vertices. If $n > k$, then the directed edge-periodic cycle $\mathcal{G}$ is robber-winning.*

Proof. We describe a winning strategy for the robber. After the cop has placed herself, the robber chooses the vertex directly behind the cop's starting vertex as his starting vertex. In all future time steps, the robber traverses the unique outgoing edge from his current vertex whenever the edge exists in that time step, unless this traversal would cause him to run into the cop. We show that this strategy is robber-wining.

We call a time step *safe* if the robber is located on the vertex directly behind the cop at the start of the time step. If, during a safe time step t, the cop and the robber both traverse their outgoing edge or both remain at their current vertex, then it is clear that time step $t + 1$ is also safe. A maximal period of consecutive safe time steps is called a *safe period*. It is clear that the cop cannot catch the robber during a safe period.

Now, consider the case that time step t is safe but time step $t + 1$ is not safe. This happens if the cop can traverse her outgoing edge in time step t, but the robber is forced to remain at his vertex because his outgoing edge is absent in time step t. In this case we say that a *chase period* starts in time step t. Note that time step t belongs both to the safe period ending at time step t and to the chase period starting at time step t. If there is a time step $t' > t$ that is safe, then the chase period ends at time step $t' - 1$ and a new safe period begins at time step t'.

If a chase period that starts at some time step t does not last forever, there are two possibilities how it could end:

(1) There is a smallest time step $t' > t$ at the start of which the robber is located at the vertex behind the cop. Then t' is a safe time step, and a new safe period starts at time step t'.

(2) The cop catches the robber: There is a time step $t' > t$ in which the cop traverses her outgoing edge and reaches the vertex on which the robber is located.

We claim that (2) cannot happen. Let t be the time step in which the chase period begins. Assume that the robber is located at vertex r_t and the cop at vertex c_t in time step t, with $(r_t, c_t) \in E$. Consider the directed infinite path P_t that is obtained by the unique infinite walk in G starting at c_t. Intuitively, we unroll the directed cycle G infinitely many times, starting at vertex c_t. Let c'_t denote the start vertex of P_t and r'_t the first occurrence of r_t in P_t. Let $\mathcal{P}_t$ be the infinite edge-periodic path with underlying graph P_t, with edge labels inherited from $\mathcal{G}$ in the obvious way. Throughout the chase period, the moves by the cop and the robber in the directed cycle correspond to moves in the infinite edge-periodic path $\mathcal{P}_t$, with the cop starting at c'_t and the robber at r'_t. Furthermore, by the definition of the robber strategy, the robber traverses his outgoing edge whenever possible throughout the chase period. If the cop were to catch the robber during the chase period in $\mathcal{G}$, the cop would also catch the robber in $\mathcal{P}_t$. There are $n - 1 \geq \mathrm{lcm}(L_{\mathcal{G}}) = \mathrm{lcm}(L_{\mathcal{P}_t})$ edges between the cop's vertex c'_t and the robber's vertex r'_t at the beginning of time step t. Furthermore, as we have $\mathrm{lcm}(L_{\mathcal{G}}) > \max L_{\mathcal{G}}$, we also have $\mathrm{lcm}(L_{\mathcal{P}_t}) > \max L_{\mathcal{P}_t}$. Hence, by Lemma 17, the cop cannot catch the robber in $\mathcal{P}_t$. Thus, we have shown that (2) cannot happen. This means that the chase period either continues indefinitely without the cop catching the robber, or we eventually enter a safe time step t'.

The argument above applies to any chase period. Hence, it follows that $\mathcal{G}$ is robber-winning. □

Lemma 22. *For every $k \geq 2$, there exists a cop-winning directed edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\max L_{\mathcal{G}} = \mathrm{lcm}(L_{\mathcal{G}}) = k$ and $n = \mathrm{lcm}(L_{\mathcal{G}}) + 1$ vertices.*

Proof. Let $k \geq 2$. We show that the cop has a winning strategy on the directed edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ for which the underlying graph consists of the directed cycle $(v_0, \ldots, v_k)$, where the edge (v_{i-1}, v_i) is labeled with a constant 1 for each $i \in [1, k]$ and the edge (v_k, v_0) is labeled with $0^{k-1}1$.

By construction, starting at vertex v_0 at time step 0, the cop can traverse edge (v_i, v_{i+1}) in time step i for each $i \in [0, k-1]$, whereas the first time step in which the edge (v_k, v_0) can be traversed is time step $k-1$. Since the cop always moves first in each time step, the cop thus needs at most $k-1$ time steps to catch the robber. $\square$

Lemma 23. *Let $k \geq 6$ be a number with at least two distinct prime factors. There exists a cop-winning directed edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\max L_{\mathcal{G}} < \operatorname{lcm}(L_{\mathcal{G}}) = k$ and $n = k$ vertices.*

Proof. Let $p_0, \ldots, p_{r-1}$ be the prime factorization of k, i.e., for each distinct prime p that divides k there is a unique i such that $p_i = p^{m_p}$, where p^{m_p} is the largest power of p that divides k. For example, if $k = 200$, then we have $r = 2$, $p_0 = 2^3 = 8$ and $p_1 = 5^2 = 25$. Note that $r \leq \frac{k}{2}$. We set $V := \{v_0, \ldots, v_{k-1}\}$ and $E := \{e_i := (v_i, v_{i+1}) \mid i \in [0, k-2]\} \cup \{e_{k-1} := (v_{k-1}, v_0)\}$. Moreover, for each $i \in [0, k-1]$, we set the label x_i of the edge e_i to be the string of length $p_{(i \bmod r)}$ containing a single 1 at position $i \bmod |x_i|$. Note that for each $i \in [0, k-1]$, $x_i[i]^\circ = 1$. Hence, starting at vertex v_0 at any time step t divisible by $k = \operatorname{lcm}(L_{\mathcal{G}})$, the cop can traverse edge e_i in time step $t + i$ for each $i \in [0, k-1]$ and end back on vertex v_0 at the end of time step $t + k - 1$. In other words, the cop has a strategy in which she can always immediately traverse the unique outgoing edge. To show that $\mathcal{G}$ is cop-winning, it thus remains to show that starting from any vertex v_ℓ distinct from v_0 at any time step t divisible by k, the robber cannot traverse each edge of the graph in the next k time steps.

Let $\ell \in [1, k-1]$. We consider the edge labels $y_0, \ldots, y_{r-1}$ of r specific consecutive edges out of the total of k edges of the cycle that the robber has to traverse in order to arrive back at vertex v_ℓ. We choose the vertices $y_0, \ldots, y_{r-1}$ by setting $y_i := x_i$ for all $i \in [0, r-1]$ if $\ell \geq r$ and by setting $y_i := x_{\ell+i}$ for all $i \in [0, r-1]$ otherwise. That is, if $\ell \geq r$, we consider the edge labels of the unique path with r edges starting in v_0 and if $\ell < r$, we consider the edge labels of the unique path with r edges starting in v_ℓ. Note that the set $\{|y_i| \mid i \in [0, r-1]\}$ of lengths of these edge labels is exactly the set of prime factors $\{p_i \mid i \in [0, r-1]\}$ of k, and each edge label y_i contains exactly one 1. Hence, due to the Chinese Remainder Theorem and the fact that the length of any pair of distinct edge labels y_i and y_j are coprime, there is exactly one integer $t' \in [0, \operatorname{lcm}(p_0, \ldots, , p_r) - 1] = [0, k-1]$ such that for each $i \in [0, r-1]$, $y_i[t' + i]^\circ = 1$. By construction and the above argumentation, t' is the time step in which the cop may traverse this edge

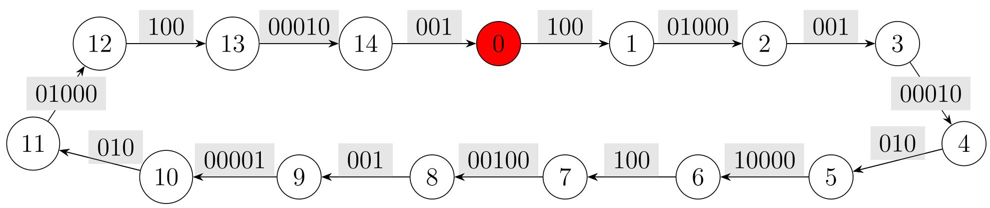


Figure 4: Directed edge-periodic cycle for the case $k = 3 \cdot 5$ in Lemma 23 with $3 \cdot 5 = 15$ vertices and $\operatorname{lcm}(L_{\mathcal{G}}) = 15$ with a cop-winning strategy from the start vertex marked in red. For each edge e, its label $\tau(e)$ is shown with gray background.

by starting at vertex v_0 at time step 0 and traversing one edge each time step. Consequently, the robber cannot traverse all edges assigned with the labels $y_0, \ldots, y_{r-1}$ without waiting at least one time step at some endpoint of the respective edges. As a consequence, the cop has a strategy to always reduce the distance to the robber by at least one within k time steps. Hence, after at most k^2 time steps, the cop can catch the robber, and thus $\mathcal{G}$ is cop-winning. Figure 4 illustrates the construction for the example $k = 3 \cdot 5$. □

5.3. Undirected edge-periodic cycles

In this section, assume that we are given an edge-periodic cycle $\mathcal{G} = (V, E, \tau)$. First, we give an upper bound on the length n of $\mathcal{G}$ that guarantees that $\mathcal{G}$ is robber-winning.

Theorem 24. *Let $\mathcal{G} = (V, E, \tau)$ be an edge-periodic cycle on n vertices and $L_{\mathcal{G}} = \{|\tau(e)| \mid e \in E\}$. Then, if $n \geq 2 \cdot \ell \cdot \operatorname{lcm}(L_{\mathcal{G}})$, $\mathcal{G}$ is robber-win, where $\ell = 1$ if $\operatorname{lcm}(L_{\mathcal{G}}) \geq 2 \cdot \max L_{\mathcal{G}}$, and $\ell = 2$ otherwise.*

Proof. We let c_t and r_t denote the vertex at which the cop and the robber are positioned at the start of time step t, respectively. Consider now some edge $e \in E(\mathcal{G})$ and classify its vertices as a 'left' and 'right' vertex arbitrarily; let the left vertex of each edge be the right vertex of the following edge in the cycle. Furthermore, we say that two vertices $u, v \in V$ are *antipodal* (in $\mathcal{G}$) if their distance in the cycle (V, E) is maximum, i.e., equal to $\lfloor n/2 \rfloor$. If n is even, every vertex has exactly one antipodal vertex; if n is odd, every vertex has two antipodal vertices. We proceed by specifying a strategy for the robber. Initially, let the cop choose c_0; the robber chooses r_0 to be a vertex antipodal to c_0 in $\mathcal{G}$. (If n is odd, the robber can select r_0 to be either of the two vertices that are antipodal to c_0.) We now distinguish between two modes of play, *Hide* and *Escape*, and specify the robber's strategy in each of them.

Hide mode: The first *Hide period* begins in time step 0, and a further Hide period begins in every time step $t \geq 2$ such that c_t and r_t are antipodal, but c_{t-1} and r_{t-1} were not. As such, any game in which the robber follows our strategy begins in a Hide period. The Hide period beginning at time step t consists of the time steps in the interval $[t, t+x]$ such that $c_{t'}$ and $r_{t'}$ are antipodal for each $t' \in [t, t+x]$ but c_{t+x+1} and r_{t+x+1} are not. Any Hide period is followed directly by an escape period, which will start in time step $t+x+1$.

The robber's **Hide strategy**: If the game is in a Hide period during time step t, the robber should observe the cop's choice of c_{t+1}, and always try to move to a vertex antipodal to it. We claim that the robber cannot be caught in any time step belonging to a Hide period. To see this, observe that regardless of whether $\mathrm{lcm}(L_{\mathcal{G}}) = \max L_{\mathcal{G}}$ or $\mathrm{lcm}(L_{\mathcal{G}}) \geq 2 \cdot \max L_{\mathcal{G}}$, we have that $n \geq 4 \cdot \max L_{\mathcal{G}} \geq 4$. As a result, antipodal vertices in $\mathcal{G}$ are at least distance 2 apart from one another, and the claim follows.

Escape mode: An *Escape period* always begins in a time step t such that time step $t-1$ was the last time step of some Hide period. As such, an Escape period consists of time steps in the interval $[t, t+x]$ such that for each $t' \in [t, t+x]$ it holds that $c_{t'}$ and $r_{t'}$ are not antipodal but c_{t+x+1} and r_{t+x+1} are. The last time step of the Escape period is then $t+x$, and the first time step of the next Hide period is $t+x+1$.

The robber's **Escape strategy**: Assume that some Escape period starts in time step t. Then, at the start of time step $t-1$, c_{t-1} and r_{t-1} were antipodal to one another, and during time step $t-1$, we had a situation in which the cop was able to move towards the robber in some direction, but the edge incident to r_{t-1} leading in the same direction was not present. Now, recall that if $\ell = 2$, so that $\mathrm{lcm}(L_{\mathcal{G}}) = \max L_{\mathcal{G}}$, then $n \geq 4 \cdot \mathrm{lcm}(L_{\mathcal{G}})$; and if $\ell = 1$ so that $\mathrm{lcm}(L_{\mathcal{G}}) \geq 2 \cdot \max L_{\mathcal{G}}$, then $n \geq 2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$. Therefore, since c_{t-1} and r_{t-1} are antipodal in $\mathcal{G}$, if $\ell = 2$ holds we have that the distance between them is at least $2 \cdot \mathrm{lcm}(L_{\mathcal{G}})$ and if $\ell = 1$ holds, the distance between them is at least $\mathrm{lcm}(L_{\mathcal{G}})$. Observe now that we are able to view any edge-periodic cycle of finite length as an infinite path whose edge labels repeat infinitely often. Combining these two facts, it then follows from Lemma 17 that when the Escape period starts in time step t, there exists a strategy for the robber (which started in the previous time step from vertex r_{t-1}) that will enable him to evade the cop until the Escape period ends.

Finally, since every time step t belongs to either a Hide period or an Escape period, we have shown that the cop can never catch the robber, and the proof is complete. □

Next, we show that the bounds of Theorem 24 are best possible. First, we note that $\max L_{\mathcal{G}} = 1$ implies $\operatorname{lcm}(L_{\mathcal{G}}) = 1$ and hence $\ell = 2$. Then every edge is a 1-edge, and it is easy to see that an edge-periodic cycle with $n = 3$ nodes is cop-winning, showing that $2 \cdot \ell \cdot \operatorname{lcm}(L_{\mathcal{G}}) = 4$ nodes are indeed necessary to guarantee that the cycle is robber-winning. For the case $\max L_{\mathcal{G}} = 2$, it also follows that $\operatorname{lcm}(L_{\mathcal{G}}) = 2$ and hence $\ell = 2$. If $\max L_{\mathcal{G}} \geq 3$, we can have $\ell = 1$ or $\ell = 2$. We now present infinite families of cop-winning edge-periodic cycles with $n = 2 \cdot \ell \cdot \operatorname{lcm}(L_{\mathcal{G}}) - 1$ vertices for all values of $\max L_{\mathcal{G}} \geq 2$.

Theorem 25. *For $k = 2$ and $\ell = 2$, and for every $k \geq 3$ and $\ell \in \{1, 2\}$, there exists a cop-winning edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\max L_{\mathcal{G}} = k$ and $n = 2 \cdot \ell \cdot \operatorname{lcm}(L_{\mathcal{G}}) - 1$ vertices, where $\operatorname{lcm}(L_{\mathcal{G}}) \geq 2k$ if $\ell = 1$ and $\operatorname{lcm}(L_{\mathcal{G}}) = k$ otherwise.*

In order to prove Theorem 25, we give families of edge-periodic cycles for $\ell = 1$ and $\ell = 2$ separately, beginning with $\ell = 2$, i.e., the case that $\operatorname{lcm}(L_{\mathcal{G}}) < 2 \cdot \max L_{\mathcal{G}}$ and hence $\operatorname{lcm}(L_{\mathcal{G}}) = \max L_{\mathcal{G}}$.

Lemma 26. *For every $k \geq 2$ there exists an edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\operatorname{lcm}(L_{\mathcal{G}}) = k = \max L_{\mathcal{G}}$, and $n = 4k - 1$ vertices that is cop-winning.*

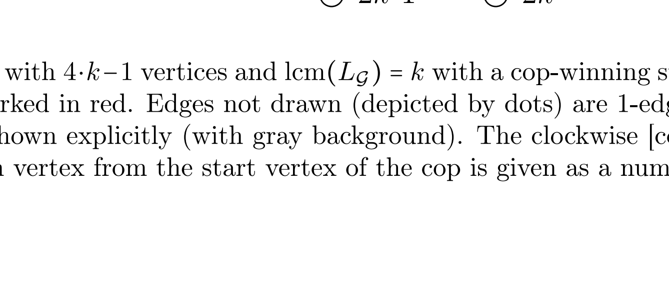
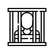


Figure 5: Cycle with $4 \cdot k - 1$ vertices and $\operatorname{lcm}(L_{\mathcal{G}}) = k$ with a cop-winning strategy from the start vertex marked in red. Edges not drawn (depicted by dots) are 1-edges; for all other edges, $\tau(e)$ is shown explicitly (with gray background). The clockwise [counterclockwise] distance of each vertex from the start vertex of the cop is given as a number preceded by ↻ [by ↺].

time step	pos. cop	pos. robber
s	0	$\circlearrowright\ 2k-1$
0	$\circlearrowright\ 1$	$\circlearrowright\ 2k-1$
$k-1$	$\circlearrowright\ k$	$\circlearrowright\ 2k$
$2k-3$	$\circlearrowright\ 2k-2$	$\circlearrowright\ 3k-2$
$2k-2$	$\circlearrowright\ 2k-1$	$\circlearrowright\ 3k-2$
$2k-1$	$\circlearrowright\ 2k$	$\circlearrowright\ 3k-2$
$2k$	$\circlearrowright\ 2k+1$	$\circlearrowright\ 3k-1$
$3k-3$	$\circlearrowright\ 3k-2$	$\circlearrowright\ 4k-4$
$3k$	$\circlearrowright\ 3k-1$	0
$3k+1$	$\circlearrowright\ 3k$	0
$4k-1$	$\circlearrowright\ 4k-2$	0
$4k$	0	

time step	pos. cop	pos. robber
s	0	$\circlearrowleft\ 2k-1$
$k-1$	$\circlearrowleft\ k$	$\circlearrowleft\ 2k$
k	$\circlearrowleft\ k+1$	$\circlearrowleft\ 2k+1$
$2k-2$	$\circlearrowleft\ 2k-1$	$\circlearrowleft\ 3k-1$
$2k-1$	$\circlearrowleft\ 2k$	$\circlearrowleft\ 3k$
$3k-3$	$\circlearrowleft\ 3k-2$	$\circlearrowleft\ 4k-2$
$3k$	$\circlearrowleft\ 3k+1$	0
$4k-3$	$\circlearrowleft\ 4k-2$	$\circlearrowleft\ k-3$
$4k$	0	$\circlearrowleft\ k$
$5k-1$	$\circlearrowleft\ k-1$	$\circlearrowleft\ k$
$5k$	$\circlearrowleft\ k$	

Table 1: Time steps with corresponding positions of cop and robber in the edge-periodic cycle depicted in Figure 5. All positions are *after* moving in this time step. The time step s denotes the start configuration. Recall that the cop moves first, and that the binary strings are indexed starting from 0, so that the cop can cross edge $\{\circlearrowright\ 2k-1, \circlearrowright\ 2k\}$ at time $2k-1$. Icon: Flaticon.com

Proof. Consider the edge-periodic cycle $\mathcal{G}_k = (V, E, \tau)$ depicted in Figure 5 with $|V| = 4k-1$. This graph admits a cop-winning strategy if the cop picks the highlighted vertex with index 0 as her start vertex. The vertices are indexed by numbers preceded with $\circlearrowright$ indicating their clockwise distance to the start vertex of the cop, and with numbers preceded with $\circlearrowleft$ indicating their counterclockwise distance. Let the cop pick vertex 0. We consider the antipodal vertices $\circlearrowright\ 2k-1$ and $\circlearrowleft\ 2k-1$ as potential start vertices of the robber. We show that if the robber picks vertex $\circlearrowright\ 2k-1$, then the cop has a winning strategy by continuously running clockwise, starting in time step zero, and if the robber picks vertex $\circlearrowleft\ 2k-1$, the same applies running counterclockwise. Note that these two positions represent extrema, and being able to catch the robber at these vertices implies being able to catch him at all vertices in the graph. Table 1 shows the positions of the cop and robber for these strategies for $k \geq 4$. For each time step, the positions after both players have moved are depicted; s is the start configuration. We abbreviate consecutive 1-edges and only depict the time steps and positions when one of the players reaches a non-trivial edge. For the cases of $k = 2$ and $k = 3$ the cop catches the robber earlier than depicted in Table 1, namely in time step $t = 6$ clockwise and $t = 8$ counterclockwise for $k = 2$ and in time step $t = 6$ clockwise and $t = 9$ counterclockwise for $k = 3$ if the robber chooses the corresponding antipodal start vertices.

Details on the cases $k = 2$ and $k = 3$, as well as a detailed illustration of the chase for $k = 4$, can be found in the appendix. $\square$

For the case that $\ell = 1$, i.e., when $\text{lcm}(L_{\mathcal{G}}) \geq 2 \cdot \max L_{\mathcal{G}}$, we slightly adapt the family of graphs depicted in Figure 5. Note that for $\max L_{\mathcal{G}} = 2$ there is no edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\text{lcm}(L_{\mathcal{G}}) > \max L_{\mathcal{G}} = 2$.

Lemma 27. *For every $k \geq 3$ with $k \neq 2^m$ for all $m \in \mathbb{N}$, there exists an edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\text{lcm}(L_{\mathcal{G}}) = 2 \cdot \max L_{\mathcal{G}} = 2 \cdot k$, and $n = 2 \cdot 2k - 1$ vertices that is cop-winning.*

Proof. Note that we have $\ell = 1$. We introduce an artificial edge label in the edge-periodic cycle in Figure 5, such that $\text{lcm}(L_{\mathcal{G}})$ is exactly $2k$. This edge will not affect the run of the cop. Its purpose is to introduce a factor 2 in the number of vertices, compensating for the missing factor 2 due to $\ell = 1$. Therefore, note that the edge $e_{1,2}$ connecting vertex $\circlearrowright 1$ and $\circlearrowright 2$ is taken by the cop only once, in the clockwise run in time step 1 and in the counterclockwise run in time step $4k - 3$. Hence, the cop only crosses the edge in an odd time step. We can write k as $k = 2^i \cdot j$ where j is an odd number with $j > 1$ since $k \neq 2^m$. Then, introducing a string $\tau(e_{1,2}) = 01^{2^{i+1}-1}$ of length 2^{i+1} yields a least common multiple of $\text{lcm}(L_{\mathcal{G}}) = 2^{i+1} \cdot j = 2 \cdot k$. $\square$

In the case of $\max L_{\mathcal{G}} = k = 2^m$ for some $m \in \mathbb{N}$, it holds that for the smallest possible value of $\text{lcm}(L_{\mathcal{G}})$ with $\text{lcm}(L_{\mathcal{G}}) > \max L_{\mathcal{G}}$, we have $\text{lcm}(L_{\mathcal{G}}) \geq 3 \cdot \max L_{\mathcal{G}}$. Hence, in these cases we need a separate family of graphs.

Lemma 28. *For every $k = 2^m$ with $m \geq 2$, there exists an edge-periodic cycle $\mathcal{G} = (V, E, \tau)$ with $\text{lcm}(L_{\mathcal{G}}) = 3 \cdot \max L_{\mathcal{G}} = 3 \cdot k$, and $n = 6 \cdot k - 1$ vertices that is cop-winning.*

Proof. Consider the edge-periodic cycle $\mathcal{G}_k = (V, E, \tau)$ depicted in Figure 6 with $|V| = 6k - 1$. This graph admits a cop-winning strategy if the cop picks the vertex with index 0 as her start vertex. The vertices are indexed by numbers indicating their clockwise distance from the start vertex of the cop. Let the cop pick vertex 0. We show that if the robber picks vertex $3k-1$, then the cop has a winning strategy by continuously running clockwise, starting in

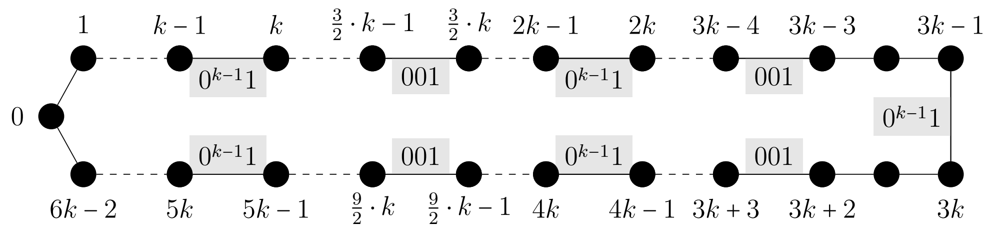


Figure 6: Cycle with $6\cdot k-1$ vertices and $\mathrm{lcm}(L_{\mathcal{G}})=3k$ with a cop-winning strategy from the start vertex 0 where $k=2^m$ and $m\geq 2$. Edges not drawn (depicted by dots) or edges without an explicit label are 1-edges; for all other edges, $\tau(e)$ is shown explicitly (with gray background).

time step zero. Since for each j, starting from vertex 0, the label of the j-th edge clockwise is equal to the label of the j-th edge counterclockwise, the same applies running counterclockwise if the robber picks vertex $3k$. Note that these two positions represent extrema, and being able to catch the robber at these vertices implies being able to catch him at all vertices in the graph. Suppose that the robber picks vertex $3k-1$. Since $k=2^m$ for some $m\geq 2$, $\frac{3}{2}\cdot k$ and $\frac{9}{2}\cdot k$ are divisible by 3. Hence for each $j\in[0,6k-3]$, the cop can traverse the edge $\{j,j+1\}$ in time step j and, thus, reach the vertex $5k-1$ in time step $5k-2$. We show that, starting from vertex $3k-1$ and running clockwise, the robber cannot reach vertex $5k$ prior to time step $5k-1$. This then implies that the cop catches the robber after at most $5k-2$ time steps. Note that the first time the robber can traverse the edge $\{3k-1,3k\}$ is at time step $k-1$. Hence, the robber cannot reach the vertex $3k+2$ prior to time step $k+1$. Since k is not divisible by 3, the robber cannot traverse the edge $\{3k+2,3k+3\}$ in time step $k+2$. Thus, the robber cannot reach the vertex $4k-1$ prior to time step $2k$ and, consequently, he cannot traverse the edge $\{4k-1,4k\}$ prior to time step $3k-1$. Hence, the robber cannot reach the vertex $\frac{9}{2}k-1$ prior to time step $\frac{7}{2}k-2$. Since k is not divisible by 3, the robber cannot traverse the edge $\{\frac{9}{2}k-1,\frac{9}{2}k\}$ in time step $\frac{7}{2}k-1$. Thus, the robber cannot reach the vertex $5k-1$ prior to time step $4k$ and, consequently, he cannot traverse the edge $\{5k-1,5k\}$ prior to time step $5k-1$. Hence, the statement holds.

We explicitly give the edge-periodic cycle for $k=4$ as an example. The edge-periodic cycle is depicted in Figure 7 and the chase is described in Table 2. $\square$

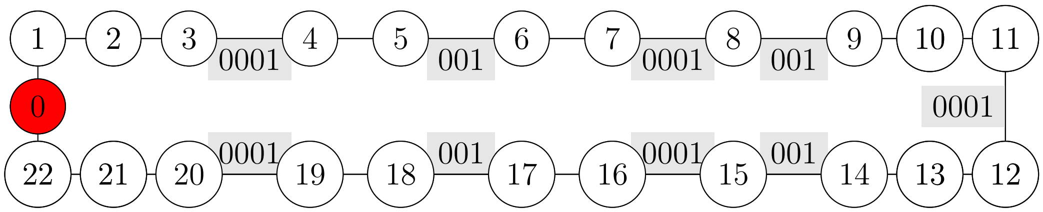


Figure 7: Cycle with $23 = 6k - 1$ vertices and $\mathrm{lcm}(L_{\mathcal{G}}) = 12 = 3k$ with a cop-winning strategy from the start vertex 0 where $k = 4$. Edges without an explicit label are 1-edges.

time step	pos. cop	pos. robber	time step	pos. cop	pos. robber
s	0	11	s	0	12
0	1	11	0	22	12
1	2	11	1	21	12
2	3	11	2	20	12
3	4	12	3	19	11
4	5	13	4	18	10
5	6	14	5	17	9
6	7	14	6	16	9
7	8	14	7	15	9
8	9	15	8	14	8
9	10	15	9	13	8
10	11	15	10	12	8
11	12	16	11	11	7
12	13	17	12	10	6
13	14	17	13	9	6
14	15	18	14	8	5
15	16	19	15	7	4
16	17	19	16	6	4
17	18	19	17	5	4
18	19		18	4	

Table 2: Time steps with corresponding positions of cop and robber in the edge-periodic cycle depicted in Figure 7. All positions are *after* moving in this time step. The time step s denotes the start configuration. Recall that the cop moves first. Icon: Flaticon.com

6. Discussion

While we have shown that the Periodic Cop & Robber problem is contained in PSPACE and NP-hard even for edge-periodic cycles, an exact characterization of the complexity of the problem remains elusive. In particular, it would be very interesting to determine whether Periodic Cop & Robber is contained in NP. This question remains open even for the special case of (directed or undirected) edge-periodic cycles. On the one hand, our representation of edge-periodic graphs is quite compact: A natural representation of a cop-winning strategy might be of exponential length in the input size, since the periodicity of the whole graph is the least common multiple of the periodicities of all edges. This prevents the use of a simple guess & check approach to show membership in NP. On the other hand, the representation is still exponentially larger than the representation by on-line programs used in [22] where PSPACE-completeness for the reachability problem on a related but different class of periodic TVGs was obtained.

If we consider *directed* edge-periodic cycles, then determining whether the given cycle is cop-winning boils down to deterministically simulating the chase starting from a (guessed) cop vertex and time step, as the optimal strategies for the cop and robber are both to keep running whenever possible (without the robber bumping into the cop). For the robber, the optimal start vertex is directly behind the cop. Since $\mathrm{lcm}(L_{\mathcal{G}})$ can be exponentially large in the size of $\mathcal{G}$, the only known upper bound on the number of time steps in the simulation of the chase starting in some time step t is exponential in the size of $\mathcal{G}$, while the chase itself does not present any complexity. The simulation could even be performed by a log-space Turing-Machine being equipped with a clock that allows for modulo queries of logarithmic size. To better understand the precise complexity of Periodic Cop & Robber on directed edge-periodic cycles, the theoretical analysis of potential families of cycles with shortest cop-winning strategies of exponential length would be of great interest and might indicate the necessity for a new complexity class consisting of simple simulation problems with exponential time duration.

Finally, it would also be interesting to find other classes of edge-periodic graphs that can be shown to be robber-winning if the number of vertices is larger than some constant factor times the global period. For example, edge-periodic cycles with a small number of chords could be studied from this perspective.

7. Rolf Niedermeier's Influence

Rolf Niedermeier was one of the leading researchers in the study of algorithms for temporal graphs. The work done by him and his group at TU Berlin, in particular with respect to the parameterized complexity of problems on temporal graphs, has inspired and influenced our work on this topic. Furthermore, Rolf also had direct or indirect personal impact on several of the authors. Thomas Erlebach first met Rolf at TU München in 1993/94, when Rolf was Klaus-Jörn Lange's PhD student and co-supervised Thomas's Master's thesis. Rolf's positive, friendly attitude and enthusiasm for research was inspiring. Over the years, Thomas also met Rolf numerous times at workshops and conferences, has visited him in Tübingen and Berlin, and considered him a good friend. Nils Morawietz's PhD supervisor (Christian Komusiewicz) has himself been supervised by Rolf. Jakob T. Spooner had his first experience of a Dagstuhl seminar (albeit online due to Covid restrictions) when he attended the seminar on Temporal Graphs that was co-organized by Rolf in April 2021. Petra Wolf's PhD supervisor (Henning Fernau) worked together with Rolf when Henning was a post-doc in Tübingen with Klaus-Jörn Lange, who had been Rolf's PhD supervisor and Petra Wolf's Master's thesis supervisor.

References

[1] T. Erlebach, J. T. Spooner, A game of cops and robbers on graphs with periodic edge-connectivity, in: Proceedings of 46th International Conference on Current Trends in Theory and Practice of Informatics, Vol. 12011 of Lecture Notes in Computer Science, Springer, 2020, pp. 64–75. doi:10.1007/978-3-030-38919-2_6.

[2] N. Morawietz, P. Wolf, A timecop's chase around the table, in: Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021), Vol. 202 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 77:1–77:18. doi:10.4230/LIPIcs.MFCS.2021.77.

[3] T. H. Chung, G. A. Hollinger, V. Isler, Search and pursuit-evasion in mobile robotics, Autonomous Robots 31 (4) (2011) 299–316. doi:10.1007/s10514-011-9241-4.

[4] P. Seymour, R. Thomas, Graph searching and a min-max theorem for tree-width, Journal of Combinatorial Theory, Series B 58 (1) (1993) 22–33. doi:10.1006/jctb.1993.1027.

[5] A. Quilliot, Jeux et pointes fixes sur les graphes, Ph.D. thesis, Ph. D. Dissertation, Université de Paris VI (1978).

[6] R. J. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, Discret. Math. 43 (2-3) (1983) 235–239. doi:10.1016/0012-365X(83)90160-7.

[7] M. Aigner, M. Fromme, A game of cops and robbers, Discrete Applied Mathematics 8 (1) (1984) 1 – 12. doi:10.1016/0166-218X(84)90073-8.

[8] A. Bonato, R. Nowakowski, The Game of Cops and Robbers on Graphs, Vol. 61 of Student Mathematical Library, American Mathematical Society, 2011. doi:10.1090/stml/061.

[9] N. E. Clarke, G. MacGillivray, Characterizations of k-copwin graphs, Discret. Math. 312 (8) (2012) 1421–1425. doi:10.1016/j.disc.2012.01.002.

[10] A. Casteigts, P. Flocchini, W. Quattrociocchi, N. Santoro, Time-varying graphs and dynamic networks, Int. J. Parallel Emergent Distributed Syst. 27 (5) (2012) 387–408. doi:10.1080/17445760.2012.668546.

[11] O. Michail, P. G. Spirakis, Elements of the theory of dynamic networks, Communications of the ACM 61 (2) (2018) 72–72. doi:10.1145/3156693.

[12] N. Ganguly, A. Deutsch, A. Mukherjee, Dynamics on and of complex networks, Applications to Biology, Computer Science, and the Social Sciences (2009).

[13] P. Holme, Modern temporal network theory: a colloquium, The European Physical Journal B 88 (9) (2015) 1–30. doi:10.1140/epjb/e2015-60657-4.

[14] P. Holme, J. Saramäki, Temporal networks, Physics reports 519 (3) (2012) 97–125. doi:10.1016/j.physrep.2012.03.001.

[15] Z. Zhang, Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges, IEEE Commun. Surv. Tutorials 8 (1-4) (2006) 24–37. doi:10.1109/COMST.2006.323440.

[16] B. Ding, J. X. Yu, L. Qin, Finding time-dependent shortest paths over large graphs, in: Proceedings of the 11th International Conference on Extending Database Technology, Vol. 261 of ACM International Conference Proceeding Series, ACM, 2008, pp. 205–216. doi:10.1145/1353343.1353371.

[17] N. Boehmer, V. Froese, J. Henkel, Y. Lasars, R. Niedermeier, M. Renken, Two influence maximization games on graphs made temporal, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, ijcai.org, 2021, pp. 45–51. doi:10.24963/ijcai.2021/7.

[18] E. Arrighi, N. Grüttemeier, N. Morawietz, F. Sommer, P. Wolf, Multi-parameter analysis of finding minors and subgraphs in edge-periodic temporal graphs, in: SOFSEM 2023: Theory and Practice of Computer Science - 48th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2023, Nový Smokovec, Slovakia, January 15-18, 2023, Proceedings, Vol. 13878 of Lecture Notes in Computer Science, Springer, 2023, pp. 283–297. doi:10.1007/978-3-031-23101-8_19.

[19] S. Bhadra, A. Ferreira, Complexity of connected components in evolving graphs and the computation of multicast trees in dynamic networks, in: Proceedings of the 2nd International Conference on Ad-Hoc, Mobile, and Wireless Networks, Vol. 2865 of Lecture Notes in Computer Science, Springer, 2003, pp. 259–270. doi:10.1007/978-3-540-39611-6_23.

[20] O. Michail, P. G. Spirakis, Traveling salesman problems in temporal graphs, Theor. Comput. Sci. 634 (2016) 1–23. doi:10.1016/j.tcs.2016.04.006.

[21] K. Wehmuth, A. Ziviani, E. Fleury, A unifying model for representing time-varying graphs, in: 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, Campus des Cordeliers,

Paris, France, October 19-21, 2015, IEEE, 2015, pp. 1–10. doi:10.1109/DSAA.2015.7344810.

[22] K. Sutner, W. Maass, Motion planning among time dependent obstacles, Acta Informatica 26 (1-2) (1988) 93–122. doi:10.1007/BF02915447.

[23] S. Balev, J. L. J. Laredo, I. Lamprou, Y. Pigné, E. Sanlaville, Cops and robbers on dynamic graphs: Offline and online case, in: Structural Information and Communication Complexity - 27th International Colloquium, SIROCCO 2020, Paderborn, Germany, June 29 - July 1, 2020, Proceedings, Vol. 12156 of Lecture Notes in Computer Science, Springer, 2020, pp. 203–219. doi:10.1007/978-3-030-54921-3_12.

[24] N. Morawietz, C. Rehs, M. Weller, A timecop's work is harder than you think, in: Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020), Vol. 170 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 71:1–71:14. doi:10.4230/LIPIcs.MFCS.2020.71.

[25] J. De Carufel, P. Flocchini, N. Santoro, F. Simard, Cops & robber on periodic temporal graphs: Characterization and improved bounds, in: S. Rajsbaum, A. Balliu, J. J. Daymude, D. Olivetti (Eds.), Structural Information and Communication Complexity - 30th International Colloquium, SIROCCO 2023, Alcalá de Henares, Spain, June 6-9, 2023, Proceedings, Vol. 13892 of Lecture Notes in Computer Science, Springer, 2023, pp. 386–405. doi:10.1007/978-3-031-32733-9_17.

[26] J. De Carufel, P. Flocchini, N. Santoro, F. Simard, Copnumbers of periodic graphs, CoRR abs/2310.13616 (2023). arXiv:2310.13616, doi:10.48550/ARXIV.2310.13616.

[27] G. Hahn, G. MacGillivray, A note on k-cop, l-robber games on graphs, Discrete Mathematics 306 (19) (2006) 2492–2497, creation and Recreation: A Tribute to the Memory of Claude Berge. doi:10.1016/j.disc.2005.12.038.

[28] A. Bonato, G. MacGillivray, Characterizations and algorithms for generalized cops and robbers games, Contributions Discret. Math. 12 (1) (2017). doi:10.11575/cdm.v12i1.62435.

[29] A. Berarducci, B. Intrigila, On the cop number of a graph, Advances in Applied Mathematics 14 (4) (1993) 389 – 403. doi:10.1006/aama.1993.1019.

[30] A. Kehagias, G. Konstantinidis, Cops and robbers, game theory and Zermelo's early results, CoRR abs/1407.1647 (2014). arXiv:1407.1647.

[31] E. Grädel, W. Thomas, T. Wilke (Eds.), Automata Logics, and Infinite Games: A Guide to Current Research, Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[32] D. Berwanger, Graph games with perfect information, preprint (2013).

[33] A. Kehagias, D. Mitsche, P. Pralat, The role of visibility in pursuit/evasion games, Robotics 4 (2014) 371–399. doi:10.3390/robotics3040371.

[34] V. Patsko, S. Kumkov, V. Turova, Pursuit-evasion games, in: T. Basar, G. Zaccour (Eds.), Handbook of Dynamic Game Theory, Springer International Publishing, 2017, pp. 1–87. doi:10.1007/978-3-319-27335-8-30-1.

[35] L. de Alfaro, T. A. Henzinger, O. Kupferman, Concurrent reachability games, Theor. Comput. Sci. 386 (3) (2007) 188–217. doi:10.1016/j.tcs.2007.07.008.

[36] N. Immerman, Number of quantifiers is better than number of tape cells, J. Comput. Syst. Sci. 22 (3) (1981) 384–406. doi:10.1016/0022-0000(81)90039-8.

Appendix A. Details for small k in the proof of Lemma 26

In this appendix, we provide the details on the cases $k = 2$ and $k = 3$ of the construction presented in the proof of Lemma 26. We explicitly give the edge-periodic cycles for $k = 2$, $k = 3$, and $k = 4$. For $k = 2$ and $k = 3$ the chase of the cop will be shorter than described in Table 1 and for $k \geq 4$ the chase will be exactly as described in general in Table 1. The edge-periodic cycle for $k = 2$ is depicted in Figure A.8 and the chase is described in Table A.3. For $k = 3$ the edge-periodic cycle is depicted in Figure A.9 and the chase is described in Table A.4. Finally, for $k = 4$, the edge-periodic cycle is depicted in Figure A.10 and the explicit chase is described in Table A.5. Note that Table A.5 is identical to Table 1 if we set $k = 4$ in Table 1.

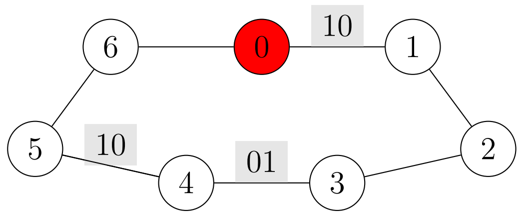
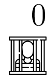


Figure A.8: Edge-periodic cycle for the case $k = 2$ in Lemma 26 with $4 \cdot k - 1 = 7$ vertices and $\mathrm{lcm}(L_{\mathcal{G}}) = 2$ with a cop-winning strategy from the start vertex marked in red. Edges without edge label are 1-edges; for all other edges, $\tau(e)$ is shown explicitly (with gray background).

time step	pos. cop	pos. robber
s	0	3
0	1	3
1	2	4
2	3	5
3	4	6
4	5	0
5	6	0
6	0	

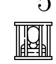
time step	pos. cop	pos. robber
s	0	4
0	6	4
1	5	3
2	4	2
3	3	1
4	2	0
5	1	6
6	0	5
7	6	5
8	5	

Table A.3: Time steps with corresponding positions of cop and robber in the edge-periodic cycle depicted in Figure A.8. All positions are *after* moving in this time step. The time step s denotes the start configuration. Recall that the cop moves first. Icon: Flaticon.com

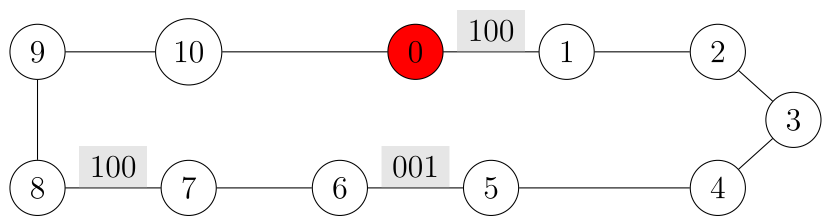


Figure A.9: Edge-periodic cycle for the case $k = 3$ in Lemma 26 with $4 \cdot k - 1 = 11$ vertices and $\mathrm{lcm}(L_{\mathcal{G}}) = 3$ with a cop-winning strategy from the start vertex marked in red. Edges without edge label are 1-edges; for all other edges, $\tau(e)$ is shown explicitly (with gray background).

time step	pos. cop	pos. robber
s	0	5
0	1	5
1	2	5
2	3	6
3	4	7
4	5	7
5	6	7
6	7	

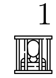
time step	pos. cop	pos. robber
s	0	6
0	10	6
1	9	6
2	8	5
3	7	4
4	6	3
5	5	2
6	4	1
7	3	1
8	2	1
9	1	

Table A.4: Time steps with corresponding positions of cop and robber in the edge-periodic cycle depicted in Figure A.9. All positions are *after* moving in this time step. The time step s denotes the start configuration. Recall that the cop moves first. Icon: Flaticon.com

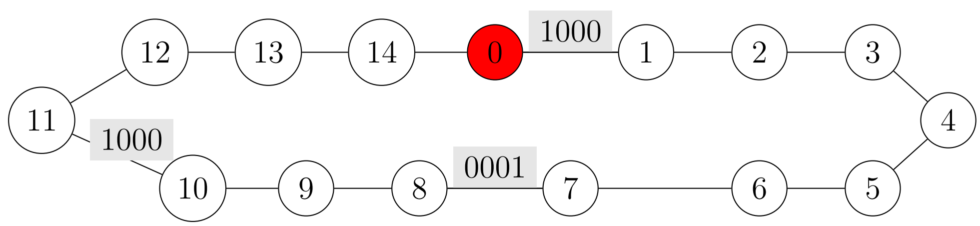
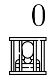


Figure A.10: Edge-periodic cycle for the case $k = 4$ in Lemma 26 with $4 \cdot k - 1 = 15$ vertices and $\mathrm{lcm}(L_{\mathcal{G}}) = 4$ with a cop-winning strategy from the start vertex marked in red. Edges without edge label are constant 1-edges; for all other edges, $\tau(e)$ is shown explicitly (with gray background).

time step	pos. cop	pos. robber
s	0	7
0	1	7
1	2	7
2	3	7
3	4	8
4	5	9
5	6	10
6	7	10
7	8	10
8	9	11
9	10	12
10	10	13
11	10	14
12	11	0
13	12	0
14	13	0
15	14	0
16	0	

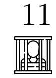
time step	pos. cop	pos. robber
s	0	8
0	14	8
1	13	8
2	12	8
3	11	7
4	10	6
5	9	5
6	8	4
7	7	3
8	6	2
9	5	1
10	4	1
11	3	1
12	2	0
13	1	14
14	1	13
15	1	12
16	0	11
17	14	11
18	13	11
19	12	11
20	11	

Table A.5: Time steps with corresponding positions of cop and robber in the edge-periodic cycle depicted in Figure A.10. Note that the positions of the cop and robber are as described in Table 1 for the general case of $k \geq 4$. All positions are *after* moving in this time step. The time step s denotes the start configuration. Recall that the cop moves first. Icon: Flaticon.com