Cops & Robber on Periodic Temporal Graphs: Characterization and Improved Bounds

Jean-Lou De Carufel[1](✉), Paola Flocchini[1], Nicola Santoro[2], and Frédéric Simard[1]

[1] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada
jdecaruf@uottawa.ca

[2] School of Computer Science, Carleton University, Ottawa, Canada

Abstract. We study the classical *Cops and Robber* game when the cops and the robber move on an infinite periodic sequence $\mathcal{G} = (G_0, \ldots, G_{p-1})^*$ of graphs on the same set V of n vertices: in round t, the topology of $\mathcal{G}$ is $G_i = (V, E_i)$ where $i \equiv t \pmod{p}$. As in the traditional case of *static graphs*, the main concern is on the characterization of the class of *periodic temporal graphs* where k cops can capture the robber. Concentrating on the case of a *single* cop, we provide a characterization of copwin periodic temporal graphs. Based on this characterization, we design an algorithm for determining if a periodic temporal graph is copwin with time complexity $O(p\ n^2 + n\ m)$, where $m = \sum_{i \in \mathbb{Z}_p} |E_i|$, improving the existing $O(p\ n^3)$ bound. Let us stress that, when $p = 1$ (i.e., in the *static* case), the complexity becomes $O(n\ m)$, improving the best existing $O(n^3)$ bound.

1 Introduction

Cops & Robber Games. *Cops & Robber* (*C&R*) is a pursuit-evasion game played in rounds on a finite graph G between a set of $k \geq 1$ cops and a single robber. Before starting the game, an initial position on the vertices of G is chosen first by the cops, then by the robber. Then, in each round, first the cops, then the robber, move to a neighbouring vertex or (if allowed by the variant of the game) stay in their current location. The game ends if at least one cop moves to the vertex currently occupied by the robber, in which case the cops *capture* the robber and win. The robber wins by forever avoiding capture. In the original version [30,33], the graph G is connected and undirected, there is a single cop and, in each round, the players are allowed to move to a neighbouring vertex or not to move. Moreover, the cops and the robber have perfect information. It has been then extended to permit multiple cops [2]. This version, which we shall call *standard*, is the most commonly investigated.

Supported by the Natural Sciences and Engineering Research Council of Canada.

S. Rajsbaum et al. (Eds.): SIROCCO 2023, LNCS 13892, pp. 386–405, 2023.
https://doi.org/10.1007/978-3-031-32733-9_17

Among the many variants of this game (for a partial list, see [4,5]), two are of particular interest to us. The first is the (much less investigated) natural generalization when the graph G is a strongly connected directed graph [25,27]; we shall refer to this version as *directed*. Also of interest is the variant, called *fully active* (or *restless*), in which the players must move in every round; proposed for the standard game [19], it can obviously be extended to the directed version.

In the extensive research (see [5] for a review), the main focus is on characterizing the class of *k-copwin* graphs; i.e., those graphs where there exists a strategy allowing k cops to capture the robber regardless of the latter's decisions. Related questions are to determine the minimum number of cops capable of winning in G, called the *copnumber of* G, or just to decide whether k cops suffice. Currently, the most efficient algorithm for deciding whether or not a graph is k-copwin in the standard game is $O(k\ n^{k+2})$ [32], which yields $O(n^3)$ for the case $k = 1$.

In the existing literature on the *C&R* game, with only a couple of recent exceptions, all results are based on the assumption that the graph on which the game is played is *static*; that is, its link structure is the same in every round. The question naturally arises: what happens if the link structure of the graph changes in time, possibly in every round? This question is particularly relevant in view of the intense research on time-varying graphs in the last two decades.

Temporal Graphs. The extensive investigations on computational aspects of time-varying graphs have been motivated by the development and increasing importance of highly dynamic networks, where the topology is continuously changing. Such systems occur in a variety of settings, ranging from wireless ad-hoc networks to social networks. Various formal models have been advanced to describe the dynamics of these networks (e.g., [8,21,34]).

When time is *discrete*, as in the *C&R* game, the dynamics of these networks is usually described as an infinite sequence $\mathcal{G} = (G_0, G_1, \ldots)$, called *temporal graph* (or *evolving graph*), of static graphs $G_i = (V, E_i)$ on the same set V of vertices; the graph G_i is called *snapshot* (of $\mathcal{G}$ at time i), and the aggregate graph $G = (V, \cup_i E_i)$ is called the *footprint* (or *underlying*) graph. This model, originally suggested in [15,20], has become the de-facto standard in the ensuing investigations.

All the studies are being carried out under some assumptions restricting the arbitrariness of the changes. Some of these assumptions are on the "connectivity" of the graphs G_i in the sequence; they range from the (strong) *1-interval connectivity* requiring every G_i to be connected (e.g., [22,26,31]), to the weaker *temporal connectivity* allowing each G_i to be disconnected but requiring the sequence to be *connected over time* (e.g., [7,17]). Another class of assumptions is on the "frequency" of the existence of the links in the sequence. An important assumption in this class is *periodicity*: there exists a positive integer p such that $G_i = G_{i+p}$ for all $i \in \mathbb{Z}$ (e.g., [16,23,24]).

A large number of studies has focused on *mobile entities* operating on temporal graphs, under different combinations of the above (and other) restrictive assumptions. Among them, computations include *graph exploration*, *dispersion*, and *gathering* (e.g., [1,6,10–13,17,18]; for a recent survey see [9]). Until very recently, none of these studies considered *C&R*.

Conceptually, the extension of *C&R* to a temporal graph $\mathcal{G} = (G_0, G_1, \dots)$ is quite natural. Initially, first the cops, then the robber, choose a starting position on the vertices of G_0. At the beginning of round $t \geq 0$, the players are in G_t and, after making their decisions and moves (according to the rules of the game), they find themselves in G_{t+1} in the next round. The game ends if and only if a cop moves to the vertex currently occupied by the robber; in this case the cops have won. The robber wins by forever preventing the cops from winning.

Existing Results. This extension has been first investigated by Erlebach and Spooner [14]. They considered the standard game with a single cop under the *periodic* frequency restriction; they presented an algorithm to determine if a periodic temporal graph is copwin, and mentioned that it can be extended to $k > 1$ cops. In this pioneering study, the results are obtained by reformulating the problem in terms of a reachability problem and solving the latter; this, unfortunately, does not provide insights on the temporal nature of the game.

Using the same reduction to reachability games, and thus with the same drawbacks as [14], Balev et al. [3] studied the standard game in temporal graphs under the *1-interval connectivity* restriction. They showed how to determine whether a single cop can capture the robber in a fixed temporal window, and indicated how their algorithm can be extended to the case of $k > 1$ cops. They also considered an "on-line" version of the problem, i.e. where the sequence of graphs is a priori unknown; these results however are not relevant for the "full-disclosure" problem studied here.

Finally, if the temporal graph is not given explicitly (i.e., as the sequence of snapshots), but only implicitly by means of the Boolean *edge-presence* function (e.g., [8]), the problem of deciding whether a single cop has a winning strategy in the standard game on a periodic temporal graph has been shown to be NP-hard [28,29], answering a question raised in [14].

Contributions. We focus on the *C&R* game in *periodic* temporal graphs, concentrating on the case of a *single* cop. We study the *unified* version of the game defined as follow: in every round $i \geq 0$, G_i is directed and the players are restless. Observe that the standard and the directed versions, both in the original and restless variant, can be expressed as a restless game played on (appropriately chosen) directed graphs.

For the unified game, we provide a complete characterization of copwin periodic temporal graphs, establishing several basic properties on the nature of a copwin game in such graphs. We do so by employing a compact representation of periodic temporal graphs, introducing the novel notion of augmented arenas, and using these structures to extend to the temporal domain classical concepts such as *corners* and *covers*.

These characterization results are *general*, in the sense that they do not rely on any assumption on properties such as connectivity, symmetry, reflexivity held (or not held) by the individual snapshot graphs in the sequence. The only requirement, for the game to be defined, and thus *playable*, is that every node in the graph must have an outgoing edge.

Based on these results, we design an algorithm that determines if a periodic temporal graph is copwin in time $O(p\,n^2 + n\,m)$, where $m = \sum_{i\in\mathbb{Z}_p} |E_i|$, improving on the existing $O(p\,n^3)$ bound established by [14]. Let us stress that, in the static case studied in the literature, the complexity becomes $O(n\,m)$, improving the best existing $O(n^3)$ bound [32]; in particular our bound becomes $O(n^2)$ for sparse graphs.

All our results are established for the unified version of the game. Therefore, all the characterization properties and algorithmic results hold for the standard and for the directed games studied in the literature, both when the players are restless and when they are not. They hold also for all those settings, not considered in the literature, where there is a mix of nodes: those where the players must leave and those where the players can wait; furthermore such a mix might be time-varying (i.e., different in every round). Due to space constraints, some proofs are missing.

2 Definitions and Terminology

2.1 Graphs and Time

Static Graphs. We denote by $G = (V, E)$, or sometimes by $G = (V(G), E(G))$, the *directed graph* with set of vertices V and set of edges $E \subseteq V \times V$. A self-loop is an edge of the form (u, u); if $(u, u) \in E$ for all $u \in V$, then we will say that G is *reflexive*. If $(v, u) \in E$ whenever $(u, v) \in E$, we will say that G is *symmetric* (or *undirected*). Given a graph G', if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$, then we say G' is a *subgraph* of G and write $G' \subseteq G$. A subgraph $G' \subseteq G$ is *proper*, written $G' \subset G$, if $G' \neq G$. For reasons apparent later, we shall refer to a graph G so defined as a *static graph*, and say it is *playable* if every node has at least one outgoing edge.

Temporal Graphs. A *time-varying graph* $\mathcal{G}$ is a graph whose set of edges changes in time[1]. A *temporal graph* is a time-varying graph where time is assumed to be discrete and to have a start; i.e., *time* is the set $\mathbb{Z}^+$ of positive integers including 0. A temporal graph $\mathcal{G}$ is represented as an infinite sequence $\mathcal{G} = (G_0, G_1, \dots)$ of static graphs $G_i = (V, E_i)$ on the same set of vertices V; we shall denote by $n = |V|$ the number of vertices. The graph G_i is called the *snapshot* of $\mathcal{G}$ *at time* $i \in \mathbb{Z}^+$, and the aggregate graph $G = (V, \bigcup_i E_i)$ is called the *footprint* of $\mathcal{G}$. A temporal graph $\mathcal{G}$ is said to be *reflexive* if all its snapshots are reflexive, *symmetric* if all its snapshots are symmetric.

Given two nodes $x, y \in V$, a strict *journey* (or *temporal walk*), from x to y starting at time t is any finite sequence $\pi(x, y) = \langle (z_0, z_1), (z_1, z_2), \dots, (z_{k-1}, z_k) \rangle$ where $z_0 = x, z_k = y$, and $(z_i, z_{i+1}) \in E_{t+i}$ for $0 \leq i < k$. In the following, for simplicity, we will omit the adjective "strict". A temporal graph $\mathcal{G}$ is *temporally connected* if for any $u, v \in V$ and any time $t \in \mathbb{Z}^+$ there is a journey from u to v that starts at time t. Observe that, if $\mathcal{G}$ is temporally connected, then its

[1] The terminology in this section is from [8].

footprint is strongly connected even when all its snapshots are disconnected. A temporal graph $\mathcal{G}$ is said to be *always connected* (or 1-*interval connected*) if all its snapshots are strongly connected.

A temporal graph $\mathcal{G}$ is *periodic* if there exists a positive integer p such that for all $i \in \mathbb{Z}^+$, $G_i = G_{i+p}$. If p is the smallest such integer, then p is called the *period* of $\mathcal{G}$ and $\mathcal{G}$ is said to be p-periodic. We shall represent a p-periodic temporal graph $\mathcal{G}$ as $\mathcal{G} = (G_0, \ldots, G_{p-1})^*$; all operations on the indices will be taken modulo p. An example of a temporal periodic graph $\mathcal{G}$ with $p = 4$ is shown in Fig. 1; observe that $\mathcal{G}$ is temporally connected, however most of its snapshots are disconnected directed graphs, and none of them is strongly connected.

Let $\mathcal{G} = (G_0, G_1, \ldots, G_{p-1})^*$ and $\mathcal{H} = (H_0, H_1, \ldots, H_{p-1})^*$ be two temporal periodic graphs with the same period on the same set V of vertices; we say $\mathcal{H}$ is a *periodic subgraph* of $\mathcal{G}$, written $\mathcal{H} \subseteq \mathcal{G}$, if $H_i \subseteq G_i$ for every $i \in \mathbb{Z}_p = \{0, 1, \ldots, p-1\}$. We shall denote by $\mathcal{H} \subset \mathcal{G}$ the fact that $\mathcal{H}$ is a *proper* subgraph of $\mathcal{G}$; i.e., $\mathcal{H} \subseteq \mathcal{G}$ but $\mathcal{H} \neq \mathcal{G}$. Let us point out the obvious but useful fact that static graphs are temporal periodic graphs with period $p = 1$. In this paper we focus on *C&R* games in *periodic* temporal graphs, henceforth referred to simply as *periodic graphs*, concentrating on the case of a *single* cop.

Consider the following class of directed static graphs, we shall call *arenas*.

Definition 1 (Arena). *Let $k \geq 1$ be an integer and W be a non-empty finite set. An* arena *of length k on W is any static directed graph $\mathcal{M} = (\mathbb{Z}_k \times W, E(\mathcal{M}))$ where $E(\mathcal{M}) \subseteq \{((i, w), ([i+1]_k, w')) | i \in \mathbb{Z}_k$ and $w, w' \in W\}$, and $[i]_k$ denotes i modulo k.*

A periodic graph $\mathcal{G} = (G_0, \ldots, G_{p-1})^*$ with period p and set of nodes V has a unique correspondence with the arena $\mathcal{D} = (\mathbb{Z}_p \times V, E(\mathcal{D}))$ where, for all $i \in \mathbb{Z}_p$, $((i, u), ([i+1]_p, v)) \in E(\mathcal{D})$ if and only if $(u, v) \in E_i$, called the *arena of* $\mathcal{G}$. In particular, the arena $\mathcal{D}$ of $\mathcal{G}$ explicitly preserves the snapshot structure of $\mathcal{G}$: for all $i \in \mathbb{Z}_p$, there is an obvious one-to-one correspondence between the snapshot G_i of $\mathcal{G}$ and the subgraph S_i of $\mathcal{D}$, called *slice* (or *stage*), where $V(S_i) = \{(i, v), v \in V\}$ and $E(S_i) = \{((i, u), ([i+1]_p, v)) | (u, v) \in E_i)\}$. An example of a periodic graph $\mathcal{G}$ and its arena $\mathcal{D}$ is shown in Fig. 1. In the following, when no ambiguity arises, $\mathcal{D}$ shall indicate the arena of $\mathcal{G}$.

The vertices of an arena $\mathcal{D}$ will be called *temporal nodes*. Given a temporal node $(i, u) \in V(S_i)$ we shall denote by $N_i(u, \mathcal{D})$ the set of its outneighbours, and by $\Gamma_i(u, \mathcal{D}) = \{v \in V | ([i+1]_p, v) \in N_i(u, \mathcal{D})\}$ the corresponding set of nodes in G_i. We define $\Gamma_i^{\text{in}}(u, \mathcal{D})$ similarly for the inneighbours. A temporal node $(i, u) \in V(S_i)$ is said to be a *star* if $\Gamma_i(u, \mathcal{D}) = V$. It is said to be *anchored* if there exists a journey from some node $(0, v) \in V(S_0)$ to (i, u). A *subarena* of $\mathcal{D} = (\mathbb{Z}_p \times V, E(\mathcal{D}))$ is any arena $\mathcal{D}' = (\mathbb{Z}_p \times V, E(\mathcal{D}'))$ where $E(\mathcal{D}') \subseteq E(\mathcal{D})$; we shall denote by $\mathcal{D}' \subset \mathcal{D}$ the fact that $\mathcal{D}'$ is a subarena of $\mathcal{D}$ with $E(\mathcal{D}') \subset E(\mathcal{D})$.

2.2 Cop & Robber Game in Periodic Graphs

The extension of the game from static to temporal graphs is quite natural. Initially, first the cop, then the robber, chooses a starting position on the vertices

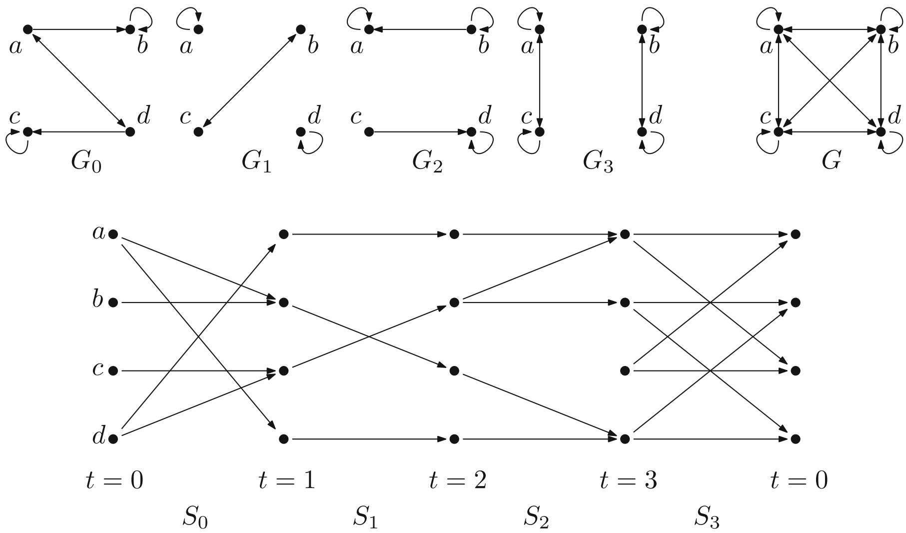

Fig. 1. A periodic graph $\mathcal{G} = (G_0, G_1, G_2, G_3)^*$, its footprint G, and the corresponding arena.

of G_0. Then, at each time $t \in \mathbb{Z}^+$, first the cop, then the robber, moves to a vertex adjacent to its current position in G_i, where $i = [t]_p$. Thus, in round t, the players are in $G_{[t]_p}$ and, after making their decisions and moves, they find themselves in $G_{[t+1]_p}$ in the next round. The game ends if and only if the cop moves to the vertex currently occupied by the robber; in this case the cop has won. The robber wins by forever preventing the cop from winning. Moreover, the cops and the robber have perfect information.

A play on the arena $\mathcal{D}$ of $\mathcal{G}$ follows the play on $\mathcal{G}$ in a direct obvious way: at each time $t \in \mathbb{Z}^+$, first the cop, then the robber, chooses a new node in the outneighbourhood of its current position and moves there. The cop wins and the game ends if it manages to move to a temporal node $([t+1]_p, u)$ while the robber is on $([t]_p, u)$. The robber wins by forever escaping capture from the cop, in which case the game never ends.

We consider the version of the game where all players are restless, i.e., they all move to a different node in each round. In this version, the only requirement on $\mathcal{G}$ is that it is *playable*: in each snapshot, every node must have an outgoing edge. In what follows we only consider playable periodic graphs. No other requirement such as connectivity, symmetry or reflexivity is imposed on $\mathcal{G}$.

We call this version of the game *unified*. Observe that the standard version, both in the original or restless variant, as well as the non-restless directed version can actually be redefined as a restless game played in this unified version: a pair of directed edges between a pair of nodes corresponds to an unidirected link between them, and the presence of a self-loop at a node allows the players currently there not to move to a different node in the current round.

A *configuration* is a triple $(t, c, r) \in \mathbb{Z}^+ \times V \times V$, denoting the position $c \in V$ of the cop and $r \in V$ of the robber at the beginning of round $t \in \mathbb{Z}^+$. Let $\mathcal{CG} = (V(\mathcal{CG}), E(\mathcal{CG}))$ be the infinite directed graph, called *configuration graph* of $\mathcal{D}$, describing all the possible configurations (t, u, v) together with the following subset of their temporal connections in $\mathcal{D}$:

$$V(\mathcal{CG}) = \{(t, u, v)|\ t \in \mathbb{Z}^+; ([t]_p, u), ([t]_p, v) \in V(\mathcal{D})\}$$
$$E(\mathcal{CG}) = \{((t, u, v), (t + 1, u', v'))|\ t \in \mathbb{Z}^+; u \neq v; u' \in \Gamma_{[t]_p}(u, \mathcal{D}), v' \in \Gamma_{[t]_p}(v, \mathcal{D})\}.$$

Observe that $\mathcal{CG}$ is acyclic; the *source* nodes (i.e., the nodes with no in-edges) are those with $t = 0$, the *sink* nodes (i.e., the nodes with no out-edges) are those with $u = v$. A playing *strategy for the cop* is any function $\sigma_c : V(\mathcal{CG}) \to V$ where, for every $(t, u, v) \in V(\mathcal{CG})$, $\sigma_c(t, u, v) \in \Gamma_{[t]_p}(u, \mathcal{D})$, and $\sigma_c(t, u, v) = u$ if $u = v$; it specifies where the cop should move in round t if the cop is at $([t]_p, u)$, the robber is at $([t]_p, v)$, and it is the cop's turn to move. A playing strategy σ_r for the robber is defined in a similar way.

A configuration (t, u, v) is said to be *copwin* if there exists a strategy σ_c such that, starting from (t, u, v), the cop wins the game regardless of the strategy σ_r of the robber; such a strategy σ_c will be said to be *copwin for* (t, u, v). A strategy σ_c is said to be *copwin* if there exists a temporal node $(0, u)$ such that σ_c is winning for all $(0, u, v)$, $v \in V$. If a copwin strategy exists, then $\mathcal{G}$ and its arena $\mathcal{D}$ are said to be *copwin*, else they are *robberwin*.

3 Copwin Periodic Graphs

3.1 Preliminary

In the analysis of the standard game played in a static graph, an important role is played by the notions of *corner* node and its *cover*. The usual meaning is that if the robber is on the corner, after the cop has moved to the cover, no matter where the robber plays, the robber gets captured by the cop *in the next round.*

In an arena $\mathcal{D}$, the same meaning is provided directly by the notions of "temporal corner" and "temporal cover".

Definition 2 *(Temporal Corner and Temporal Cover). A temporal node* (t, u) *in an arena* $\mathcal{D}$ *is said to be a* temporal corner *of temporal node* $(t+1, v)$ *if* $u \neq v$ *and* $\Gamma_t(u, \mathcal{D}) \subseteq \Gamma_{t+1}(v, \mathcal{D})$. *The temporal node* $(t+1, v)$ *is said to be a* temporal cover *of* (t, u).

Lemma 1. *Every copwin arena contains a temporal corner.*

This necessary condition, although important, provides only limited indications on how to solve the characterization problem.

3.2 Augmented Arenas and Characterization

The crucial element in the characterization of copwin periodic graphs is the notion of *augmented arena.*

Definition 3 *(Augmented Arena). Let $\mathcal{D}$ be the arena of $\mathcal{G}$. An* augmented arena *$\mathcal{A}$ of $\mathcal{D}$ is an arena of length p such that $\mathcal{D} \subseteq \mathcal{A}$ and, for each edge $((t,x),(t+1,y)) \in E(\mathcal{A})$, the configuration (t,x,y) is winning for the cop in $\mathcal{D}$.*

We shall refer to the edges of the augmented arena $\mathcal{A}$ of $\mathcal{D}$ as *shadow edges*. Observe that, by definition, all edges of $\mathcal{D}$ are shadow edges of $\mathcal{A}$. Let $\mathbb{A}(\mathcal{D})$ denote the set of augmented arenas of $\mathcal{D}$. Observe that, by definition of $\mathcal{D}$, for each edge $((t,x),(t+1,y)) \in E(\mathcal{D})$, the configuration (t,x,y) is winning for the cop in $\mathcal{D}$. Therefore, $\mathcal{D} \in \mathbb{A}(\mathcal{D})$. Further observe the following:

Property 1. The partial order $(\mathbb{A}(\mathcal{D}), \subset)$ induced by edge-set inclusion on $\mathbb{A}(\mathcal{D})$ is a complete lattice. Hence $(\mathbb{A}(\mathcal{D}), \subset)$ has a maximum which we denote by $\mathcal{A}^*$.

We have now the elements for the characterization of copwin periodic graphs.

Theorem 1 (Characterization Property). *An arena $\mathcal{D}$ is copwin if and only $\mathcal{A}^*$ contains an anchored star.*

Proof (**only if**). Let $\mathcal{A}^*$ contain an anchored star (t,u), $t \in \mathbb{Z}_p$. By definition of star, $\Gamma_t(u, \mathcal{A}^*) = V$; thus, by definition of augmented arena, for every $v \in V$ the configuration (t,u,v) is copwin, i.e. there is a copwin strategy σ_c from (t,u,v).

Since (t,u) is anchored, there exists a journey $\pi((0,x),(t,u))$, starting at time 0 and ending at time t, to (t,u) from some temporal node $(0,x)$. Consider now the cop strategy σ'_c of: (1) initially positioning itself on the temporal node $(0,x)$, (2) then moving according to the journey $\pi((0,x),(t,u))$ and, once on (t,u), (3) following the copwin strategy σ_c from (t,u,w), where w is the position of the robber at the beginning of round t. This strategy σ'_c is winning for all $(0,x,v)$, $v \in V$; hence $\mathcal{D}$ is copwin.

(**if**) Let $\mathcal{D}$ be copwin. We then show that there must exist an augmented arena $\mathcal{A}$ of $\mathcal{D}$ that contains an anchored star. Since $\mathcal{D}$ is copwin, by definition, there must exist some starting position $(0,c)$ for the cop such that, for all positions $(0,r)$ initially chosen by the robber, the cop eventually captures the robber. In other words, all the configurations $(0,c,v)$ with $v \in V$ are copwin; thus the arena $\mathcal{A}$ obtained by adding to $E(\mathcal{D})$ the set of edges $\{((0,c),(1,v)) | v \in V\}$ is an augmented arena of $\mathcal{D}$ and $(0,c)$ is an anchored star. By Property 1, $E(\mathcal{A}) \subseteq E(\mathcal{A}^*)$ and the theorem follows. $\square$

The characterization of copwin periodic graphs provided by Theorem 1 indicates that, to determine whether or not an arena $\mathcal{D}$ is copwin, it suffices to check whether $\mathcal{A}^*$ contains an anchored star. To be able to transform this fact into an effective solution procedure, some additional concepts need to be introduced and properties established.

3.3 Shadow Corners and Augmentation

Other crucial elements in the analysis of copwin periodic graphs are the concepts of corner and cover, introduced in Sect. 3.1 for arenas, now in the context of augmented arenas.

Definition 4 *(Shadow Corner and Shadow Cover). Let $\mathcal{A}$ be an augmented arena of $\mathcal{D}$. A temporal node (t, u) is a* shadow corner *of a temporal node $(t+1, v)$, with $v \neq u$, if $\Gamma_t(u, \mathcal{D}) \subseteq \Gamma_{t+1}(v, \mathcal{A})$. The temporal node $(t+1, v)$ will then be called the* shadow cover *of (t, u).*

By definition, any temporal corner is a shadow corner, and its temporal covers are shadow covers. An example is shown in Fig. 2; the red links indicate the neighbours of node (t, u) in $\mathcal{D}$, while in green are indicated the edges to the neighbours of $(t+1, v)$ that exists in $\mathcal{A}$ but not in $\mathcal{D}$.

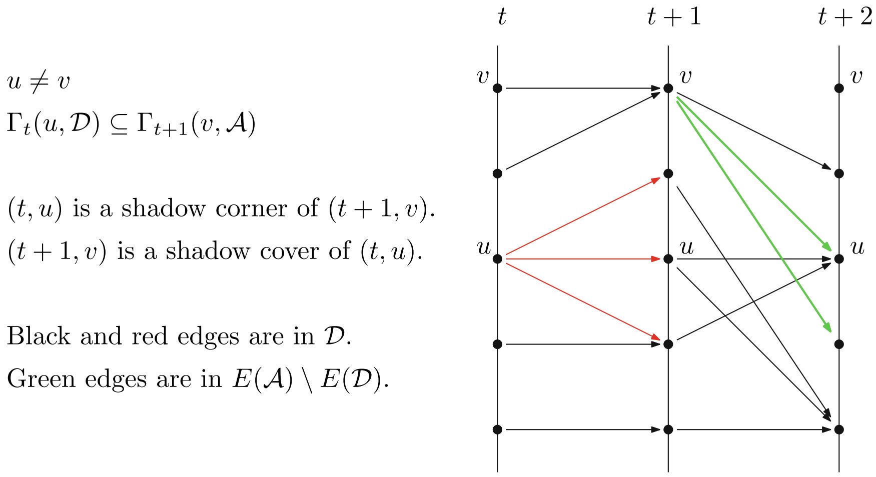

Fig. 2. Node (t, u) is a shadow corner of $(t+1, v)$. (Color figure online)

The role that shadow corners play with regards to the set $\mathbb{A}(\mathcal{D})$ of augmented arenas of $\mathcal{D}$ is expressed by the following.

Theorem 2 (Augmentation Property). *Let $\mathcal{A} \in \mathbb{A}(\mathcal{D})$; $(t, x), (t, y) \in V(\mathcal{D})$; and $z \in \Gamma_t(x, \mathcal{D})$. If (t, y) is a shadow corner of $(t+1, z)$, then the arena $\mathcal{A}' = \mathcal{A} \cup \{((t, x), (t+1, y))\}$ is an augmented arena of $\mathcal{D}$.*

Proof. Let $\mathcal{A}$ be an augmented arena of $\mathcal{D}$ and let $(t, x), (t, y), (t+1, z) \in V(\mathcal{D})$ where $z \in \Gamma_t(x, \mathcal{D})$ and (t, y) is a shadow corner of $(t+1, z)$. The theorem follows if $((t, x), (t+1, y))$ is already an edge of $\mathcal{A}$. Consider the case where $((t, x), (t+1, y)) \notin E(\mathcal{A})$. Since (t, y) is a shadow corner of $(t+1, z)$, then for every $w \in \Gamma_t(y, \mathcal{D})$ we have that $((t+1, z), (t+2, w)) \in E(\mathcal{A})$; i.e., $(t+1, z, w)$ is winning for the cop. Since $z \in \Gamma_t(x, \mathcal{D})$, if the cop moves from (t, x) to $(t+1, z)$ when the robber is on (t, y), then regardless of the robber's move, the resulting configuration would be winning for the cop. In other words, (t, x, y) is a winning configuration for the cop. It follows that $\mathcal{A}' = \mathcal{A} \cup \{((t, x), (t+1, y))\}$ is an augmented arena of $\mathcal{D}$. $\square$

In other words, given an augmented arena, by identifying a (still unconsidered) shadow corner and its covers, new shadow edges may be determined and added to form a denser augmented arena.

3.4 Determining $\mathcal{A}^*$

The properties expressed by Theorem 2, in conjunction with that of Theorem 1, provide an algorithmic strategy to construct $\mathcal{A}^*$: start from an augmented arena; determine new shadow edges; add them to the set of shadow edges, creating a denser augmented arena; repeat this process until the current augmented arena $\mathcal{A}$ either contains an anchored star or is $\mathcal{A}^*$.

To be able to employ the above strategy, a condition is needed to determine if the current augmented arena of $\mathcal{D}$ is indeed $\mathcal{A}^*$. This is provided by the following.

Theorem 3 (Maximality Property). *Let $\mathcal{A} \in \mathbb{A}(\mathcal{D})$. Then $\mathcal{A} = \mathcal{A}^*$ if and only if, for every edge $((t, x), (t+1, y)) \notin E(\mathcal{A})$, there exists no $z \in \Gamma_t(x, \mathcal{D})$ such that $\Gamma_t(y, \mathcal{D}) \subseteq \Gamma_{t+1}(z, \mathcal{A})$.*

Proof **(only if).** By contradiction, let $\mathcal{A} = \mathcal{A}^*$ but there exists an edge $((t, x), (t+1, y)) \notin E(\mathcal{A})$ and a temporal node $z \in \Gamma_t(x, \mathcal{D})$ such that $\Gamma_t(y, \mathcal{D}) \subseteq \Gamma_{t+1}(z, \mathcal{A})$. This means that (t, y) is a shadow corner of $(t+1, z)$. By Theorem 2, $\mathcal{A}' = \mathcal{A} \cup \{((t, x), (t+1, y))\}$ is an augmented arena of $\mathcal{D}$; however, $E(\mathcal{A}')$ contains one more edge than $E(\mathcal{A})$, contradicting the assumption that $\mathcal{A}$ is maximum.

(if) Let $\mathcal{A} \neq \mathcal{A}^*$; that is, there exists $((t, x), (t+1, y)) \in E(\mathcal{A}^*) \setminus E(\mathcal{A})$. By definition, the configuration (t, x, y) is copwin; let σ_c be a cop winning strategy for the configuration (t, x, y); i.e., starting from (t, x, y), the cop wins the game regardless of the strategy σ_r of the robber.

Let $\mathcal{C} = (V(\mathcal{C}), E(\mathcal{C})) \subseteq \mathcal{CG}$ be the directed acyclic graph of configurations induced by σ_c starting from (t, x, y), and defined as follows: (1) $(t, x, y) \in V(\mathcal{C})$; (2) if $(t', u, v) \in V(\mathcal{C})$ with $t' \geq t$ and $u \neq v$, then, for all $w \in \Gamma_{t'}(v, \mathcal{D})$, $(t'+1, \sigma_c(t'+1, u, v), w) \in V(\mathcal{C})$ and $((t', u, v), (t'+1, \sigma_c(t'+1, u, v), w))) \in E(\mathcal{C})$.

Observe that in $\mathcal{C}$ there is only one source (or root) node, (t, x, y), and every $(t', w, w) \in V(\mathcal{C})$ is a sink (or terminal) node. Since σ_c is a winning strategy for the root, every node in $\mathcal{C}$ is a copwin configuration, and every path from the root terminates in a sink node (see Fig. 3).

Partition $V(\mathcal{C})$ into two sets, U and W where $U = \{(i, u, v) | ((i, u), (i+1, v)) \in E(\mathcal{A})\}$ and $W = V(\mathcal{C}) \setminus U$. Observe that every sink of $V(\mathcal{C})$ belongs to U; on the other hand, since $((t, x), (t+1, y)) \notin E(\mathcal{A})$ by assumption, the root belongs to W (see Fig. 4). Given a node $\kappa = (i, u, v) \in V(\mathcal{C})$, let $\mathcal{C}[\kappa]$ denote the subgraph of $\mathcal{C}$ rooted in κ.

Claim. *There exists $\kappa \in V(\mathcal{C})$ such that all nodes of $\mathcal{C}[\kappa]$ except the root belong to U.*

Proof of Claim. Let P_0 be the set of sinks of $\mathcal{C}$. Starting from $k = 0$, consider the set P_{k+1} of all in-neighbours of any node of P_k; if P_{k+1} does not contains an

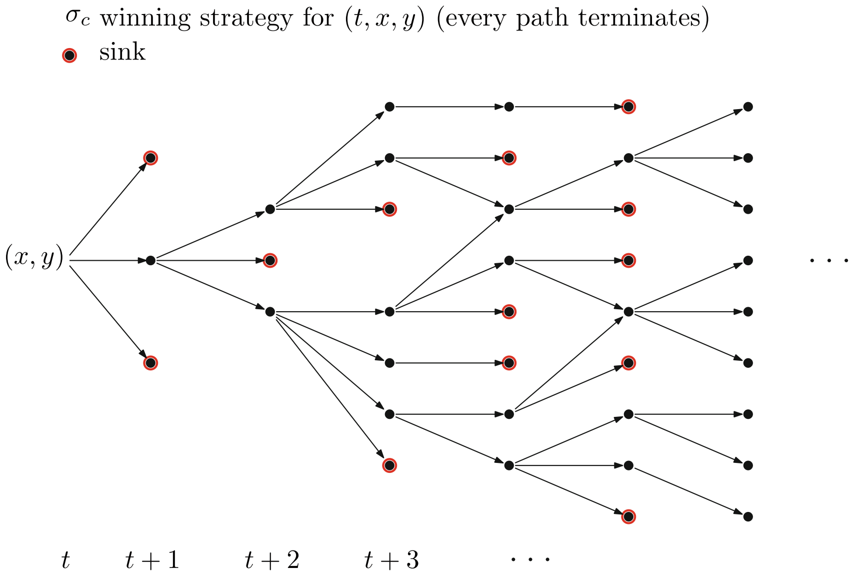

Fig. 3. The directed acyclic graph $\mathcal{C}$ of configurations induced by σ_c starting from (t, x, y). (Color figure online)

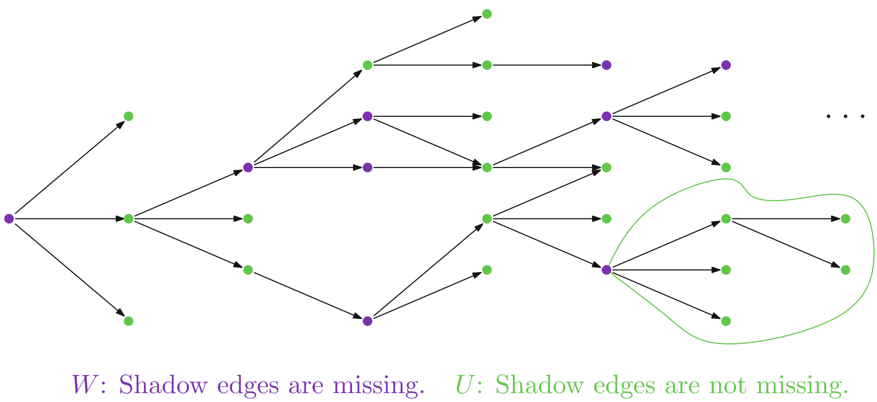

Fig. 4. The sets U (green) and W (purple). (Color figure online)

element of W, then increase k and repeat the process. Since $(t, x, y) \in W$, this process terminates for some $j \geq 1$, and the Claim holds for every $\kappa \in P_j$. $\square$

Let (t', x', y') be a node of $V(\mathcal{C})$ satisfying the above Claim (see Fig. 5). Thus $((t', x'), (t'+1, y')) \notin E(\mathcal{A})$ but, since (t', x', y') is copwin, $((t', x'), (t'+1, y')) \in \mathcal{A}^*$. By the Claim, all other nodes of $\mathcal{C}[(t', x', y')]$ belong to U, in particular the set of nodes $\{(t'+1, w, z) | w = \sigma_c(t', x', y'), z \in \Gamma_{t'}(y', \mathcal{D})\}$. This means that,

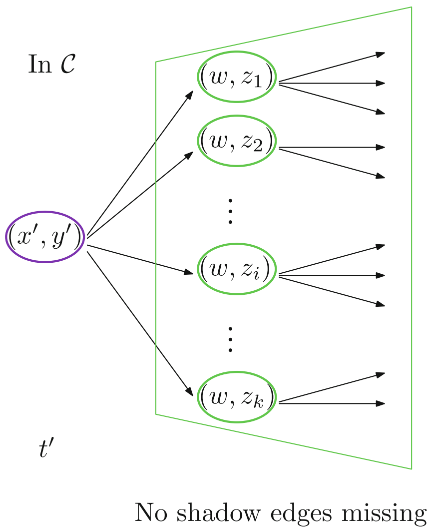

Fig. 5. (t', x', y') satisfies the Claim. (Color figure online)

for every $z \in \Gamma_{t'}(y', \mathcal{D})$, $(t' + 1, w, z) \in E(\mathcal{A})$. In other words, $\Gamma_{t'}(y', \mathcal{D}) \subseteq \Gamma_{t'+1}(w, \mathcal{A})$; that is, (t', y') is a shadow corner of $(t' + 1, w)$ (see Fig. 6).

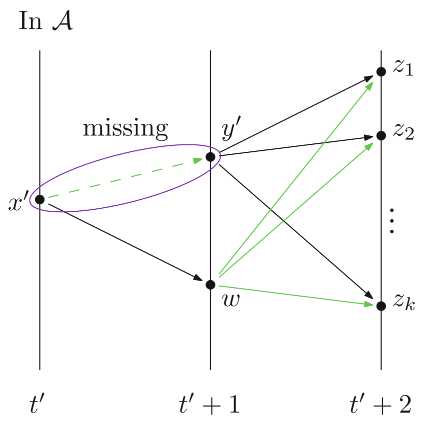

Fig. 6. (t', y') is a shadow corner of $(t' + 1, w)$. (Color figure online)

Summarizing: by assumption $\mathcal{A} \neq \mathcal{A}^*$; as shown, $((t', x'), (t' + 1, y')) \notin E(\mathcal{A}^*) \setminus E(\mathcal{A})$, and $w \in \Gamma_{t'}(x', \mathcal{D})$ is a shadow cover of (t', y'); that is, $\Gamma_{t'}(y', \mathcal{D}) \subseteq \Gamma_{t'+1}(w, \mathcal{A})$, concluding the proof of the **if** part of the theorem. $\square$

4 Algorithmic Determination

In this section we show that the results established in the previous sections provide all the tools necessary to design an algorithm to determine whether or not a periodic graph $\mathcal{G}$ is copwin. Furthermore, if $\mathcal{G}$ is copwin, the algorithm can actually provide a winning cop strategy σ_c.

4.1 Solution Algorithm

General Strategy. Given a periodic graph $\mathcal{G}$, or equivalently its arena $\mathcal{D}$, to determine whether or not it is copwin, by Theorem 1, it is sufficient to determine whether or not its maximal augmented arena $\mathcal{A}^*$ contains an anchored star. Hence, informally, a basic solution approach is to start from $\mathcal{A} = \mathcal{D}$, repeatedly determine a "new" shadow edge (i.e., in $E(\mathcal{A}^*) \setminus E(\mathcal{A})$), using Theorem 2, and consider the new augmented arena obtained by adding such an edge. This process is repeated until either the current augmented arena $\mathcal{A}$ contains an anchored star, or no other "missing" shadow edge exists. In the former case, by Theorem 1, $\mathcal{D}$ is copwin; in the latter case, by Theorem 3, the current augmented arena is $\mathcal{A}^*$ and, if it does not contain an anchored star, $\mathcal{D}$ is robberwin.

A general strategy based on this approach operates in a sequence of iterations, each composed of two operations: the examination of a shadow edge, and the examination of new shadow corners (if any) determined in the first operation. More precisely, in each iteration: (i) A "new" (i.e., not yet examined) shadow edge $e = ((t, x), (t + 1, y))$ is examined to determine if its presence transforms some nodes into new shadow corners of (t, x). (ii) Each of these new shadow corners is examined, determining if its presence generates new shadow edges. By the end of the iteration, the shadow edge e and the new shadow corners of (t, x) examined in this iteration are removed from consideration. This iterative process continues until there are no new shadow edges to be examined (i.e. $\mathcal{A} = \mathcal{A}^*$) or there is an anchored star in $\mathcal{A}$.

```
GENERAL STRATEGY

1. While there is a still unexamined shadow edge e = ((t,x),(t+1,y)) in A do:
2.    If there are still unexamined shadow corners covered by (t,x) then:
3.       For each such shadow corner (t−1,z) do:
4.          If there are new shadow edges due to (t−1,z) then:
5.                Add them to A to be examined.
6.          Remove (t−1,z) from consideration as a shadow corners of (t,x).
                                                   (i.e., mark it as examined).
7.    Remove e from consideration (i.e., mark it as examined).
8. If there is an anchored star in A, then D is copwin else it is robberwin.
```

Fig. 7. Outline of general strategy where the iterative process terminates when $\mathcal{A} = \mathcal{A}^*$.

An outline of the strategy, where the iterative process is made to terminate when $\mathcal{A} = \mathcal{A}^*$, is shown in Fig. 7.

Algorithm Description. Let us present the proposed algorithm, CopRobberPeriodic, which follows directly the general strategy described above to determine whether or not an arena $\mathcal{D} = ((Z_p \times V), E(\mathcal{D}))$ is copwin, where $V = \{v_1, \ldots, v_n\}$.

We denote by $\mathcal{A}$ the current augmented arena of $\mathcal{D}$, by A its adjacency matrix, and by A_t the adjacency matrix of slice S_t of $\mathcal{A}$. Auxiliary structures used by the algorithm include the queue $\mathcal{SE}$, of the known shadow edges that have not been examined yet; a $n \times n$ Boolean matrix SE_t for each t, initialized to A_t, used to indicate shadow edges already known; a $n \times n$ Boolean matrix SC_t for each t, initialized to zero and used to indicate the detected shadow corners; more precisely, $SC_t[x, y] = 1$ indicates that (t, x) has been determined to be a shadow corner of $(t+1, y)$,

The algorithm is composed of two phases: *Initialization*, in which all the necessary structures are set up and preliminary computations are performed; and *Iteration*, a repetitive process where the two basic operations of the general strategy (described in Sect. 4.1) are performed in each iteration: examination of a "new" shadow edge (to determine "new" shadow corners generated by that edge) and examination of the "new" shadow corners (to determine "new" shadow edges generated by that corner).

The structure used to determine new shadow corners is the set $\{\mathrm{DIF}(t, x, y) : t \in \mathbb{Z}_p, x, y \in V\}$ of $n^2 p$ Boolean arrays of dimension n. For all $x, v \in V$ and $t \in \mathbb{Z}_p$, the value of the cell $\mathrm{DIF}(t, x, y)[i]$ indicates whether $v_i \in \Gamma_t(x, \mathcal{D}) \setminus \Gamma_{t+1}(y, \mathcal{A})$ (in which case $\mathrm{DIF}(t, x, y)[i] = 1$) or $v_i \in \Gamma_t(x, \mathcal{D}) \cap \Gamma_{t+1}(y, \mathcal{A})$ (in which case $\mathrm{DIF}(t, x, y)[i] = 0$). Note that, if $v_i \notin \Gamma_t(x, \mathcal{D})$, the value of $\mathrm{DIF}(t, x, y)[i]$ is left undefined; indeed, the algorithm only initializes and uses the $|\Gamma_t(x, \mathcal{D})|$ cells corresponding to the elements of $\Gamma_t(x, \mathcal{D})$; we shall call those cells the *core* of $\mathrm{DIF}(t, x, y)$.

The algorithm also maintains a variable $\phi(\mathrm{DIF}(t, x, y))$ indicating the current number of core cells with value "1" in array $\mathrm{DIF}(t, x, y)$; this variable is initialized to $|\Gamma_t(x, \mathcal{D})|$. Observe that, by definition of $\mathrm{DIF}(t, x, y)$, $\phi(\mathrm{DIF}(t, x, y)) = 0$ iff (t, x) is a shadow corner of $(t+1, y)$.

In each iteration of the *Iteration* phase, a new shadow edge is taken from $\mathcal{SE}$, added to the augmented arena $\mathcal{A}$, and examined. The examination of a shadow edge $((t, x), (t+1, y))$ involves (i) the update of $\mathrm{DIF}(t-1, z, x)[y]$ for any in-neighbour $(t-1, z)$, in $\mathcal{D}$, of (t, y) and, for any such in-neighbour, (ii) the test to see if the presence of the edge $((t, x), (t+1, y))$ in the augmented arena has created new shadow corners among such in-neighbours[2]. If new shadow corners exist, they may in turn have created new shadow edges originating from the in-neighbours, in $\mathcal{D}$, of (t, x). In fact, any in-neighbour $(t-1, w)$ of (t, x) such that $((t-1, w), (t, z))$ is not already in the augmented arena is a new shadow

[2] Such would be any $(t-1, z)$ for which the update has resulted in an array $\mathrm{DIF}(t-1, z, x)$ that contains only zero entries.

edge: a move of the cop from $(t-1, w)$ to (t, x) is fatal for the robber wherever it goes; in such a case, the algorithm then adds $((t-1, w), (t, z))$ to $\mathcal{SE}$.

The pseudo code of the algorithm is shown in Algorithm 1. Not shown are several very low level (rather trivial) implementation details. These include, for example, the fact that the core cells of $\text{DIF}(t, x, y)$ are connected through a doubly linked list, and that, for efficiency reasons, we also maintain two additional doubly linked lists: one going through the core cells of the array containing "1", the other linking the core cells containing "0".

4.2 Analysis

Correctness. Let us prove the correctness of Algorithm CopRobberPeriodic. Let $\mathcal{D} = (\mathbb{Z}_p \times V, E(\mathcal{D}))$ be the arena of a p-periodic graph with $n = |V|$ and $m = |E(\mathcal{D})|$.

Lemma 2. *Algorithm* CopRobberPeriodic *terminates after at most* $|E(\mathcal{A}^*)| - |E(\mathcal{D})|$ *iterations.*

Given an augmented arena $\mathcal{A}$ and a shadow edge $e = ((t, x), (t+1, y)) \in E(\mathcal{A}^*) \setminus E(\mathcal{A})$, we shall say that e is an *implicit* shadow edge of $\mathcal{A}$ if there exists $z \in \Gamma_t(x, \mathcal{D})$ such that (t, y) is a shadow corner of $(t+1, z)$ in $\mathcal{A}$.

Lemma 3. *At the end of the* Initialization *phase: (i) for all and only the temporal corners (t, x) of $(t+1, y)$ in $\mathcal{D}$, $SC_t[x, y] = 1$ and $\phi(\text{DIF}(t, x, y)) = 0$; (ii) all implicit shadow edges of $\mathcal{D}$ are in $\mathcal{SE}$; furthermore, the entry in SE of all edges of $\mathcal{D}$ and implicit shadow edges of $\mathcal{D}$, is* 1.

Let us consider the *Initialization* phase as iteration 0 of the *Iteration* phase; hence, the entire algorithm can be viewed as a sequence of iterations. Denote by $\mathcal{A}_j$ the augmented arena at the beginning of the j-th iteration, with $\mathcal{A}_0 = \mathcal{D}$. We now show that, at the beginning of iteration j, all shadow corners of $\mathcal{A}_{j-1}$ have been examined and all implicit shadow edges of $\mathcal{A}_{j-1}$ are in $\mathcal{SE}$.

Lemma 4. *At the beginning of iteration $j > 0$:*

(a) $\phi(\text{DIF}(t, x, y)) = 0$ if and only if (t, x) is a shadow corner of $(t+1, y)$ in $\mathcal{A}_{j-1}$; furthermore, in such a case, $SC_t[x, y] = 1$.
(b) $\mathcal{SE}$ contains all the implicit shadow edges of $\mathcal{A}_{j-1}$; furthermore, in SE, the entry of the edges of $\mathcal{A}_{j-1}$ and of the implicit shadow edges of $\mathcal{A}_{j-1}$ is 1.

Theorem 4. *Algorithm* CopRobberPeriodic *correctly determines whether or not an arena $\mathcal{D}$ is copwin.*

Complexity. Let us analyze the cost of Algorithm CopRobberPeriodic. Given $\mathcal{D} = (\mathbb{Z}_p \times V, E(\mathcal{D}))$, let m_i denote the number of edges of slice S_i of $\mathcal{D}$, $i \in Z_p$, and $m = |E(\mathcal{D})| = \sum_{i=0}^{p-1} m_i$ the total number of edges of $\mathcal{D}$. As usual, $n = |V|$.

Algorithm 1: CopRobberPeriodic

Input: Arena $\mathcal{D} = (\mathbb{Z}_p \times V, E(\mathcal{D}))$, with $V = \{v_1, \ldots, v_n\}$

1 *Initialization*
2 $\mathcal{A} := \mathcal{D}$
3 $SE := A$
4 $\mathcal{SE} := \emptyset$
5 $SC :=$*Zero* /* a table of p zero matrices, each of size $n \times n$ */
6 **foreach** $t \in \mathbb{Z}_p,\ u, v \in V$ **do**
7     $\phi\,(\mathrm{DIF}(t,u,v)) := |\Gamma_t(u,\mathcal{D})|$
8   **foreach** $w \in \Gamma_t\,(u,\mathcal{D})$ **do**
9   **if** $A_{t+1}[v,w] = 1$ **then**
10             $\mathrm{DIF}(t,u,v)[w] := 0$
11             $\phi\,(\mathrm{DIF}(t,u,v)) := \phi\,(\mathrm{DIF}(t,u,v)) - 1$
12   **if** $\phi\,(\mathrm{DIF}(t,u,v)) = 0$ **and** $SC_t[u,v] = 0$ **then**
13                 $SC_t[u,v] := 1$
14   **foreach** $z \in \Gamma^{\mathrm{in}}_{t+1}\,(v,\mathcal{D})$ **do**
15   **if** $SE_t[z,u] = 0$ **then**
16                         $SE_t[z,u] := 1$
17                         $\mathcal{SE} \leftarrow ((t,z),(t+1,u))$
18   **else**
19             $\mathrm{DIF}(t,u,v)[w] := 1$
20 *Iteration*
21 **while** $\mathcal{SE} \neq \emptyset$ **do**
22     $((t,x),(t+1,y)) \leftarrow \mathcal{SE}$
23     $A_t(x,y) := 1$
24   **foreach** $z \in \Gamma^{\mathrm{in}}_t\,(y,\mathcal{D})$ **do**
25   **if** $\mathrm{DIF}(t-1,z,x)[y] = 1$ **then**
26             $\mathrm{DIF}(t-1,z,x)[y] := 0$
27             $\phi\,(\mathrm{DIF}(t-1,z,x)) := \phi\,(\mathrm{DIF}(t-1,z,x)) - 1$
28   **if** $\phi\,(\mathrm{DIF}(t-1,z,x)) = 0$ **and** $SC_{t-1}[z,x] = 0$ **then**
29             $SC_{t-1}[z,x] := 1$
30   **foreach** $w \in \Gamma^{\mathrm{in}}_t\,(x,\mathcal{D})$ **do**
31   **if** $SE_{t-1}[w,z] = 0$ **then**
32                     $SE_{t-1}[w,z] := 1$
33                     $\mathcal{SE} \leftarrow ((t-1,w),(t,z))$
34 **if** $\mathcal{A}$ contains an anchored star **then** $\mathcal{D}$ is copwin
35 **else** $\mathcal{D}$ is robberwin

Theorem 5. *Algorithm* CopRobberPeriodic *determines in time* $O(n^2p + nm)$ *whether or not* $\mathcal{D}$ *is copwin.*

Proof. We first derive the cost of the *Initialization* phase. Observe that the initialization of $\mathcal{A}$, SE, SC (Lines 2–4) can be performed with $O(n^2p)$ operations.

Then, Line 7 is executed n^2p times. The cost of the initialization of DIF and of ϕ (DIF) (Lines 6–13, 18–19), which includes the update of some entries of SC, plus the cost of the initialization of $\mathcal{SE}$ (Lines 14–17), which includes the update of some entries of SE, require at most

$$n^2p + \sum_{i \in \mathbb{Z}_p, u \in V} O(|\Gamma_i(u, \mathcal{D})|) + \sum_{i \in \mathbb{Z}_p, v \in V} O(|\Gamma_i^{in}(v, \mathcal{D})|) = \sum_{i=0}^{p-1} O(n(m_i + m_{i-1})),$$

which sums up to $O(n^2p + nm)$ operations for the *Initialization* phase.

Let us consider now the *Iteration* phase. The while loop will be repeated until in the current augmented arena $\mathcal{A}$ there are no more shadow edges to be examined (i.e. $\mathcal{A} = \mathcal{A}^*$). By Lemma 2, the total number of iterations is $|E(\mathcal{A}^*)| - |E(\mathcal{D})| \leq n^2p - m$. Further observe that every operation performed during an iteration requires constant time.

In each iteration, two processes are being carried out. The first process (Lines 24–27) is the determination of all new shadow corners (if any) of (t, x) created by (the addition of) the shadow edge $((t, x), (t + 1, y))$ being examined. The total cost of this process in this iteration is at most two operations for each in-neighbour of (t, y), i.e., at most $2c_1|\Gamma_t^{in}(y, \mathcal{D})|$, where $c_1 \in O(1)$ is the constant cost of performing a single operation in this process.

This process is repeated in all iterations, each time with a different shadow edge being examined. Thus, the cost of $2c_1|\Gamma_t^{in}(y, \mathcal{D})|$ will be incurred for all $((t, x), (t + 1, y)) \in E(\mathcal{A}^*)$; that is, at most n times. Summarizing, for each $y \in V, t \in \mathbb{Z}_p$ this process costs $2c_1n|\Gamma_t^{in}(y, \mathcal{D})|$. Hence the total cost of this process over all iterations is

$$\sum_{y \in V, t \in \mathbb{Z}_p} 2\ c_1\ n\ |\Gamma_t^{in}(y, \mathcal{D})| = 4\ c_1\ n \sum_{t=0}^{p-1} m_t = O(nm).$$

The second process, to be performed only if new shadow corners of (t, x) have been found in the first process, is the determination (Lines 28–33) of all the new shadow edges (if any) created by the found new shadow corners, and their addition to $\mathcal{SE}$. The cost of this process for a new shadow corner in this iteration is $c_2|\Gamma_t^{in}(x, \mathcal{D})|$, where $c_2 \in O(1)$ is the constant cost of performing a single operation in this process. Observe that, if a new shadow corner of (t, x) is found in this iteration, it will not be considered in any subsequent iteration (Lines 28–29). Hence, the cost $c_2|\Gamma_t^{in}(x, \mathcal{D})|$ will be incurred at most once for each shadow corner of (t, x); that is, at most n times. Summarizing, for each $x \in V, t \in \mathbb{Z}_p$ this process costs at most $2c_2n|\Gamma_t^{in}(x, \mathcal{D})|$. Hence the total cost of this process over all iterations is

$$\sum_{x \in V, t \in \mathbb{Z}_p} 2\ c_2\ n\ |\Gamma_t^{in}(x, \mathcal{D})| = 4\ c_2\ n \sum_{t=0}^{p-1} m_t = O(nm).$$

Consider now the last step of the algorithm, of determining if the constructed $\mathcal{A}$ contains an anchored star. To determine all the stars (if any) in $\mathcal{A}^*$ can be done by checking the degree of each temporal node in $\mathcal{A}^*$, i.e., in $O(np)$ time. To determine if at least one of them is anchored can be done by a DFS traversal of $\mathcal{A}^*$ starting from each root node $(0, x)$, for a total of at most $O(n^2 + nm)$ operations. It follows that the total cost of the algorithm is $O(pn^2 + nm)$ as claimed. □

The bound established by Theorem 5 improves on the existing $O(p\, n^3)$ bound [14]; in particular, in periodic graphs with sparse snapshots the proposed algorithm terminates in $O(p\, n^2)$ time. Furthermore, since a static graph is a periodic graph with $p = 1$, the bound of Theorem 5 becomes $O(n\, m)$, improving the existing $O(n^3)$ bound [32]; in particular our bound becomes $O(n^2)$ for sparse graphs.

4.3 Extensions

Determining a Copwin Strategy. The algorithm, as described, determines whether or not the arena $\mathcal{D}$ (and, thus, the corresponding temporal graph $\mathcal{G}$) is copwin. Simple additions to the algorithm would allow it to easily determine a copwin strategy σ_c if $\mathcal{D}$ is copwin. For any shadow edge $e = ((t, x), (t+1, y)$, let $\rho(t, x, y)$ be defined as follows. If $e = ((t, x), (t+1, y) \in E(\mathcal{D})$, then $\rho(t, x, y) = y$. If $e = ((t, x), (t + 1, y) \in E(\mathcal{A}^*) \setminus E(\mathcal{D})$, when e is inserted in $\mathcal{SE}$, either during the Initialization or the Iteration phase, then $\rho(t, x, y) = z$ where $(t+1, z)$ is the shadow cover of (t, y) determined in the corresponding phase of the algorithm (Line 12 if Initialization, Line 28 if Iteration).

Recall that, if $\mathcal{D}$ is copwin, $\mathcal{A}^*$ must contain an anchored star, say (t, x), Since (t, x) is a star, if the cop is located on (t, x) and the robber is located on (t, y), by moving according to ρ (starting with $\rho(t, x, y)$) the cop will eventually capture the robber. Since (t, x) is anchored, it is reachable from some node in G_0, say $(0, v)$; that is, there is a journey $\pi((0, v), (t, x))$ from $(0, v)$ to (T, x), where $[T]_p = t$. Consider now the following strategy σ_c for the cop: (1) choose as initial location $(0, v)$; (2) follow $\pi((0, v), (t, x))$; (3) follow ρ. Using this strategy, the cop will eventually capture the robber.

More Cops & One Robber. The framework presented so far can be generalized to the case when there are $k > 1$ cops. By shifting from a representation in terms of directed graphs to one in terms of directed multi-hypergraphs, it is possible to extend all the basic concepts introduced for $k = 1$. Indeed, all the fundamental properties of augmented arenas continue to hold in this extended setting, and the same strategy can be used to determine if a periodic graph is k-copwin. The strategy can be implemented by a direct extension of the solution algorithm for $k = 1$.

References

1. Agarwalla, A., Augustine, J., Moses, W., Madhav, S., Sridhar, A.K.: Deterministic dispersion of mobile robots in dynamic rings. In: 19th International Conference on Distributed Computing and Networking, pp. 19:1–19:4 (2018)

2. Aigner, M., Fromme, M.: A game of cops and robbers. Discret. Appl. Math. **8**, 1–12 (1984)
3. Balev, S., Laredo, J.L.J., Lamprou, I., Pigné, Y., Sanlaville, E.: Cops and robbers on dynamic graphs: offline and online case. In: 27th International Colloquium on Structural Information and Communication Complexity (SIROCCO) (2020)
4. Bonato, A., MacGillivray, G.: Characterizations and algorithms for generalized Cops and Robbers games. Contrib. Discret. Math. **12**(1) (2017)
5. Bonato, A., Nowakowski, R.: The Game of Cops and Robbers on Graphs. American Mathematical Society (2011)
6. Bournat, M., Dubois, S., Petit, F.: Computability of perpetual exploration in highly dynamic rings. In: 37th IEEE International Conference on Distributed Computing Systems, pp. 794–804 (2017)
7. Casteigts, A., Flocchini, P., Mans, B., Santoro, N.: Measuring temporal lags in delay-tolerant networks. IEEE Trans. Comput. **63**(2), 397–410 (2014)
8. Casteigts, A., Flocchini, P., Quattrociocchi, W., Santoro, N.: Time-varying graphs and dynamic networks. Int. J. Parallel Emergent Distrib. Syst. **27**(5), 387–408 (2012)
9. Di Luna, G.A.: Mobile agents on dynamic graphs. In: Flocchini, P., Prencipe, G., Santoro, N. (eds.) Distributed Computing by Mobile Entities, Current Research in Moving and Computing. LNCS, vol 11340, pp. 549–584. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11072-7_20. https://dblp.org/rec/series/lncs/Luna19.bib
10. Di Luna, G.A., Dobrev, S., Flocchini, P., Santoro, N.: Distributed exploration of dynamic rings. Distrib. Comput. **33**, 41–67 (2020)
11. Di Luna, G.A., Flocchini, P., Pagli, L., Prencipe, G., Santoro, N., Viglietta, G.: Gathering in dynamic rings. Theor. Comput. Sci. **811**, 79–98 (2020)
12. Erlebach, T., Hoffmann, M., Kammer, F.: On temporal graph exploration. In: 42nd International Colloquium on Automata, Languages, and Programming, pp. 444–455 (2015)
13. Erlebach, T., Spooner, J.T.: Faster exploration of degree-bounded temporal graphs. In: Proceedings of 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS), pp. 1–13 (2018)
14. Erlebach, T., Spooner, J.T.: A game of cops and robbers on graphs with periodic edge-connectivity. In: Chatzigeorgiou, A., et al. (eds.) SOFSEM 2020. LNCS, vol. 12011, pp. 64–75. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-38919-2_6
15. Ferreira, A.: Building a reference combinatorial model for MANETs. IEEE Netw. **18**(5), 24–29 (2004)
16. Flocchini, P., Mans, B., Santoro, N.: On the exploration of time-varying networks. Theor. Comput. Sci. **469**, 53–68 (2013)
17. Gotoh, T., Flocchini, P., Masuzawa, T., Santoro, N.: Exploration of dynamic networks: tight bounds on the number of agents. J. Comput. Syst. Sci. **122**, 1–18 (2021)
18. Gotoh, T., Sudo, Y., Ooshita, F., Kakugawa, H., Masuzawa, T.: Group exploration of dynamic tori. In: Proceedings of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 775–785 (2018)
19. Gromovikov, I., Kinnersley, W., Seamone, B.: Fully active cops and robbers. Australas. J. Comb. **76**(2), 248–265 (2020)
20. Harary, F., Gupta, G.: Dynamic graph models. Math. Comput. Model. **25**(7), 79–88 (1997)

21. Holme, P., Saramäki, J.: Temporal networks. Phys. Rep. **519**(3), 97–125 (2012)
22. Ilcinkas, D., Klasing, R., Wade, A.M.: Exploration of constantly connected dynamic graphs based on cactuses. In: Halldórsson, M.M. (ed.) SIROCCO 2014. LNCS, vol. 8576, pp. 250–262. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09620-9_20
23. Ilcinkas, D., Wade, A.M.: On the power of waiting when exploring public transportation systems. In: Fernàndez Anta, A., Lipari, G., Roy, M. (eds.) OPODIS 2011. LNCS, vol. 7109, pp. 451–464. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25873-2_31
24. Jathar, R., Yadav, V., Gupta, A.: Using periodic contacts for efficient routing in delay tolerant networks. Ad Hoc Sens. Wirel. Netw. **22**(1,2), 283–308 (2014)
25. Khatri, D., et al.: A study of cops and robbers in oriented graphs, pp. 1–23 (2018)
26. Kuhn, F., Lynch, N., Oshman, R.: Distributed computation in dynamic networks. In: 42nd ACM Symposium on Theory of Computing, pp. 513–522 (2010)
27. Loh, P.-S., Oh, S.: Cops and robbers on planar-directed graphs. J. Graph Theory **86**(3), 329–340 (2017)
28. Morawietz, N., Rehs, C., Weller, M.: A timecop's work is harder than you think. In: 45th International Symposium on Mathematical Foundations of Computer Science (MFCS), pp. 71:1–14 (2020)
29. Morawietz, N., Wolf, P.: A timecop's chase around the table. In: 46th International Symposium on Mathematical Foundations of Computer Science (MFCS), pp. 77:1–18 (2021)
30. Nowakowski, R., Winkler, P.: Vertex-to-vertex pursuit in a graph. Discret. Math. **43**(2–3), 235–239 (1983)
31. O'Dell, R., Wattenhofer, R.: Information dissemination in highly dynamic graphs. In: Joint Workshop on Foundations of Mobile Computing, pp. 104–110 (2005)
32. Petr, J., Portier, J., Versteegen, L.: A faster algorithm for cops and robbers. Discret. Appl. Math. **320**, 11–14 (2022)
33. Quilliot, A.: Jeux et points fixes sur les graphs. Ph.D. thesis. Université de Paris VI (1978)
34. Wehmuth, K., Ziviani, A., Fleury, E.: A unifying model for representing time-varying graphs. In: 2nd IEEE International Conference on Data Science and Advanced Analytics, (DSAA), pp. 1–10 (2015)