A Timecop's Work Is Harder Than You Think

Nils Morawietz

Fachbereich Mathematik und Informatik, Philipps-Universität Marburg, Germany
morawietz@informatik.uni-marburg.de

Carolin Rehs

Institute of Computer Science, Heinrich Heine Universität, Düsseldorf, Germany
carolin.rehs@hhu.de

Mathias Weller

CNRS, LIGM, Université Gustave Eiffel, Marne-la-Vallée, France
mathias.weller@u-pem.fr

Abstract

We consider the (parameterized) complexity of a cop and robber game on periodic, temporal graphs and a problem on periodic sequences to which these games relate intimately. In particular, we show that it is NP-hard to decide (a) whether there is some common index at which all given periodic, binary sequences are 0, and (b) whether a single cop can catch a single robber on an edge-periodic temporal graph. We further present results for various parameterizations of both problems and show that hardness not only applies in general, but also for highly limited instances. As one main result we show that even if the graph has a size-2 vertex cover and is acyclic in each time step, the cop and robber game on periodic, temporal graphs is NP-hard and W[1]-hard when parameterized by the size of the underlying input graph.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Dynamic graph algorithms

Keywords and phrases edge-periodic temporal graphs, cops and robbers, tally-intersection, congruence satisfyability

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.71

Acknowledgements We thank our anonymous reviewers for their various helpful suggestions, in particular the connection to Tally-DFAs.

1 Introduction

A cops and robbers game in graph theory is a pursuit-evasion game with two teams of players, the cops and the robbers, moving from vertex to vertex along the edges of a graph. The cops try to move onto the vertices where the robbers are positioned, thereby "catching" them, while the robbers try to evade such capture. Cops and robbers games with varying rules have been popular in graph theory as, on the one hand, they model a range of applications of pursuit-evasion games (see, for example [9]), on the other hand, they relate to useful graph parameters, such as path-width and tree-width [7, 27], directed path-width [4], directed tree-width [21], DAG-width [5] or Kelly-width [19] (for surveys, see Amiri et al. [3], Fomin and Thiliko [16] and Nisse [23], who considers the problem of "cleaning" a graph with mobile agents which is equivalent to a cops and robber game in which the robbers location is unknown to the cops). The special case of one robber trying to evade one cop has been fully characterized and is shown to be solvable in polynomial time by Nowakowski and Winkler [25] in the 80s using the concept of "dismantlable" graphs (see also [8]). With the recent rise of dynamic graphs ("temporal graphs"), the cop and robber game regains traction. In particular, the one-cop one-robber version was recently reconsidered on "edge-periodic temporal graphs" by Erlebach and Spooner [13]. Temporal graphs exist in

45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020).
Editors: Javier Esparza and Daniel Král'; Article No. 71; pp. 71:1–71:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

so-called time-steps, in which every edge may or may not appear. To this end, each edge is labeled with a set of integers corresponding to the time-steps this edge appears in [22]. Such graphs have recently received increased attention in the graph-theory community due to their versatility in modeling relations that evolve in time (see [1, 2, 12, 15]). Considering time as infinite yields inputs of infinite size, unless we assume that the relation repeats periodically. This is modeled by "edge-periodic" temporal graphs, where each edge additionally has a number indicating the length of the period after which its occurrences repeat. This setting can also be modeled by assigning each edge a sequence $x \in \{0,1\}^*$, where x has a 1 at position p if and only if the edge exists in the time-steps p, $p+|x|$, $p+2|x|$, ...

In their paper, Erlebach and Spooner [13] consider the one-cop one-robber variant on edge-periodic temporal graphs (called Cop-Win Periodic Cop and Robber, for short PCnR), characterizing strategies for both cops and robbers on edge-periodic cycles and showing how to solve the one-cop one-robber variant on any edge-periodic temporal graph in $\mathcal{O}(\mathrm{lcm}(\mathcal{L})n^3)$, where $\mathcal{L}$ is the set of lengths of sequences occurring in the input and n is the number of vertices in the underlying static graph. Prominently, they leave open (and state as explicit open question) whether deciding the winner of their cop and robber game is NP-hard. We resolve this open question by reducing an arithmetic problem of independent interest to PCnR. Indeed, introducing this periodic temporal concept makes the problem much harder than one would expect from the results on static graphs [25, 8]. To introduce the arithmetic problem from which we derive hardness of PCnR, let $x[j]$ denote the j^{th} symbol (starting with 0) of a sequence $x \in \{0,1\}^*$ and let us abbreviate $x[j]^\circ := x[j \bmod |x|]$.

Periodic Character Alignment (PCA)
Input: A finite set $X \subseteq \{0,1\}^*$ and $k \in \mathbb{N}$.
Question: Is there some $i \in \mathbb{N}$ such that $|\{x \in X \mid x[i]^\circ = 0\}| \geq k$?

The special case where $k = |X|$ is called Periodic Full Character Alignment.

Periodic Full Character Alignment (PFA)
Input: A finite set $X \subseteq \{0,1\}^*$.
Question: Is there some $i \in \mathbb{N}$ such that, for all $x \in X$, we have $x[i]^\circ = 0$?

We assume that $|x| > 1$ for all $x \in X$, since otherwise the instance is either a trivial no-instance (for PFA) or the sequences of length one can be removed. Likewise, we can assume for PFA that no two sequences, in X have the same length since we can replace two sequences $x, y \in X$ of same length with the sequence resulting from the "bitwise-or" of x and y. Moreover, we abbreviate $\sum_{x \in X} |x| =: n$.

Interestingly, PFA and PCA are closely related to the well-known Intersection problem for deterministic finite automata over the (unary) alphabet $\{1\}$ (called *Tally-DFAs*, see [18] for a survey). Being deterministic and over a unary alphabet, each state of a Tally-DFA can only transition into at most one other state. Thus, any q-state Tally-DFA A can be represented as a rooted, directed graph G_A with q vertices of out-degree one, some of which are marked as accepting states. Thus, A consists of a path plus one arc from the last vertex of the path to some other vertex of the path. Then, a word of length more than q is accepted by A if and only if it consists of a prefix 1^q followed by any number of repetitions of 1^{c_A} (where c_A is the number of vertices spanned by the last arc) followed by a suffix 1^{s_A} (depending on which states on the path are accepting). Now, the Tally-Intersection problem asks whether some word is accepted by each of t given Tally-DFAs A_i, that is, $\bigcap_i L(A_i) \neq \varnothing$. Clearly, Periodic Full Character Alignment is a special case of Tally-Intersection and they coincide if $c_{A_i} = q_i$ (that is, G_{A_i} is a cycle) for all i. We can achieve that $c_{A_i} = q_i$ for all i by simulating each automaton for $q_{\max} := \max_i q_i$ steps, checking whether the automata accept a common word of length at most $q_{\max}$.

▶ **Corollary 1.** *Periodic Full Character Alignment is polynomial-time equivalent to Tally-Intersection.*

Regarding previous work, Fernau and Krebs [14] show that Tally-Intersection for k automata, each with at most q states, cannot be solved in $2^{o(\min\{k\sqrt{\log q}, \sqrt{q}\})}n^{\mathcal{O}(1)}$ time (unless the Exponential Time Hypothesis fails), but can be solved in $q^k n^{\mathcal{O}(1)}$ and $2.9^q n^{\mathcal{O}(1)}$ time.

The main relevance of PFA and PCA for this study are their strong relation to edge-periodic temporal graphs: PCA is equivalent to the question whether there is a time-step in which a given edge-periodic temporal graph misses at least k edges and consequently PFA is equivalent to asking whether there is a time-step in which the graph is edgeless. We show that PFA is NP-complete (implying that PCA is) and reduce it to PCnR on temporal graphs that are acyclic in all time steps. Note that due to 1, the NP-hardness of Periodic Full Character Alignment is already clear but we present a new reduction from Multi-Colored Clique to Periodic Full Character Alignment (and, thus, to Tally-Intersection) which gives new insights into the parameterized complexity of both problems. We further prove that this hardness holds for the modification of PCnR where the robber moves first. We initiate the study of the parameterized complexity of PCA, PFA, and PCnR, proving positive and negative results for various parameterizations. The parameterized landscape for these problems seems surprisingly desolate, with W[1]-hardness for most of the (single) parameters, culminating in the W[1]-hardness of PCnR when parameterized by the size of the underlying input graph even for extremely restricted instances. A corollary that may be of independent interest is that Multi-Colored Clique is W[1] hard when parameterized by the size k of the sought clique, even if the induced subgraph between any pair of distinct color classes is the union of vertex-disjoint bicliques.

Due to lack of space, some proofs are deferred to an appendix (statements marked as corollaries or observations do not have separate proofs).

2 Preliminaries

For $a, b \in \mathbb{N}$, we abbreviate $\{a, a+1, a+2, \dots, b-1, b\} =: [a, b]$. We assume the reader to be familiar with parameterized complexity basics (see [11, 10]) and we refer to [28] for an overview of graph parameters considered in this work.

2.1 Sequences and Temporal Graphs

Let $w \in \{0,1\}^*$ be a sequence and let $j \in \mathbb{N}$. Then the length of w is denoted by $|w|$ and the j^{th} symbol (starting with 0) of w is denoted by $w[j]$. For a set X of sequences, we let $\mathcal{L}(X) := \{|x| \mid x \in X\}$ denote the set of lengths of sequences occurring in X and we let $L(X) := \max(\mathcal{L}(X))$ denote the maximum length of a sequence in X. If X is clear from the context, we may only write $\mathcal{L}$ and L. We often consider the infinite, periodic sequence resulting from repeating a sequence w indefinitely. In this case, we write $w[j]^\circ := w[j \bmod |w|]$. For two periodic sequences w and q, we define $w \,\&\, q$ as the "bitwise-and" of w and q, that is, $w \,\&\, q$ has length $\operatorname{lcm}(|w|, |q|)$ and $(w \,\&\, q)[i]^\circ = w[i]^\circ \cdot q[i]^\circ$. A *run* of a sequence w is a consecutive part of w consisting only of 1s. Runs can be represented by the pair of their first and last index in w. We let $Bl_1(w)$ denote the set of runs in the sequence w, encoded as such pairs. For $c \in \mathbb{N}$, we define the *c-fold blow up* of w as the sequence $c \times w$ with $(c \times w)[i] = w[\lfloor i/c \rfloor]$ for all $i \in [0, c|w| - 1]$. Moreover, for a set of sequences X we denote with $c \times X := \{c \times w \mid w \in X\}$ the *c-fold blow up* of X.

An (edge-) periodic (temporal) graph $G^\tau = (V, E, \tau)$ (see also [13]) 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 sequence $\tau(e) \in \{0, 1\}^*$ such that e exists in a time step $t \geq 0$ if and only if $\tau(e)[t]^\circ = 1$. 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 $\tau(e)[t_e] = 1$.

2.2 Number Theory

We assume that the reader is familiar with the concepts of modulo arithmetic, the greatest common divisor (gcd), the least common multiple (lcm) and the extended Euclidean algorithm.

Note that congruence modulo any integer q is an equivalence relation. In particular, given $a \equiv b \pmod q$ and $a \equiv c \pmod q$, we also have $b \equiv c \pmod q$. Further, the modulo operation is linear, that is, for integers $a, b, c \in \mathbb{Z}$, we have $ac \bmod bc = c(a \bmod b)$. The Chinese Remainder Theorem (see for example [20]) roughly states that the congruence system $x \equiv a_i \pmod{n_i}$ with $n_i \in \mathbb{N}$ pairwise coprime and $a_i \in \mathbb{Z}$ always has a solution $x \in \mathbb{Z}$ and all solutions are congruent modulo $N := \prod_i n_i$. In this work, we use a slightly more general version of the Chinese Remainder Theorem, proved as Lemma 4.

▶ **Lemma 2.** *Let $x, y, p_0, p_1, \ldots \in \mathbb{Z}$. Then, $x \equiv y \pmod{\operatorname{lcm}(p_0, p_1, \ldots)}$ if and only if $x \equiv y \pmod{p_i}$ for all i.*

Proof. $(\Rightarrow)$ Since p_i divides $\operatorname{lcm}(p_0, p_1, \ldots)$ which in turn divides $x - y$, we know that p_i divides $x - y$. $(\Leftarrow)$ $x - y$ is clearly a common multiple of all p_i, so it is divided by their least common multiple. ◀

▶ **Lemma 3.** *Let $x, y, a, b \in \mathbb{Z}$. Then, $ax \equiv by \pmod{\gcd(a, b)}$.*

Proof. Since $\gcd(a, b)$ divides a and b, it divides ax and by and thus $ax - by$. ◀

▶ **Lemma 4.** *Let $m \geq 1$, let $a_0, a_1, \ldots, a_m \in \mathbb{Z}$, let $p_0, p_1, \ldots, p_m \in \mathbb{N}$. Then there is some $j \in \mathbb{N}$ with $j \equiv a_i \pmod{p_i}$ for all $i \in [0, m]$ if and only if $a_i \equiv a_{i'} \pmod{\gcd(p_i, p_{i'})}$ for all $i, i' \in [0, m], i < i'$.*

Proof. The proof is by induction on m starting with $m = 1$.

$(\Leftarrow)$ For the induction base, let $u_0, u_1 \in \mathbb{Z}$ such that $\gcd(p_0, p_1) = u_0 p_0 + u_1 p_1$ be the Bézout coefficients [6] of p_0 and p_1. Let $z \in \mathbb{Z}$ such that $a_0 = z(u_0 p_0 + u_1 p_1) + a_1$. Then, $a_0 - z u_0 p_0 = a_1 + z u_1 p_1 =: i$ and $a_0 - z u_0 p_0 \equiv a_0 \pmod{p_0}$ and $a_1 + z u_1 p_1 \equiv a_1 \pmod{p_1}$. For the induction step, we replace a_m and a_{m-1} by some j' with $j' \equiv a_m \pmod{p_m}$ and $j' \equiv a_{m-1} \pmod{p_{m-1}}$, which exists by induction hypothesis, and we replace p_m and p_{m-1} by $\ell := \operatorname{lcm}(p_m, p_{m-1})$. Then, for all $i \in [0, m-1]$, as $a_m \equiv a_i \pmod{\gcd(p_m, p_i)}$ and $a_{m-1} \equiv a_i \pmod{\gcd(p_{m-1}, p_i)}$, Lemma 2 implies $j' \equiv a_i \pmod{\operatorname{lcm}(\gcd(p_i, p_m), \gcd(p_i, p_{m-1}))}$ which, by distributivity of gcd and lcm, implies $j' \equiv a_i \pmod{\gcd(\operatorname{lcm}(p_m, p_{m-1}), p_i)}$, that is, $j' \equiv a_i \pmod{\gcd(\ell, p_i)}$. Thus, we can apply the induction hypothesis, granting existence of $j \in \mathbb{N}$ with $j \equiv a_i \pmod{p_i}$ and $j \equiv j' \pmod{\ell}$. By Lemma 2, $j \equiv j' \pmod{p_m}$ and $j \equiv j' \pmod{p_{m-1}}$, implying $j \equiv a_m \pmod{p_m}$ and $j \equiv a_{m-1} \pmod{p_{m-1}}$.

$(\Rightarrow)$ For all $i \in [0, m]$, let $u_i \in \mathbb{Z}$ such that $u_i p_i + a_i = j$. Then, for all $i, i' \in [0, m]$, we have $a_i - a_{i'} = (j - u_i p_i) - (j - u_{i'} p_{i'}) = u_{i'} p_{i'} - u_i p_i$ and, since $u_{i'} p_{i'} - u_i p_i$ is divisible by $\gcd(p_i, p_{i'})$ (see Lemma 3), so is $a_i - a_{i'}$. ◀

To compute a solution for a system of congruences $x \equiv a_i \pmod{n_i}$, assuming $a_i < n_i$ for all i, we replace pairs of congruences by one, equivalent congruence using Lemma 2. Since this can be done using the extended Euclidean algorithm in $\mathcal{O}(\log n_i \cdot \log n_j) \subseteq \mathcal{O}(\log^2 N)$ time, the whole congruence system can be solved in time $\mathcal{O}(m \log^2 N) = \mathcal{O}(m \left(\sum_i \log n_i\right)^2)$, where m is the number of input congruences, which is polynomial in the input size,

3 Complexity of PFA and PCA

While the NP-hardness of PFA already follows from its equivalence to Tally-Intersection (see Corollary 1), we present here another reduction with implications on parameterized complexity, which can then be transferred to Cop-Win Periodic Cop and Robber. As a side note, we remark that PCA is in NP since the input sequences can easily be verified to be 0 at a given common index.

Multi-Colored Clique (MCC)
Input: An integer k and an undirected k-partite graph $G = (V_1 \uplus V_2 \uplus \ldots \uplus V_k, E)$.
Question: Is there a clique of size k in G?

▶ **Lemma 5.** *Periodic Full Character Alignment is* NP*-hard.*

Proof. Given an instance $(G = (V, E), k)$ of MCC with k-partition $(V_1, \ldots, V_k)$ of V, we construct an equivalent instance X of PFA in polynomial time such that $|X| = \binom{k}{2}$. To this end, we compute distinct prime numbers $p_1, \ldots, p_k$ in polynomial time[1] such that $|V_i| \leq p_i$ for all $i \in [1, k]$. Let $V_i := \{v_1^i, \ldots, v_{|V_i|}^i\}$. We define a sequence $w_{i,j} \in \{0, 1\}^{p_i \cdot p_j}$ for all $i, j \in [1, k], i < j$ to represent all edges in the induced subgraph $G[V_i \cup V_j]$. We set for every $r \in [0, |w_{i,j}| - 1]$,

$$w_{i,j}[r] := 0 \text{ if and only if } \{v^i_{r \bmod p_i}, v^j_{r \bmod p_j}\} \in E. \tag{1}$$

Finally, we set $X := \{w_{i,j} \mid 1 \leq i < j \leq k\}$.

▷ Claim 6. Let $i, j \in [1, k], i < j$ and let $t \in \mathbb{N}$. Then, $w_{i,j}[t]^\circ = 0$ if and only if $\{v^i_{t \bmod p_i}, v^j_{t \bmod p_j}\} \in E$.

Proof. By construction of $w_{i,j}$, we have $|w_{i,j}| \bmod p_i = |w_{i,j}| \bmod p_j = 0$ and thus $(t \bmod |w_{i,j}|) \bmod p_i = t \bmod p_i$ and $(t \bmod |w_{i,j}|) \bmod p_j = t \bmod p_j$. Then, by (1), $w_{i,j}[t \bmod |w_{i,j}|] = 0 \iff \{v^i_{t \bmod p_i}, v^j_{t \bmod p_j}\} \in E$. ◁

Next, we show that (G, k) is a yes-instance of MCC if and only if X is a yes-instance of PFA.

($\Rightarrow$) Assume that (G, k) is a yes-instance of MCC. Then, there is a clique $S \subseteq V$ such that $S \cap V_i = \{v^i_{a_i}\}$ for all $i \in [1, k]$. By construction, p_i and p_j are coprime for all $i, j \in [1, k], i < j$. Hence, the Chinese Remainder Theorem implies that the congruence system $\forall_{i \in [1,k]} t \equiv a_i \pmod{p_i}$ has a solution $t \in \mathbb{N}$. We show $w_{i,j}[t]^\circ = 0$ for all $i, j \in [1, k], i < j$. Since S is a clique, $\{v^i_{a_i}, v^j_{a_j}\} \in E$ and therefore by Claim 6, $w_{i,j}[t]^\circ = 0$.

($\Leftarrow$) Let X be a yes-instance of Periodic Full Character Alignment. Then, there is an index $t \in \mathbb{N}$ with $w[t]^\circ = 0$ for all $w \in X$. Let $a_i := t \bmod p_i$ for all $i \in [1, k]$. We set $S := \{v^i_{a_i} \mid i \in [1, k]\}$. Since $w_{i,j}[t]^\circ = 0$ for all $i, j, \in [1, k], i < j$, Claim 6 implies that $\{v^i_{a_i}, v^j_{a_j}\} \in E$. Hence, S is a clique of size k in G. ◀

Note that it is possible to "blow up" the input sequences for PCA by any factor $c \in |X|^{\mathcal{O}(1)}$ without changing the answer to the instance.

▶ **Observation 7.** *Let $c, j \in \mathbb{N}$, let $X \subseteq \{0, 1\}^*$ such that $X = c \times X'$ for some X', and let $x \in X$. Then, $x[j]^\circ = x[c \cdot \lfloor j/c \rfloor]^\circ$.*

[1] Since, by the Prime Number Theorem, the k^{th} prime p_k is in $\mathcal{O}(k \log k)$, these can be computed in $\mathcal{O}(k^2 \log k)$ time.

We show for every $c \in |X|^{\mathcal{O}(1)}$ with $c \geq 1$ that PFA (and, thus, PCA) remains NP-complete even if $X = c \times X'$ for some X'.

▶ **Proposition 8.** *Let (X', k) be an instance of PFA and let $c \in |X'|^{\mathcal{O}(1)}$. Then, (X', k) is a yes-instance of PFA if and only if (X, k) is a yes-instance of PFA where $X = c \times X'$.*

Proof. Let $S' \subseteq X'$. We show that S' is a yes-instance of PFA if and only if $S = c \times S' \subseteq X$ is a yes-instance of PFA.

($\Rightarrow$) Let $j \in \mathbb{N}$ such that for all $x' \in S'$ we have $x'[j]^\circ = 0$. Then $cj \in \mathbb{N}$ and for all $x \in S$

$$x[cj \bmod |x|] = x[cj \bmod c|x'|] = (c \times x')[c(j \bmod |x'|)] = x'[j \bmod |x'|] = 0.$$

($\Leftarrow$) Let $j \in \mathbb{N}$ such that, for all $x \in S$, it holds that $x[j]^\circ = 0$. By Observation 7, we also have $x[c \cdot \lfloor j/c \rfloor]^\circ = 0$. Then, $\lfloor j/c \rfloor \in \mathbb{N}$ and, for all $x' \in S'$,

$$\begin{aligned} x'[\lfloor j/c \rfloor \bmod |x'|] &= x'[\lfloor j/c \rfloor \bmod |x|/c] \\ &= x[c \cdot (\lfloor j/c \rfloor \bmod |x|/c)] \\ &= x[c \cdot \lfloor j/c \rfloor \bmod |x|] = x[j]^\circ = 0. \end{aligned}$$ ◀

▶ **Theorem 9.** *For every $c \in |X|^{\mathcal{O}(1)}, c \geq 1$, Periodic Full Character Alignment and Periodic Character Alignment are* NP*-complete even if $X = c \times X'$ for some X'.*

3.1 Parameterized Complexity of PFA and PCA

The quest for finding a good parameterization is one of the main challenges of every parameterized analysis. However, seeing that the input consists of a set of sequences over a size-2 alphabet, options are limited. An immediate choice might be the total number of sequences $|X|$ in the input, but a closer inspection of Lemma 5 reveals that Periodic Full Character Alignment (and, thus Periodic Character Alignment) is W[1]-hard with respect to this parameter. It is thus natural to combine $|X|$ with other structural parameters capturing the complexity of the input sequences. One way that the input sequences might be "well-behaved" is that they may all be short, but combining $|X|$ with $\max \mathcal{L}$ already bounds the input size and is therefore not interesting. Motivated by the observation that an instance of PFA is trivially yes if the lengths of sequences in X are pairwise prime, we can parameterize by the "distance to this triviality", by considering the maximum pairwise GCD d of the input lengths. Indeed, we show that even PCA is tractable for the parameter $|X| + d$. Another trivial special case is that each input sequence contains at most one letter 1 or one letter 0. This can be generalized to the condition that the input sequences have a small number of "runs" of 1s and we show that already PCA is tractable with respect to this parameter.

A different approach would be to restrict the *lengths* of the sequences instead of their number. Since, by Lemma 2, it suffices to consider solutions $j \in \mathbb{N}$ with $j < \operatorname{lcm}(\mathcal{L})$, implying that PCA can trivially be solved in $L^L \cdot n^{\mathcal{O}(1)}$ time (recall that $L = \max \mathcal{L}$). It turns out, however, that the stronger parameter $|\mathcal{L}|$ does not lead to fixed-parameter tractability.

3.1.1 Parameter $|X|$

The next corollary follows directly by the fact that Multi-Colored Clique is W[1]-hard when parameterized by k [11, 10] and $|X| = \binom{k}{2}$ in the construction of the proof of Lemma 5.

▶ **Corollary 10.** *PFA is* W[1]*-hard when parameterized by $|X|$.*

The equivalence with Tally-Intersection (see Corollary 1) implies the following.

▶ **Corollary 11.** *Determining whether k Tally-DFAs accept a common word is* W[1]*-hard wrt. k.*

To solve PFA, we can construct a graph whose vertices are all positions of sequences in X and two such positions are adjacent if and only if they will eventually coincide. Then, PFA becomes equivalent to solving Multi-Colored Clique on this particular graph. In order to check if two positions will eventually coincide, we use Lemma 4.

Now, we formally define an $|X|$-partite graph $G_X := (V, E)$ with $V := \{(x, j) \mid x \in X \wedge x[j] = 0\}$ and there is an edge between (x, j) and (y, ℓ) in G_X if and only if $x \neq y$ and $j \equiv \ell \pmod{\gcd(|x|, |y|)}\}$. For $Y \subseteq X$ let $G[Y]$ denote the $|Y|$-partite subgraph induced by $\{(x, j) \mid x \in Y \wedge 0 \leq j < |x|\}$. Note that $G[\{x, y\}]$ consists of at most $\gcd(|x|, |y|)$ vertex-disjoint bicliques (one for each congruence class modulo $\gcd(|x|, |y|)$). We prove the correctness of our reduction to Multi-Colored Clique.

▶ **Lemma 12.** *Let $k \in \mathbb{N}$. Then, G has a size-k clique if and only if there is some $j \in \mathbb{N}$ and some $Y \subseteq X$ with $|Y| = k$ and $x[j]^\circ = 0$ for all $x \in Y$.*

Proof. (⇒) Let C be a size-k clique in G. Let a_i denote its vertices and let $x_i \in X$ denote the sequence in X that contains a_i. Then, for each edge $\{a_i, a_{i'}\}$ in C, we have $a_i \equiv a_{i'} \pmod{\gcd(|x_i|, |x_{i'}|)}$ and, by Lemma 4, there is some $j \in \mathbb{N}$ with $j \equiv a_i \pmod{|x_i|}$ for each $a_i \in V(C)$. Thus, $x_i[j]^\circ = x_i[a_i]^\circ = 0$ for all $a_i \in V(C)$.

(⇐) Let $j \in \mathbb{N}$ and let $Y \subseteq X$ be a size-k set of sequences with $x[j]^\circ = 0$ for all $x \in Y$. For each $x \in Y$, let $a_x := j \bmod |x|$. Then, by Lemma 4, we have $a_x \equiv a_y \pmod{\gcd(|x|, |y|)}$ for all distinct $x, y \in Y$, implying that the edge $\{(x, a_x), (y, a_y)\}$ exists in E. ◀

Lemma 12 can also be used to reduce PCA to a version of Multi-Colored Clique that allows more color classes than vertices in the target colorful clique.

A straight-forward method of solving Multi-Colored Clique on instances produced by Lemma 12 is to guess one of the at most $\gcd(|x|, |y|)$ bicliques for each pair of color classes of the vertex $|X|$-partition.

▶ **Corollary 13.** *PCA can be solved in $d^{|X|^2} n^{\mathcal{O}(1)}$ time where $d = \max_{x,y \in X, x \neq y} \gcd(|x|, |y|)$.*

Note that in the worst case, this algorithm is not faster than the known algorithm for Tally-Intersection but our algorithm works even for Periodic Character Alignment and might be much faster on instances where d has a small value.

Since the number of color classes in the constructed Multi-Colored Clique instance is $|X|$ and PFA is W[1]-hard by Corollary 10, Lemma 12 implies that MCC remains W[1]-hard, even on such restricted instances. While this has no further consequences for the problems at hand, we found this to be a noteworthy fact, in particular given the central role that MCC plays in many W[1]-hardness proofs.

▶ **Corollary 14.** *Multi-Colored Clique is* W[1] *hard with respect to k even if the induced subgraph between any pair of distinct color classes is a union of vertex-disjoint bicliques.*

3.1.2 Parameter #runs

We present an ILP formulation that will set a variable i to the index we are looking for. We further use variables z_j to indicate which sequences x_j have a 1 at position i and variables r_j indicating how many times x_j is repeated before reaching position i. Thus, for each x_j, either

$z_j = 1$ or, for each run (s_j, t_j), we have $i < s_j$ or $i > t_j$ (herein, we use another variable $y_{j,st}$ to indicate which of the two applies).

$$\begin{aligned} z_j \in \{0,1\}, r_j \in \mathbb{N} & \qquad \text{for all } x_j \in X \\ y_{j,st} \in \{0,1\} & \qquad \text{for all } x_j \in X \text{ and } (s,t) \in Bl_1(x_j) \\ i \in \mathbb{N} & \end{aligned}$$

minimize $\sum z_j$ subject to

$$r_j|x_j| \leq i < (r_j+1)|x_j| \qquad \text{for all } x_j \in X \tag{2}$$
$$i - r_j|x_j| + (1 - y_{j,st})|x_j| > t \qquad \text{for all } x_j \in X, (s,t) \in Bl_1(x_j) \tag{3}$$
$$i - r_j|x_j| - (y_{j,st} + z_j)|x_j| < s \qquad \text{for all } x_j \in X, (s,t) \in Bl_1(x_j) \tag{4}$$

Note that $i - r_j|x_j|$ denotes the position $i_j := i \bmod |x_j|$ in x_j. If this position is in the interval $[s,t]$ for any $(s,t) \in Bl_1(x_j)$, then $y_{j,st} = 0$ (since, otherwise $i_j + (1 - y_{j,st})|x_j| = i_j \leq t$ contradicting (3)) and $z_j = 1$ (since, otherwise, $i_j - (0 + z_j)|x_j| = i_j \geq s$ contradicting (4)).

▶ **Proposition 15.** *PCA and PFA can be solved in $b^{\mathcal{O}(b)} \cdot n^{\mathcal{O}(1)}$ time, where $b = \sum_{x \in X} Bl_1(x)$.*

Note that, while seemingly natural, guessing the correct run for each sequence in X (plus one for "not in the solution") is not enough to solve PCA since, by Proposition 16, the problem remains NP-hard, even if each sequence contains a single 0. However, this hardness breaks down for PFA (corresponding to prepending an $2^{|X|}$-way guessing step for PCA). We consider it an interesting open question whether PFA is fixed-parameter tractable with respect to the maximum number of runs in any input sequence. A possible hint that PFA might also be hard for a constant number of runs per sequence is given by Lemma 2 which could be used to represent a multi-run sequence by a set of single-run sequences.

3.1.3 Parameter #lengths $|\mathcal{L}|$

While PCA is fixed-parameter tractable for $L = \max \mathcal{L}$, the stronger parameter $|\mathcal{L}|$ is not enough to yield tractability results even if every $x \in X$ contains exactly one 0.

▶ **Proposition 16.** *PCA is* W[1]*-hard with respect to $k + |\mathcal{L}(X)|$ even if every $x \in X$ contains exactly one 0.*

Proof. Let X be an instance of PFA and let $Z_x := \{z \mid x[z] = 0\}$ denote the set of positions where x has a 0 for each $x \in X$. We set $k := |X|$ and $X' := \{x_z \mid x \in X, z \in Z_x\}$ where $|x_z| := |x|$ and $x_z[t] := 0$ if and only if $t = z$ for all $t \in [0, |x| - 1]$. We show that X is a yes-instance of PFA if and only if (X', k) is a yes-instance of PCA.

($\Rightarrow$) If X is a yes-instance of PFA, there is some $i \in \mathbb{N}$ such that, for all $x \in X$, we have $x[i]^\circ = 0$ and, by construction, $x_{i \bmod |x|}[i]^\circ = 0$. Thus, (X', k) is a yes-instance of PCA.

($\Leftarrow$) If (X', k) is a yes-instance, then there are $Y \subseteq X'$ and $k \in \mathbb{N}$ with $|Y| \geq k$ and $y[i]^\circ = 0$ for all $y \in Y$. By construction (and no two sequences in X have the same length), Y contains exactly one sequence of length $|x|$ for each $x \in X$. Thus, for each $x \in X$ there is some $z \in Z_x$ such that $x_z \in Y$. Hence, $x_z[i]^\circ = 0$ and therefore $x[i]^\circ = 0$ for all $x \in X$. Consequently, X is a yes-instance of PFA.

Since, by Corollary 10, PFA is W[1]-hard when parameterized by $|X|$, PCA is W[1]-hard when parameterized by $k + |\mathcal{L}(X)|$ even if every $x \in X$ contains exactly one zero. ◀

4 Periodic Cop and Robber

We consider a cops and robbers game with rules identical to those introduced in [26, 24] (variant with one cop and one robber), which has been defined on edge-periodic graphs by Erlebach and Spooner [13]. There are two players, a cop C and a robber R. First C, then R (in knowledge of C's choice) choose a start vertex on a given edge-periodic graph $G^\tau = (V, E, \tau)$. Then for each time step $t \geq 0$, both players take alternating turns, either remaining on their current vertex or moving to a vertex that is adjacent to their current vertex in time step t. In every time step, C moves first, knowing R's position and R moves second, knowing the move C just made. The game terminates whenever C moves onto a vertex on which R resides in that moment. If there is a strategy for C such that the game terminates, then G^τ is called cop-win and this strategy is a winning strategy for C. Otherwise, G^τ is called robber-win and the strategy for R that enables infinite evasion of C is called winning strategy for R.

Cop-Win Periodic Cop and Robber (PCnR)
Input: An edge-periodic graph $G^\tau = (V, E, \tau)$.
Question: For the above defined cop and robber game, is there a winning strategy for C, that is, is G^τ cop-win?

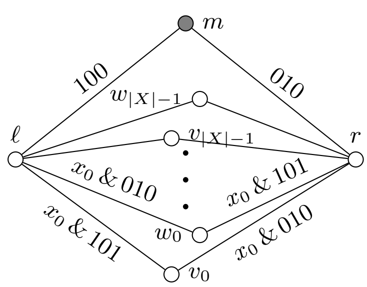
We will show NP-hardness of this problem by reducing from PFA.

▶ **Theorem 17.** *Cop-Win Periodic Cop and Robber is* NP*-hard even on edge-periodic temporal graphs G^τ whose underlying graph is $K_{2,n}$ for some $n \in \mathbb{N}$ and G^τ consists of two disjoint stars in each time step. Further, Cop-Win Periodic Cop and Robber is* W[1]*-hard when parameterized by $|G| = |V| + |E|$ in this setting.*

Proof. Given an instance X of PFA which is the 3-fold blow up of some X', we construct a $K_{2,2|X|+1}$ on the vertex set $V_1 \uplus V_2$ with $V_1 = \{\ell, r\}$ and $V_2 = \{m\} \cup \{v_j, w_j \mid 0 \leq j < |X|\}$

$$\tau(\{\ell, m\}) := 100 \qquad \forall_{x_j \in X}\ \tau(\{\ell, w_j\}) := \tau(\{r, v_j\}) := x_j \,\&\, 010$$
$$\tau(\{m, r\}) := 010 \qquad \forall_{x_j \in X}\ \tau(\{\ell, v_j\}) := \tau(\{r, w_j\}) := x_j \,\&\, 101$$

Note that G^τ consists of two disjoint stars in each time step. To show that X is a yes-instance of PFA if and only if the constructed G^τ is cop-win, we use the following claim.



▷ Claim 18. At the start of a time step t with $t \bmod 3 = 0$, the cop has a winning strategy if

- C is on vertex ℓ and R is on any of the vertices $\{v_j \mid 0 \leq j < |X|\}$ or
- C is on vertex r and R is on any of the vertices $\{w_j \mid 0 \leq j < |X|\}$.

Proof. Suppose that C is on ℓ and R is on v_j for some $j \in [0, |X| - 1]$. By definition, there is a time step $t' \geq t$ such that the edge $\{\ell, v_j\}$ exists in time step t' but not in any time step between t and $t' - 1$. Hence, C can stay on the vertex ℓ until the beginning of time step t' and move onto v_j in step t'. To show that this is a winning strategy, it remains to show that R must stay on v_j in all time steps between t and $t' - 1$. By construction, if $\{\ell, v_j\}$ does not exist in some time step t'' with $t'' \bmod 3 = 0$, then the edge $\{v_j, r\}$ does not exist in the time steps t'', $t'' + 1$, and $t'' + 2$. Thus, R must stay on v_j until the beginning of time step t' since $\{v_j, r\}$ is the only other edge adjacent to v_j. The case for C is on r and R is on $w_j, j \in [0, |X| - 1]$ is symmetric. ◁

($\Rightarrow$) Suppose that X is a yes-instance of PFA. We describe a winning strategy for the cop in G^τ. Choose vertex ℓ as the start position. Since X is a yes-instance of Periodic Full Character Alignment, Observation 7 implies that there is some $i \in \mathbb{N}$ with $i \bmod 3 = 0$ and $x_j[(i + p)]^\circ = 0$ for all $x_j \in X$ and $p \in [0, 2]$. Thus, in time steps i, $i + 1$ and $i + 2$, no edge except $\{\ell, m\}$ and $\{m, r\}$ exists. The strategy for C is thus to stay on ℓ for the first $i - 1$ time steps. By Claim 18 we can assume that at the beginning of time step i, the robber is not on any vertex of $\{v_j \mid 0 \leq j < |X|\}$ as, otherwise, C has a winning strategy. In time step i, the cop moves to m which is a valid move since $i \bmod 3 = 0$ and $\tau(\{\ell, m\})[0] = 1$. If R is currently on m then C wins immediately. Otherwise, the robber is on r or on any of the vertices $\{w_j \mid 0 \leq j < |X|\}$. As no edge incident with any of these vertices exists in time step i, the robber can only stay on his current vertex. Since $\tau(\{m, r\})[1] = 1$, the cop can move to r in time step $i + 1$ and thus win the game if R is on r. Otherwise, R has to be on w_j for some $j \in [0, |X| - 1]$ and can only stay on this vertex on the time steps $i, i + 1$, and $i + 2$. Thus, C can stay on r until the beginning of time step $i + 3$. Due to Claim 18, C has winning strategy. Hence, C has a winning strategy in all cases.

($\Leftarrow$) We show the contraposition. Supposing that X is a no-instance of Periodic Full Character Alignment, we show that the robber has a winning strategy on G^τ, that is, G^τ is a no-instance of Cop-Win Periodic Cop and Robber. Since X is a no-instance, for each time step t, there is some index $i(t)$ such that $x_{i(t)}[t]^\circ \neq 0$ and $i(t) = i(t - 1)$ unless $t \bmod 3 = 0$. In the following, we call a vertex u_r *safe* for a vertex $u_c \neq m$ if (a) $u_r \neq m$, (b) u_c and u_r are at distance two in the underlying graph G and (c) $u_c \in \{v_j, w_j\}$ implies $u_r \in \{v_j, w_j\} \setminus \{u_c\}$ for all $j \in [0, |X| - 1]$. Further, we also call u_r *safe* for m in time step t if (a) $u_r = w_{i(t)}$ and $t \bmod 3 \neq 2$ or (b) $u_r = r$ and $t \bmod 3 = 2$.

▷ Claim 19. Let the cop move onto (or stay on) a vertex u_c in time step t. Then, the robber can move onto a vertex u_r that is safe for u_c in step t.

Proof. The proof is by induction on t. For $t = 0$, note that r and ℓ are safe with respect to each other, $v_{i(0)}$ is safe for all vertices except m, r, ℓ, and $v_{i(0)}$ and $w_{i(0)}$ is safe for m and $v_{i(0)}$. Thus, no matter the start position of the cop, the robber can move onto a safe vertex. For the induction step, we know that, in step $t - 1$, the robber was on a safe vertex u'_r with respect to the vertex u'_c of the cop.

Case 1: $u_c = m$. If $t \bmod 3 = 0$, then $u'_c \in \{m, \ell\}$ by construction of G^τ, implying $u'_r = r$ (by induction hypothesis), and the robber can move to $w_{i(t)}$, which is safe for m in step t. If $t \bmod 3 = 1$ then, by construction of G^τ, we have $u'_c \in \{r, m\}$, implying $u'_r \in \{\ell, w_{i(t-1)} = w_{i(t)}\}$ and the robber can move to $w_{i(t)}$, which is safe for m in step t. If $t \bmod 3 = 2$ then, $u'_c = m$ by construction of G^τ, implying $u'_r = w_{i(t-1)} = w_{i(t)}$ and the robber can move to r which is safe for m in time step t.

Case 2: $u_c \neq m$ and $u'_c = m$. Then, by construction of G^τ, either $t \bmod 3 = 0$ and $u_c = \ell$ and $u'_r = r$ (by induction hypothesis), which is also safe for u_c, or $t \bmod 3 = 1$ and

$u_c = r$ and $u'_r = w_{i(t-1)} = w_{i(t)}$ (by induction hypothesis), implying that the robber can move to ℓ which is safe for r in step t.

Case 3: $u_c \neq m$ and $u'_c \neq m$. If $u'_c = u_c$, then u'_r is still safe for u_c so the robber can stay put. Thus, suppose $u'_c \neq u_c$. If $u_c = r$, then $u'_c \in \{v_j, w_j\}$ for some $j \in \mathbb{N}$ by construction of G^τ and $u'_r = \{v_j, w_j\} - u'_c$ (by induction hypothesis) and, since the edge $\{r, u'_c\}$ exists in time step t, so does $\{\ell, u'_r\}$, and the robber can move to ℓ. The case that $u_c = \ell$ is symmetric. Finally, if $u_c = v_j$ for some $j \in \mathbb{N}$, then $u'_c \in \{\ell, r\}$ and $u'_r \in \{\ell, r\} - u'_c$ (by induction hypothesis). Since the edge $\{u'_c, v_j\}$ exists in $G[t]$, so does $\{u'_r, w_j\}$ and, thus, the robber can move to w_j. The case that $u_c = w_j$ is symmetric. ◁

Now, if $u_c = m$, $u_r = r$ in time step t with $t \bmod 3 = 2$, then the cop cannot catch the robber in time step $t+1$. Otherwise, being safe implies being at distance two in the underlying graph and the cop cannot catch the robber in time step $t+1$. Thus, Claim 19 implies a winning strategy for the robber.

The W[1]-hardness for Cop-Win Periodic Cop and Robber when parameterized by $|G|$ in this setting now follows directly by Corollary 10 and the fact that $|G| \in \mathcal{O}(|X|)$. ◀

A quick inspection of the proof of Theorem 17 allows us to also exclude polynomial compressions, even when combining structural parameters with $L(\tau(E))$.

▶ **Proposition 20.** *Cop-Win Periodic Cop and Robber does not admit a polynomial compression when parameterized by the sum of $L(\tau(E))$, the maximum degree of the underlying graph, and the treedepth, unless* NP ⊆ coNP/poly.

Sketch. Given k instances G_i^τ of PCnR as constructed in the proof of Theorem 17 (where we assume that k is a power of 2), construct a binary tree T (whose edges are labeled with 1) with k leaves, which we identify with the ℓ-vertices in the instances G_i^τ. The cop chooses the root of T as its start vertex and follows the robber into one of the G_i^τ-subgraphs. At this point, the winning strategies of the cop and robber, respectively, hold[2] as in the proof of Theorem 17. Thus, the cop wins if and only if all input instances are cop-win, implying that PCnR is AND-composable, which implies the result (see [10, Section 15]). ◀

Note that Theorem 17 excludes any kind of graph-structural parameterization as the hardness relies entirely on the hardness of the underlying algebraic problem. However, since the problem can be solved trivially if the underlying graph G is a tree or a clique, we might still hope that restricting G (and G^τ) to be either very sparse or very dense yields tractability. It turns out that such restrictions would have to be severe.

▶ **Theorem 21.** *Cop-Win Periodic Cop and Robber is* NP*-hard, even if (a) for all $t \in \mathbb{N}$, there are at most two edges in $G[t]$ and (b) the underlying graph is bipartite, planar, and has a max-leaf number of four. Further, Cop-Win Periodic Cop and Robber is* W[1]*-hard when parameterized by $|G| = |V| + |E|$ in this setting.*

▶ **Corollary 22.** *Cop-Win Periodic Cop and Robber is* NP*-hard even on edge-periodic temporal graphs whose underlying graph can be turned into a clique by deleting four vertices. Further, Cop-Win Periodic Cop and Robber is* W[1]*-hard when parameterized by $|G| = |V| + |E|$ in this setting.*

[2] Recall that the cop wins by staying on ℓ until the point where he can win in 3 moves using m. Thus, the robber cannot escape the chosen G_i^τ in this scenario.

It seems thus that the intrinsic hardness of the underlying algebraic problem can hardly be overcome by structural limitations of the temporal graph. therefore, we investigated parameters of the τ-function leading to tractability for PFA, such as the number of lengths $|\mathcal{L}(\tau(E))|$ and the number of 1s in each $\tau(e)$. For PCnR, however, we are met with hardness again.

▶ **Theorem 23.** *Cop-Win Periodic Cop and Robber is* W[1]*-hard when parameterized by the sum of* $|\mathcal{L}(\tau(E))|$ *and the vertex cover number of the underlying graph, even if (a) the underlying graph is bipartite and planar, (b) at most two edges exist in each time step, and (c)* $\tau(e)$ *contains a single 1 for each* $e \in E$.

▶ **Theorem 24.** *Unless* NP ⊆ coNP/poly*, there is no polynomial compression for Cop-Win Periodic Cop and Robber parameterized by the sum of* $L(\tau(E))$ *and the number of vertices to delete to turn the underlying graph into a clique.*

For the parameter $L := |\mathcal{L}(\tau(E))|$, we can use the $\mathcal{O}(\operatorname{lcm}(\mathcal{L}(\tau(E))) \cdot n^3)$-time algorithm of Erlebach and Spooner [13] (as $\operatorname{lcm}(\mathcal{L}(\tau(E))) < L^L$). Together with Theorem 24, we obtain the following corollary.

▶ **Corollary 25.** *Cop-Win Periodic Cop and Robber is solvable in* $\mathcal{O}(L^L n^3)$ *time and does not admit a polynomial compression when parameterized by* L*, unless* NP ⊆ coNP/poly.

Curiously, by extending standard techniques [17, 10, 11], kernelization can be pushed to any root of $\operatorname{lcm}(\mathcal{L}(\tau(E)))$ but not to its logarithm.

▶ **Proposition 26.** *For every* $d \in \mathbb{N}$*, Cop-Win Periodic Cop and Robber admits a kernel of size* $\sqrt[d]{\operatorname{lcm}(\mathcal{L}(\tau(E)))}$ *but does not admit a kernel whose size is polynomial in* $\log(\operatorname{lcm}(\mathcal{L}(\tau(E))))$*, unless* NP ⊆ coNP/poly.

With dynamic graphs, the modification of the cop and robber game in which the robber moves first in each round makes sense (contrary to the "static" case, where this modification has no effect on the game).

Robber-First Cop-Win Periodic Cop and Robber (RfPCnR)
Input: An edge-periodic graph $G^\tau = (V, E, \tau)$.
Question: For the periodic cop and robber game where in every time step R moves first and C moves second, is there a winning strategy for C?

Unsurprisingly, this modification does little to change the complexity of the problem.

▶ **Theorem 27.** *Robber-First Cop-Win Periodic Cop and Robber is* NP*-hard even on edge-periodic temporal graphs whose underlying graph is* $K_{2,n}$ *for some* $n \in \mathbb{N}$*. Further, Robber-First Cop-Win Periodic Cop and Robber is* W[1]*-hard when parameterized by* $|G| = |V| + |E|$ *in this setting.*

5 Conclusion

In this paper we showed that, unless P = NP, there is no polynomial-time algorithm for the pursuit-evasion game with one cop and one robber on edge-periodic temporal graphs, thereby answering an open question of Erlebach and Spooner [13]. We analyzed the parameterized complexity of deciding this game and showed that it is W[1]-hard, even if the parameter is the size of the underlying input graph. Thus, unless FPT = W[1], there is no graph parameter for which Cop-Win Periodic Cop and Robber is fixed-parameter tractable.

This hardness even applies for severely restricted instances. The open question remains whether PCnR is in NP (and is thus NP-complete) or maybe even harder.

Our hardness results are based on intuitive algebraic problems called PERIODIC CHARACTER ALIGNMENT and PERIODIC FULL CHARACTER ALIGNMENT, asking whether a given set of periodic sequences over $\{0, 1\}$ is unanimously 0 at any index or, equivalently, whether all of a given set of Tally-DFAs accept a common word.

As a side node, we showed that MULTI-COLORED CLIQUE remains W[1]-hard when parameterized by the size k of the sought clique, even if the induced subgraph between any pair of distinct color classes is the union of vertex-disjoint bicliques.

References

1. Eleni C. Akrida, George B. Mertzios, and Paul G. Spirakis. The temporal explorer who returns to the base. In *Algorithms and Complexity*, pages 13–24, Cham, 2019. Springer.
2. Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *Journal of Computer and System Sciences*, 107:108–123, 2020.
3. Saeed Akhoondian Amiri, Lukasz Kaiser, Stephan Kreutzer, Roman Rabinovich, and Sebastian Siebertz. Graph Searching Games and Width Measures for Directed Graphs. In *32nd STACS'15*, volume 30 of *LIPIcs*, pages 34–47, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
4. János Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(2):161–172, 2006.
5. Dietmar Berwanger, Anuj Dawar, Paul Hunter, and Stephan Kreutzer. Dag-width and parity games. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 524–536. Springer, 2006.
6. Étienne Bézout. *Théorie générale des équations algébriques par M. Bézout.* de l'imprimerie de Ph.-D. Pierres, rue S. Jacques, 1779.
7. Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical computer science*, 209(1-2):1–45, 1998.
8. Graham R. Brightwell and Peter Winkler. Gibbs measures and dismantlable graphs. *J. Comb. Theory Ser. B*, 78(1):141–166, January 2000.
9. Timothy H Chung, Geoffrey A Hollinger, and Volkan Isler. Search and pursuit-evasion in mobile robotics. *Autonomous robots*, 31(4):299, 2011.
10. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. `doi:10.1007/978-3-319-21275-3`.
11. Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
12. Thomas Erlebach, Michael Hoffmann, and Frank Kammer. On temporal graph exploration. In *Automata, Languages, and Programming*, pages 444–455. Springer, 2015.
13. Thomas Erlebach and Jakob T Spooner. A game of cops and robbers on graphs with periodic edge-connectivity. In *46th SOFSEM'20*, pages 64–75. Springer, 2020.
14. Henning Fernau and Andreas Krebs. Problems on finite automata and the exponential time hypothesis. *Algorithms*, 10(1):24, 2017.
15. Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theoretical Computer Science*, 806:197–218, 2020.
16. Fedor V Fomin and Dimitrios M Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical computer science*, 399(3):236–245, 2008.
17. Frank Gurski, Carolin Rehs, and Jochen Rethmann. Knapsack problems: A parameterized point of view. *Theoretical Computer Science*, 775:93–108, 2019.

18 Markus Holzer and Martin Kutrib. Descriptional and computational complexity of finite automata—a survey. *Information and Computation*, 209(3):456–470, 2011. Special Issue: 3rd International Conference on Language and Automata Theory and Applications (LATA 2009).

19 Paul Hunter and Stephan Kreutzer. Digraph measures: Kelly decompositions, games, and orderings. *Theoretical Computer Science (TCS)*, 399, 2008.

20 Kenneth F. Ireland and Michael I. Rosen. *A classical introduction to modern number theory*, chapter Congruence, pages 28–38. Number 84 in Graduate texts in mathematics. Springer, New York, 2nd edition, 1990.

21 Thor Johnson, Neil Robertson, Paul D Seymour, and Robin Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

22 David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *Journal of Computer and System Sciences*, 64(4):820–842, 2002.

23 Nicolas Nisse. Network Decontamination. In *Distributed Computing by Mobile Entities*, volume 11340 of *LNCS*, pages 516–548. Springer, 2019.

24 Richard Nowakowski and Ivan Rival. The smallest graph variety containing all paths. *Discrete Mathematics*, 43(2-3):223–234, 1983.

25 Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Math.*, 43(2–3):235–239, January 1983.

26 Alain Quilliot. *Jeux et pointes fixes sur les graphes*. PhD thesis, Ph. D. Dissertation, Université de Paris VI, 1978.

27 Paul D Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1):22–33, 1993.

28 Manuel Sorge, Mathias Weller, et al. The graph parameter hierarchy. work in progress, 2019. URL: `https://manyu.pro/assets/parameter-hierarchy.pdf`.