To Satisfy Impatient Web surfers is Hard

Fedor Fomin, Frédéric Giroire, Alain Jean-Marie, Dorian Mazauric, Nicolas Nisse

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

To Satisfy Impatient Web surfers is Hard

F. V. Fomin — F. Giroire — A. Jean-Marie — D. Mazauric — N. Nisse

N° 7740

September 2011

Domaine

ISSN 0249-6399 ISRN INRIA/RR--7740--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
SOPHIA ANTIPOLIS - MÉDITERRANÉE

To Satisfy Impatient Web surfers is Hard *

F. V. Fomin† , F. Giroire‡ , A. Jean-Marie§ , D. Mazauric‡ , N. Nisse‡

Thème :
Équipe-Projet Mascotte

Rapport de recherche n° 7740 — September 2011 — 21 pages

Abstract: Prefetching is a basic mechanism to avoid to waste time when accessing data. However, a tradeoff must be established between the amount of network's resources wasted by the prefetching and the gain of time. For instance, in the Web, browsers may download documents in advance while a Web user is surfing on the Web. Since the web surfer follows the hyperlinks in an unpredictable way, the choice of the web pages to be prefetched must be computed online. The question is then to determine the minimum amount of resources used by prefetching and that ensures that all documents accessed by the web surfer have previously been loaded in the cache.

We model this problem as a game similar to Cops and Robber Games in graphs. A *fugitive* starts on a marked vertex of a (di)graph G. Turn by turn, an *observer* marks at most $k \geq 1$ vertices and then the fugitive can move along one edge/arcs of G. The observer wins if he prevents the fugitive to reach an unmarked vertex. The fugitive wins otherwise, i.e., if she enters an unmarked vertex. The *surveillance number* of a graph is the least $k \geq 1$ allowing the observer to win whatever the fugitive does. We also consider the connected variant of this game, i.e., when a vertex can be marked only if it is adjacent to an already marked vertex.

All our results hold for both variants, connected or not. We show that deciding whether the surveillance number of a chordal graph equals 2 is NP-hard. Deciding if the surveillance number of a DAG equals 4 is PSPACE-complete. Moreover, computing the surveillance number is NP-hard in split graphs. On the other hand, we provide polynomial time algorithms to compute surveillance number of trees and interval graphs. Moreover, in the case of trees, we establish a combinatorial characterization, related to isoperimetry, of the surveillance number.

Key-words: Prefetching, Cops and robber games, PSPACE-complete, Interval Graphs.

* This work has been done during the visit of Fedor V. Fomin at the INRIA team-project MASCOTTE, INRIA Sophia-Antipolis, France.

† Department of Informatics, University of Bergen, Norway. fomin@ii.uib.no
‡ MASCOTTE, INRIA, I3S(CNRS/Univ. Nice Sophia Antipolis), France. firstname.lastname@inria.fr
§ MAESTRO, INRIA and LIRMM, Univ. Montpellier 2, France. ajm@ilrmm.fr

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Satisfaire un Internaute impatient est difficile

Résumé : Le préchargement est un méchanisme classique qui exploite le parallélisme entre l'exécution d'une tâche et le transfert des informations nécessaires à la prochaine tâche à effectuer. Le préchargement minimise ainsi la perte de temps due à l'attente lorsque l'on veut accéder à des données. Cependant, un compromis doit être établi entre la quantité de ressources du réseau utilisées pour le préchargement (ce qui induit une perte de bande passante par exemple) et le temps effectivement gagné grâce au préchargement. Par exemple, il est intéressant de télécharger des documents à l'avance lorsqu'un Internaute *surfe* sur le Web. Puisque la suite de documents qui vont être examinés (la suite de liens suivis) par l'Internaute n'est pas prévisible, le choix des documents (pages web) à précharger doit être calculer au fur et à mesure que l'Internet évolue sur le Web. La question est alors de déterminer la quantité minimum de ressources utilisées pour le préchargement et qui assure que tous les documents auxquels va accéder l'Internaute auront bien été téléchargés à l'avance. L'Internaute ne doit pas attendre !

Nous modélisons ce problème sous forme d'un jeu de type *Cops and Robber* dans les graphes. Un *fugitif* débute sur un sommet initialement *marqué* d'un graphe (orienté) G. Alors, tour-à-tour un *surveillant* marque au plus k sommets de G et le fugitif peut se déplacer le long d'une arête (d'un arc) de G. Le surveillant gagne si il évite toujours que le fugitif atteigne un sommet non marqué. Le fugitif gagne dans le cas contraire. L'*indice de contrôle* d'un graphe (orienté) G est le plus petit entier $k \geq 1$ qui permet au surveillant de gagner quels que soient les déplacements du fugitif. Nous considérons également la variante *connexe* de ce jeu, dans laquelle, le surveillant ne peut marquer que des voisins de sommets préalablement marqués.

Tous nos résultats sont valides pour les deux variantes (connexe ou non) du jeu. Nous prouvons que décider si l'indice de contrôle d'un graphe cordal vaut 2 est NP-difficile. En particulier, le problème de décision associé à l 'indice de contrôle n'est pas FPT. Puis, nous montrons que calculer l'indice de contrôle est NP-difficile dans la classe des split graphes (une sous-classe des graphes cordaux). Dans le cas des graphes orientés, nous montrons que décider si l'indice de contrôle d'un DAG vaut 4 est PSPACE-complet.

Nous présentons ensuite un algorithme exponentiel exact qui calcule l'indice de contrôle d'un graphe quelconque en temps $O^*(2^n)$. Puis, nous proposons des algorithmes polynomiaux pour le calcul de l'indice de contrôle des arbres et des graphes d'intervalles. Enfin, nous montrons que l'indice de contrôle de tout arbre T est égal à $\max_S \left\lceil \frac{|N[S]|-1}{|S|} \right\rceil$ où S est un sous-arbre de T contenant le sommet de départ, et $N[S]$ est le voisinage fermé de S.

Nous concluons avec diverses questions ouvertes.

Mots-clés : Précharchement, jeux de "Cops and Robber", PSPACE-complet, graphes d'intervalles.

1 Introduction

Prefetching is a basic technique in computer science. It exploits the parallelism between the execution of one task and the transfer of information necessary to the next task, in order to reduce waiting times. The classical instance of the problem occurs in CPU, where instructions and data are prefetched from the memory while previous instructions are executed. The modern instance occurs in the Web, where browsers may download documents connected to the currently viewed document (web page, video, etc.) while it is being read or viewed. Accessing the next document appears to be instantaneous to the user, and gives the impression of a large navigation speed [phd]. For this reason, link prefetching has been proposed as a draft Internet standard by Mozilla [reference?]. However, prefetching all documents that can be accessed in the current state may exceed networking capacities, or at least, result in a waste of bandwidth since most of the alternatives will not be used. Hence, it is necessary to balance the gain of time against the waste of networking resources. Local storage memory is also a potential issue, and prefetching is classically associated with the question of cache management. However, memory in modern computers is not scarce anymore, which makes network resources the critical ones.

The models developed so far in the literature to study prefetching problems are based on the *execution digraph* where the nodes represent the tasks (e.g., web pages) and arcs model the fact that a task can be executed once another has been done (e.g., arcs represent hyperlinks that can be followed from a web page). The execution of the program or the surfing of the web then corresponds to a path in the execution digraph. The quantitative optimization of prefetching will then be based on some cost function defined on paths, reflecting for instance the inconvenience of waiting for some information while executing the tasks or surfing the web, and possibly taking into account the consumption of network or memory resources. The related dimensioning problem consists in determining how much network bandwidth should be available so that the prefetching performance stays within some predetermined range.

It is quite likely that such optimization problems are very difficult to solve exactly. For instance, in *Markovian* models [JG97], where arcs of the execution digraph are associated with transition probabilities (modeling a random web surfer), the prefetching problem can then be cast as an optimization problem in the Stochastic Dynamic Programming framework [GCD02,MJM10]. Its exact solution requires a computational effort which is exponential with respect to the number of nodes in the execution digraph: this is the size of the state space of these Markov Decision models.

As a first step in the analysis of prefetching optimization, we therefore consider a the following simpler problem. We consider a surfer evolving over the execution digraph, and we are concerned with *perfect* prefetching, i.e., ensuring that the web surfer never accesses an document that has not been prefetched yet. In other words, the surfer is "impatient" in the sense that she does not tolerate waiting for information. Due to network's capacity (bandwidth) limitation, it is important to limit the number of web pages that can be prefetched at each step. We aim at determining the minimum amount of web pages to be prefetched at each step. In addition to being simpler than a fully specified optimization problem, this question does not need specific assumptions on the behavior of the web surfer as in [GCD02, MJM10].

Given an execution digraph D and a node $v_0 \in V(D)$ corresponding to the web page from which the surfer starts, the *surveillance number* of D starting in v_0 is the least number of web pages prefetched at each step that avoid the web surfer to wait (whatever the surfer does).

1.1 Our results

We model the above prefetching problem as a Cop and Robber game (e.g., see [NW83,Qui83,FT08,Als04]). Using this framework, we prove that deciding whether the surveillance number of a chordal graph equals 2 is NP-hard. In particular, this shows that the decision problem associated to surveillance number is not Fixed Parameter Tractable. Then, we show that computing the surveillance number is NP-hard in split graphs, a subclass of chordal graphs. In the case of digraphs, we show that deciding if the surveillance number of a DAG equals 4 is PSPACE-complete.

On the other hand, we provide polynomial time algorithms that compute the surveillance number s and a corresponding optimal strategy in trees and interval graphs. Moreover, in the case of trees, we establish a combinatorial characterization, related to isoperimetry, of the surveillance number. That is, we show that the surveillance number of a tree T starting in $v_0 \in V(T)$ equals $\max_S \left\lceil \frac{|N[S]|-1}{|S|} \right\rceil$ where S is taken among all subtrees of T containing v_0 and $N[S]$ denotes the closed neighborhood of S. We conclude with several open questions.

1.2 Cops and Robber games

Two players turn-by-turn games in graphs are classically referred to as *Cops and Robber games*. In the initial variant of these games [NW83, Qui83], one cop is placed at a vertex of a graph, then the robber chooses one vertex to go, and then the players move their token along edges of the graph, alternatively starting with the cop. The cop wins if at some step of the game it occupies the same vertex as the robber. In [AF84], the Cop-Player is allowed to use a team of $k \geq 1$ cops. One optimization problem is then to decide the *cop-number* of a graph G, i.e., the minimum number of cops that are required to capture the robber in G. It is known to be W[2]-hard in general [GR95, FGK$^+$10]. Lower and upper bounds on the cop-number of various classes of graphs have been proved [And86, Fra87, Sch01, BKL08].

Several variants have been studied such as when the cops and the robber have different speeds [FGK$^+$10, CCNV11, AM10, Meh11], when the robber can be captured at some distance [BCP10], when each cop can be moved a bounded number of time [FGL10], etc. In the variant proposed in [FGH$^+$08, FGL09], the goal for the cops is to *guard* some part of a graph, i.e., to prevent the robber to reach some particular vertices in the graph. Eternal dominating set and eternal vertex cover can also be viewed as cops and robber games where the robber has no token but can *attack* a vertex, resp., an edge, at each step and the cop must move its tokens in response to the attack [GK08, FGG$^+$10].

2 Preliminaries

In this section, we formally define the problems we consider and present the notations used throughout the paper. We also present some basic results.

For any (di)graph $G = (V,E)$ considered in this paper, when $v_0 \in V$ is fixed as the starting vertex, we assume that, for any $v \in V$, there is a (directed) path from v_0 to v. In particular, if G is an undirected graph, we assume that G is connected.

Let $\Delta(G)$ be the maximum degree of G (we denote it by Δ when no ambiguity occurs). If G is a digraph, we denote by $\Delta^+(G)$ the maximum out-degree.

For any undirected graph $G = (V,E)$ and any $S \subseteq V$, let $G[S]$ be the subgraph induced by S in G. The *open neighbourhood* $N(S)$ of a vertex subset S is the subset of vertices in $V \setminus S$ having a neighbour in S and the *closed neighbourhood* is $N[S] = N(S) \cup S$. If $S = \{v\}$, we use $N(v)$ and $N[v]$ instead of $N(\{v\})$ and $N[\{v\}]$.

A (di)rected graph is a tree (resp., a Directed Acyclic Graph, or *DAG*) if it as no (directed) cycle as a subgraph. A graph is *chordal* if it does not contain induced cycles of length more than 3. A graph $G = (V,E)$ is a *split graph* if there is a partition (A,B) of V such that A induces a clique and B induces an independent set. Finally, G is an *interval graph* if V is a set of real intervals and two vertices are adjacent if the corresponding intervals intersect.

2.1 The game

Let $G = (V,E)$ be an n-node (di)graph and let $v_0 \in V$ be a particular vertex of it, the *starting vertex*, which is initially *marked*. Let $k \geq 1$.

The *surveillance problem* deals with the following two players game where a fugitive wants to escape to the control of an observer whose purpose is to keep the fugitive under constant surveillance. There are two players,

fugitive and *observer*. The fugitive wants to escape the control of an observer whose purpose is to keep the fugitive under constant surveillance. Let $k \geq 1$ be a fixed integer. The game starts when the fugitive stands at v_0 which is initially marked. Then, turn by turn, the observer controls, or *marks*, at most k vertices and then the fugitive either moves along an edge to a (out-)neighbor of her current position, or skip her move. In other words, at every step of the game the observer enlarges observable part of the graph by adding to it k, not necessarily adjacent, vertices. His task is to ensure that the fugitive is always in the observable area. Note that, once a vertex has been marked, it remains marked until the end of the game. The fugitive wins if, at some step, she The fugitive wins if, at some step, she reaches an unmarked vertex and the observer wins otherwise. In other words, the game ends when either the fugitive enters an unmarked vertex (and then she wins) or all vertices have been marked (and then observer wins).

More formally, a *k-strategy* (for the observer) is a function σ that assigns a subset $S \subseteq V$, $|S| \leq k$, to any *configuration* (M, f) of the game where $M \subseteq V$ is the set of the vertices that have already marked before this step of the game, $f \in M$ is the current position of the fugitive, and $S = \sigma(M, f)$ is the set of vertices to be marked at this step. Clearly, we can restrict our investigation to the case where $\sigma(M, f) \subset V \setminus M$ and $|\sigma(M, f)| = k$ or $\sigma(M, f) = V \setminus M$. That is, at each step, the observer has interest to mark as many unmarked vertices as possible. In particular, a game consists of at most $\lceil n/k \rceil$ steps. A k-strategy is *winning* if it allows the observer to win whatever be the walk followed by the fugitive. Note that any winning strategy must ensure that $N(f) \setminus M \subseteq \sigma(M, f)$ for any $M \subseteq V$ and $f \in M$. Finally, the *surveillance number* of G, denoted by $sn(G, v_0)$, is the smallest k such that there is a winning k-strategy in G starting from v_0.

2.2 Connectivity and Bounds

In this section, we define some variants of the game by introducing new natural constraints and prove basic results.

In the *connected* variant of the surveillance game, the observer is constraint to mark only vertices that have neighbors already marked, i.e., the set of marked vertices must always induce a connected subgraph. That is, a connected strategy σ is a strategy with the additional constraint that $\sigma(M, f) \cup M$ must induce a connected subgraph for any connected subset $M \subseteq V$ containing v_0. Note that it is not required that $\sigma(M, f)$ induces a connected subgraph. Let $csn(G, v_0)$ be the smallest k such that there is a winning connected k-strategy in G when the fugitive starts from v_0.

We first show that imposing the connectedness of a strategy is a strong constraint.

Lemma 1. *Let $k \geq 2$. There exist a graph G and $v_0 \in V$ such that $csn(G, v_0) > sn(G, v_0) = k$.*

Proof. Let $k \geq 2$. Let G be the graph with $6k$ vertices: a path (v_0, v_1, v_2) then $2k$ vertices a_i and b_i, $1 \leq i \leq k$, such that a_i adjacent to v_2 and b_i, and finally a set K of $4k-3$ vertices each of which is adjacent to all b_i, $i \leq k$. Then $k = sn(G, v_0) < csn(G, v_0) = k+1$.

Indeed, the following strategy is winning: at each step, mark the $i \geq 0$ unmarked neighbors of the current position of the fugitive, and then mark $k-i$ vertices in K. Hence $sn(G, v_0) \leq k$. On the other hand, in the connected variant, at least 4 vertices, say $\{v_1, v_2, a_1, b_1\}$, must be marked before any vertex in K is marked. The fugitive first goes to v_1 and v_2. Then if a vertex a_i is unmarked, she goes to it and wins. Otherwise, she goes to a_2 and then b_2. When it is the fifth turn of the fugitive, at least $k+4$ vertices not in K must have been marked. Then, when at most k vertices can be marked per step, at most $4k-4$ vertices of K have been marked, so the fugitive can win. It is easy to show that $csn(G, v_0) \leq k+1$ and that $sn(G, v_0) > k-1$. □

Question 1. *Does there exist a constant bounding the ratio (resp., the difference) between csn and sn in any graph?*

The surveillance number of a graph is clearly constrained by the degrees of its vertices. More precisely:

Claim 1. *For any (di)graph G with maximum (out-)degree $\Delta^{(+)}$ and for any v_0 with (out-) degree $deg^{(+)}(v_0)$, $deg^{(+)}(v_0) \leq sn(G, v_0) \leq csn(G, v_0) \leq \Delta^{(+)}$. Moreover, in undirected graphs, $csn(G, v_0) = \Delta$ iff v_0 has degree Δ.*

Proof. Clearly $sn(G,v_0) \geq deg^{(+)}(v_0)$ and by definition $sn(G,v_0) \leq csn(G,v_0)$. On the other hand, the following strategy is clearly winning. At each step, the observer simply marks all unmarked (out-)neighbors of the current position of the fugitive. Hence, $csn(G,v_0) \leq \Delta^{(+)}$. Moreover, in the undirected case, the fugitive always arrives to any vertex (but v_0) by an already marked neighbor. Hence, following the previous strategy, the observer marks at most $\Delta - 1$ vertices at each steps but the first one. So, if $deg^{(+)}(v_0) < \Delta$ then we get that $csn(G,v_0) < \Delta$. □

Next lemma is straigthforward following the previous claim.

Lemma 2. *Let G be a connected undirected graph with maximum degree $\Delta \leq 3$ and at least one edge. Then, $1 \leq csn(G,v_0) = sn(G,v_0) \leq 3$ and*

- *$csn(G,v_0) = sn(G,v_0) = 1$ iff G is a path where v_0 has degree one;*
- *$csn(G,v_0) = sn(G,v_0) = 3$ iff v_0 has degree 3.*

So, computing the surveillance number of a graph with maximum degree at most 3 is trivial.

Question 2. *What is the complexity of computing the surveillance number in the class of graphs with maximum degree 4? with bounded degree?*

The proof of the following lemma is straitforward.

Lemma 3. *Let G be an undirected graph with a universal vertex. For any $v_0 \in V(G)$, $sn(G,v_0) = csn(G,v_0) = \max\{deg(v_0), \lceil \frac{n-1}{2} \rceil\}$.*

2.3 To restrict the fugitive to induced paths does not help the observer

Finally, we define a restriction of the game that will be useful throughout this paper.

In the *monotone* variant of the surveillance game, the fugitive is restricted to move at every step and to follow only induced paths in G. That is, for any $\ell > 0$, after having followed a path $(v_0, \cdots, v_\ell)$, the fugitive is not allowed reaching a vertex in $N[\{v_0, \cdots, v_{\ell-1}\}]$ anymore. Let $msn(G,v_0)$ be the smallest k such that there is a winning monotone k-strategy in G when the fugitive starts from v_0, i.e., the observer can win, marking at most k vertics at each step, against a fugitive constrained to follow induced paths.

The monotone game is interesting since it is easier to study. Indeed, we now prove that "monotonicity does not help", that is, for any graph G and $v_0 \in V(G)$, $msn(G,v_0) = sn(G,v_0)$. In other words, if the fugitive follows only induced paths, the observer needs to mark the same amount of vertices at each step as he does when the fugitive has no restriction. This means that in the following proofs, we can always consider that the fugitive follows induced paths, and in particular, the fugitive has to move at every step.

To prove the announced result, we give an alternative definition of a winning strategy in terms of decomposition of graphs. Recall that a strategy is defined by a function $\sigma : 2^V \times V \to 2^V$ where $|\sigma(M,f)| \leq k$ for any $M \subseteq V, f \in V$ and $\sigma(M,f)$ represents the set of vertices that must be marked when the fugitive is in f and the vertices in M have already been marked. Clearly, such a strategy can be viewed as a *decision-tree* where each vertex of this decision tree represents a path that have been followed by the fugitive.

We first describe a tree-structure to represent the paths of G, starting from v_0. An *internal vertex* of a rooted tree is a vertex with at least one child, other vertices are called the *leaves*. In what follows, $N^{(+)}[X]$ denotes the closed neighborhood of a set of vertices in a graph, respectively, the close out-neighborhood of a set of vertices in a digraph.

Definition 1. *let G be a (di)graph and $v_0 \in V(G)$. A* path-tree *is a pair (T,ω) where T is a tree rooted in $r \in V(T)$ and $\omega : V(T) \to V(G)$ such that $\omega(r) = v_0$ and any internal vertex $t \in V(T)$ has $\ell = |N^{(+)}[\omega(t)]|$ children $\{t_1, \cdots, t_\ell\}$ with $\{\omega(t_1), \cdots, \omega(t_\ell)\} = N^{(+)}[\omega(t)]$.*

In that way, any vertex $t_i \in V(T)$ ($i \geq 0$) where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T represents the walk $P_{t_i} = (v_0 = \omega(r), \omega(t_1), \cdots, \omega(t_i))$ in G. The next structure restricts the paths we want to represent to the induced paths of G starting from v_0.

Definition 2. *let G be a (di)graph and $v_0 \in V(G)$. An* induced path-tree *is a pair (T, ω) where T is a tree rooted in r and $\omega : V(T) \to V(G)$ such that $\omega(r) = v_0$ and, for any internal vertex $t_i \in V(T)$ where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T and $N = N_G^{(+)}(\omega(t_i)) \setminus N_G^{(+)}[\{t_0, t_1, \cdots, t_{i-1}\}])$, then t_i has $\ell = |N|$ children $\{u_1, \cdots, u_\ell\}$ with $\{\omega(u_1), \cdots, \omega(u_\ell)\} = N$.*

Definition 3. *A* (monotone) k-decision-tree *(k-DT) rooted in v_0 of a (di)graph G is a triple (T, ω, M) defined as follows. (T, ω) is a (induced) path-tree rooted in r and $M : V(T) \to 2^V$ and the following properties are satisfied. $v_0 \in M(r)$ and, for any vertex $t_i \in V(T)$ where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T,*

- *$t_i \in V(T)$, $|M(t_i) \setminus \{v_0\}| \leq k$;*
- *for any child t of t_i, $w(t) \in \cup_{j \leq i} M(t_j)$;*
- *t_i is a leaf iff $\cup_{j \leq i} M(t_j) = V(G)$.*

The (induced) path-tree allows to represent all walks (induced paths) starting in v_0 in G. Namely, given $t_i \in V(T)$ with $P = (r, t_1, \cdots, t_i)$ the path in T from r to t_i, t_i represents the (induced) path $P_{t_i} = (v_0 = \omega(r), \omega(t_1), \cdots, \omega(t_i))$ in G. Moreover, for any $t \in V(T)$, the *bag* $M(t)$ represents the subset of vertices that must be marked at the step after the fugitive has followed the path P_t in G. By the properties stated in Definition 3, no more than k vertices are marked at each step and no path in G may allow the fugitive to avoid marked vertices.

Decision-trees should be more constrained to express that it is useless to mark several times the same vertex or not to mark the maximum number of vertices at each step.

Definition 4. *A (monotone) k-decision-tree (T, ω, M) is said* refined *if*

- *for any internal vertex $t \in V(T)$, $|M(t) \setminus \{v_0\}| = k$;*
- *for any vertex $t_i \in V(T)$ where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T, $M(t_i) \in V(G) \setminus \cup_{j<i} M(t_j)$.*

Note that a refined k-DT of n-node graph G has height at most $\lceil \frac{n-1}{k} \rceil$.

Lemma 4. *If G admits a (monotone) k-decision-tree rooted in v_0, then G admits a (monotone) refined k-decision-tree rooted in v_0.*

Proof. Let (T, ω, M) be a (monotone) k-DT rooted in v_0. Let $t_i \in V(T)$ closest from r the root of T, where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T, that does not satisfy the conditions of a refined k-DT. If $M(t_i) \cap \cup_{j<i} M(t_j) \neq \emptyset$, then replace $M(t_i)$ with $M(t_i) \setminus \cup_{j<i} M(t_j)$. Otherwise, let $v \in V(G) \setminus \cup_{j \leq i} M(t_j)$ and replace $M(t_i)$ with $M(t_i) \cup \{v\}$. Finally, if $\cup_{j \leq i} M(t_j) = V(G)$, remove from T all subtrees rooted in a child of t_i. Clearly, such a process ends with a (monotone) refined k-decision-tree rooted in v_0. □

Lemma 5. *A (di)graph G admits a (resp., monotone) k-decision-tree rooted in v_0 iff $sn(G, v_0) \leq k$ (resp., iff $msn(G, v_0) \leq k$).*

Proof. Assume $sn(G, v_0) \leq k$ and let σ be a k-strategy such that $\sigma(M, f) \subseteq V(G) \setminus M$, and $|\sigma(M, f)| = k$ or $M \cup \sigma(M, f) = V(G)$ for any $M \subseteq V(G)$, $f \in M$. Let (T, ω) satisfying the first condition of Definition 3 and T of height $\left\lceil \frac{|V(G)|-1}{k} \right\rceil$. Then, for any vertex $t_i \in V(T)$ where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T, let us define $M(t_i) = \sigma(\cup_{j<i} M(t_j), \omega(t_i))$. Then, (T, ω, M) is a k-DT rooted in v_0.

Let (T, ω, M) be a k-DT rooted in v_0. Let $\sigma : 2^V \times V \to 2^V$ be any application satisfying that, for any $t_i \in V(T)$ where $(r = t_0, t_1, \cdots, t_i)$ is the path from r to t_i in T, $\sigma(\cup_{j<i} M(t_j), \omega(t_i)) = M(t_i)$. Then, σ is a winning k-strategy. □

Now, we can prove the main result of this section.

Theorem 3. *For any (di)graph G and $v_0 \in V(G)$, $sn(G,v_0) = msn(G,v_0)$.*

Proof. If $msn(G,v_0) \leq k$, by Lemma 5, there is a monotone k-DT of G rooted in v_0. By Lemma 4, there is a refined monotone k-DT of G rooted in v_0. We show that a k-DT rooted in v_0 of G can be built from any monotone k-DT (T,ω,M) of G rooted in v_0.

Let (T,ω,M) be a refined monotone k-DT of G rooted in v_0. Since (T,ω,M) is not a k-DT, this means that (T,ω) is an induced path-tree and not a path-tree. In other words, there is $t \in V(T)$ with children $(u_1,\cdots,u_\ell)$ such that there is $y \in N_G[\omega(t)]$ and, for any $i \leq \ell$, $\omega(u_i) \neq y$. We say that such a vertex t satisfies property $\mathcal{P}$.

Let t be a vertex, closest to r, that satisfies this property $\mathcal{P}$. Two cases are to be considered according to whether $y = \omega(t)$ of whether $y \in N_G(\omega(t))$.

- Assume first $y = \omega(t)$. Let S be the subtree of T rooted in t. We transform (T,ω,M) into (T',ω',M') by adding the sub-decision-tree "induced" by S as a child of t in T. That is T' is obtained from T by adding a copy of S with its root adjacent to t in T. Then, for any $z \in V(T') = V(T) \cup V(S)$, $\omega'(z) = \omega(z)$ and $M'(z) = M(z)$. Then, let (T^*,ω^*,M^*) be obtained by refining the obtained decomposition, i.e., applying to (T',ω',M') the process described into the proof of Lemma 5.

- Now, assume that $y \in N_G(\omega(t))$. Let $(r = t_0,t_1,\cdots,t_i = t)$ be the path from r to t in T. Since $y \notin \{\omega(u_j) \ : \ j \leq \ell\}$, by definition of monotone decision-trees, it means that there is t_j, $j < i$ such that t_i has a child $s \neq t_{j+1}$ and $\omega(s) = y$. Let S be the subtree of T rooted in s. We transform (T,ω,M) by adding the sub-decision-tree "induced" by S as a child of t in T. Then, we refine the obtained decision-tree.

The process consists in reiterating the same transformation while there is a node t of the obtained decomposition that satisfies property $\mathcal{P}$. Note that, while there still are some vertices satisfying $\mathcal{P}$, the obtained decomposition is neither a decision-tree nor a monotone decision-tree since the tree T^* of the obtained decomposition is neither a path-tree nor an induced path-tree (actually, it is "between" a path-tree and an induced path-tree). However, all other properties of a decision-tree remain satisfied.

We finally show that a finite number of such transformations is sufficient to obtain a decision-tree. Indeed, it is is sufficient to remark that the following "potential function" Φ strictly decreases each time the transformation is applied (recall that Δ is the maximum degree of G). Moreover, the size of the obtained decomposition remains bounded in the size of the initial monotone decision-tree (T,ω,M) since we consider only refined decompositions.

Let $\Phi(T,\omega,M)$ be the sum of $\phi(i_t)(|N_G^{(+)}[\omega(t)]| - |\{w(s) \ : \ s \ child \ of \ t \ in \ T\}|)$ over all vertices t in $V(T)$, where i_t is the distance between t and r and ϕ is any function such that $\phi(i) > \Delta^{\lceil \frac{n-1}{k} \rceil - i + 1}\phi(i+1)$.

Each time we apply the procedure on a node $t \in V(T)$ at distance i of the root, we add one child to t. Hence, $(|N_G^{(+)}[\omega(t)]| - |\{w(s) \ : \ s \ child \ of \ t \ in \ T\}|)$ is decreased by one and this contributes to decrease $\Phi(T,\omega,M)$ by $\Phi(i)$. However, as "child" of t, we add a subtree S with height at most $\lceil \frac{n-1}{k} \rceil - i$ (because the decomposition is refined). Therefore, S may have at most $\Delta^{\lceil \frac{n-1}{k} \rceil - i}$ vertices and the contribution of each vertex is at most $\Delta \cdot \phi(i+1)$. This contributes to increase the global sum $\Phi(T,\omega,M)$ of at most $\Delta^{\lceil \frac{n-1}{k} \rceil - i + 1}\phi(i+1)$. Since $\phi(i) > \Delta^{\lceil \frac{n-1}{k} \rceil - i + 1}\phi(i+1)$, the global sum $\Phi(T,\omega,M)$ strictly decreases at each step. □

3 Difficult problems

In this section, we study the computational complexity of the decision version of the problem: given a graph G with $v_0 \in V(G)$ and an integer k, the task is to decide whether $sn(G,v_0) \leq k$. We also consider the variant of the problem where the fugitive must win in a fixed number of steps. Moreover, for all graphs we consider in our reductions, $sn(G,v_0) = csn(G,v_0)$. Hence, our hardness results also apply to the connected variant of the problem.

We use a reduction from the 3-Hitting Set Problem. In the 3-Hitting Set Problem, we are given a set I of elements, a set $\mathcal{S}$ of subsets of size 3 of I and an integer k as an input. The question is to decide whether there exists a set $H \subseteq I$ of size at most k such that $H \cap S \neq \emptyset$ for all $S \in \mathcal{S}$. The 3-Hitting Set Problem is the classical NP-complete problem [GJ90].

We start with the proof that the problem is NP-hard on chordal graphs. Let us remind that a graph is *chordal* if it contains no induced cycle of length at least 4.

Theorem 4. *Deciding if $sn(G, v_0) \leq 2$ (resp., $csn(G, v_0) \leq 2$) is NP-hard in chordal graphs.*

Proof. Let $(I = \{e_1, \cdots, e_n\}, \mathcal{S} = \{S_1, \cdots, S_m\})$ and $k \geq 1$ be an instance of the 3-Hitting Set Problem. We construct the chordal graph G as follows. Let $P = \{v_0, \cdots, v_{m+k-2}\}$ be a path, K_m be the complete graph with vertices $\{S_1, \cdots, S_m\}$ and $e_1, \cdots, e_n$ be n isolated vertices. We add an edge from v_{m+k-2} to all vertices of K_m and, for any $i \leq n$ and $j \leq m$, add an edge between e_i and S_j if and only if $e_i \in S_j$. Clearly, G is chordal.

First, we show that, if there exists a set $H \subseteq I$ of size k such that $H \cap S \neq \emptyset$ for all $S \in \mathcal{S}$, then $csn(G, v_0) \leq 2$. The 2-strategy of the observer first consists in marking the vertices v_1 to v_{m+k-2} in order, then the vertices of K_m and finally the vertices of H. This can be done in $m+k-1$ steps and in such a way that, at each step, all neighbors of the current position of the fugitive are marked. Because H is a hitting set of $\mathcal{S}$, after the $(m+k-1)^{th}$ step, each vertex S_i, $i \leq m$, has at most two unmarked neighbors, all other vertices have all their neighbors marked and only some vertices in $e_1, \cdots, e_n$ can be unmarked. Finally, from this step, the strategy of the observer consists in marking the unmarked neighbors of the current position of the fugitive. Clearly, the fugitive cannot win and, thus, there exists a connected winning 2-strategy.

Now, assume that, for any $H \subseteq I$ of size at most k, there is $S \in \mathcal{S}$ such that $S \cap H = \emptyset$. The escape strategy for the fugitive first consists in going to v_{m+k-2} (this takes $m+k-2$ steps). Then, after the $(m+k-1)^{th}$ step of the observer, all vertices of P and K_m are marked –otherwise the fugitive would have won earlier or can go to a vertex of K_m that is still unmarked. It means that the subset H of vertices among $e_1, \cdots, e_n$ that are marked at this step is of size at most k. Hence, when it is the turn of the fugitive who is occupying v_{m+k-2}, there is $S_i \in V(K_m)$ with $H \cap S_i = \emptyset$, i.e., all the three neighbors of S_i are unmarked. Then, the fugitive goes to S_i. The observer can mark at most two of the neighbors of S_i, and the fugitive can reach an unmarked vertex. Hence, $sn(G, v_0) > 2$. □

Let $\ell \geq 1$. We define a restriction of the game where the fugitive wins if she reaches an unmarked vertex in at most ℓ steps. Let $sn(G, v_0, \ell)$ (resp., $csn(G, v_0, \ell)$) be the smallest k such that there is a (connected) winning k-strategy in G against a fugitive starting from v_0 in this setting. Note that, if both k and ℓ are fixed, then $sn(G, v_0, \ell) \leq k$ (resp., $csn(G, v_0, \ell) \leq k$) can be decided in polynomial time since the number of trajectories of the fugitive is at most n^ℓ and the number of strategies for the observer is bounded by $n^{O(f(k,\ell))}$. However:

Theorem 5. *If $k \geq 1$ is part of the input, the problem of deciding whether $sn(G, v_0, 2) \leq k$ (resp., $csn(G, v_0, 2) \leq k$) is NP-hard in split graphs.*

Let G be a DAG with maximum out-degree Δ^+ and v_0 with degree $< \Delta^+$ (Δ^+ part of the input). The problem of deciding whether $sn(G, v_0, 2) = \Delta^+ - 1$ (resp., $csn(G, v_0, 2) = \Delta^+ - 1$) is NP-hard.

Proof. Again, we reduce this problem to the 3-Hitting Set Problem. Let $(I = \{x_1, \cdots, x_n\}, \mathcal{S} = \{S_1, \cdots, S_m\})$ and $k \geq 1$ be an instance of the 3-Hitting Set Problem.

Let us build the split graph G described in Figure 1. Let K_{m+1} be the complete graph with vertices $\{v_0, S_1, \cdots, S_m\}$ and let $\{x_1, \cdots, x_n\}$ be n isolated vertices. For any $i \leq m$, add $m+k-2$ extra leaves adjacent to S_i and add an edge between x_j and S_i if $x_j \in S_i$ for all $j \leq n$.

As in the proof of Theorem 4, we prove that $sn(G, v_0, 2) \leq k+m$ (and $csn(G, v_0, 2) \leq k+m$) if and only if $(I = \{x_1, \cdots, x_n\}, \mathcal{S} = \{S_1, \cdots, S_m\})$ admits a hitting set of size at most k.

Indeed, if there is a hitting set H of size k, then the observer, allowed to mark $k+m$ vertices per step, first marks all vertices of K_{m+1} (but v_0 that is already marked) and all vertices of H. Then, at each step, the observer marks the unmarked neighbors of the current position of the fugitive.

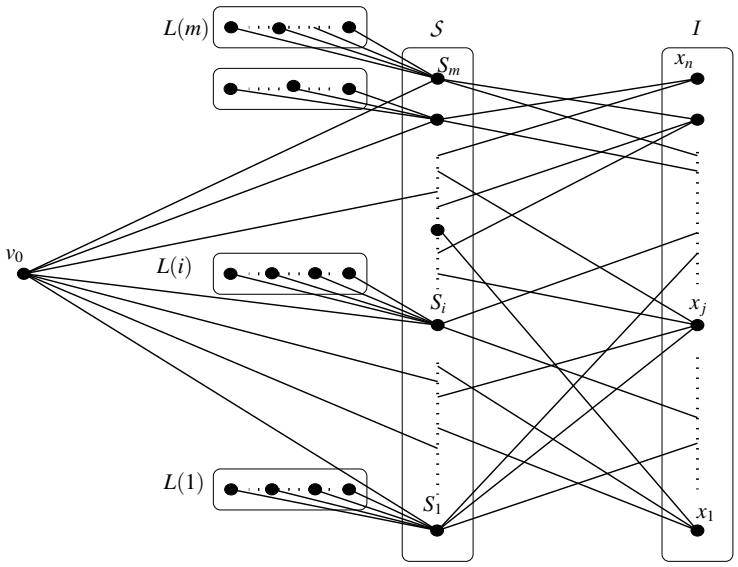

Figure 1: Example of graph G in the reduction of the proof of Theorem 5. $L(j)$, $j \leq m$, represents the set of $m+k-2$ leaves adjacent to S_j.

Otherwise, assume that $\mathcal{S}$ admits no hitting set of size at most k and let us assume that the observer can mark at most $m+k$ vertices. The first move of the fugitive consists in going toward a vertex S_j ($1 \leq j \leq m$) with at least $m+k+1$ unmarked neighbors. Then, the fugitive wins after her second move.

This concludes the proof of the first statement of the Theorem.

For the second statement, let G' be the DAG obtained by taking the underlying graph G and removing all edges between S_i and S_j, $1 \leq i,j \leq m$. Then, orient the edges in such a way that S_i, for any $i \leq m$, has in-degree 1, corresponding to the edge that comes from v_0. G' has maximum out-degree $m+k+2$. By Claim 1, $sn(G', v_0, 2) < m+k+2$. Finally, the same proof as above shows that $csn(G', v_0, 2) = sn(G', v_0, 2) \leq k+m$ if and only if $(I = \{x_1, \cdots, x_n\}, \mathcal{S} = \{S_1, \cdots, S_m\})$ admits a hitting set of size at most k. □

In the following, we reduce our problem to the 3-QSAT problem. For a set of boolean variables $x_0, y_0, x_1, y_1, \cdots, x_n, y_n$ and a boolean formula $F = C_1 \wedge \cdots \wedge C_m$, where C_j is a clause with 3 variables, the 3-QSAT problem aims at deciding whether the expression $\Phi = \forall x_0 \exists y_0 \forall x_1 \exists y_1 \cdots \forall x_n \exists y_n F$ is true. 3-QSAT is PSPACE-complete [GJ90].

Lemma 6. *The problem of deciding whether $sn(G, v_0) \leq 4$ (resp., $csn(G, v_0) \leq 4$) is PSPACE-hard in DAGs.*

Proof. Let $F = C_1 \wedge \cdots \wedge C_m$ be a boolean formula with variables $x_0, y_0, x_1, y_1, \cdots, x_n, y_n$ and $\Phi = \forall x_0 \exists y_0 \forall x_1 \exists y_1 \cdots \forall x_n \exists y_n F$ be an instance of the 3-QSAT Problem. Let D be the DAG built as follows.

We start with the set of vertices $\{u_i, v_i, x'_i, \bar{x}'_i, x_i, \bar{x}_i, y'_i, \bar{y}'_i, y_i, \bar{y}_i\}_{0 \leq i \leq n}$. For any $0 \leq i \leq n$, there are arcs from v_i to x'_i and $\bar{x}'_i$, one arc from x'_i to x_i and one arc from $\bar{x}'_i$ to $\bar{x}_i$. For any $0 \leq i \leq n$, there are arcs from x'_i and $\bar{x}'_i$ to u_i, arcs from u_i to both y'_i and $\bar{y}'_i$ and arcs from both of y'_i and $\bar{y}'_i$ to both of y_i and $\bar{y}_i$. Then, for any $0 \leq i < n$, there is one arc from u_i to v_{i+1}. Add the directed path $(w_1, \cdots, w_{m-1})$ with one arc from u_n to w_1 and such that w_{m-1} has m out-neighbors $C_1, \cdots, C_m$. For any $j \leq m$ and $0 \leq i \leq n$, add one arc from C_j to x_i (resp., $\bar{x}_i, y_i, \bar{y}_i$) if x_i (resp., $\bar{x}_i, y_i, \bar{y}_i$) appears in the clause C_j. Finally, for any $0 \leq i \leq n$, $k \leq m-1$, $j \leq m$ add two out-neighbors leaves to each vertex in $\{v_i, x'_i, \bar{x}'_i, w_k, C_j\}$, and, for any $0 \leq i \leq n$, add three out-neighbors leaves to each of y'_i and $\bar{y}'_i$. An example of such DAG D is depicted in Figure 2.

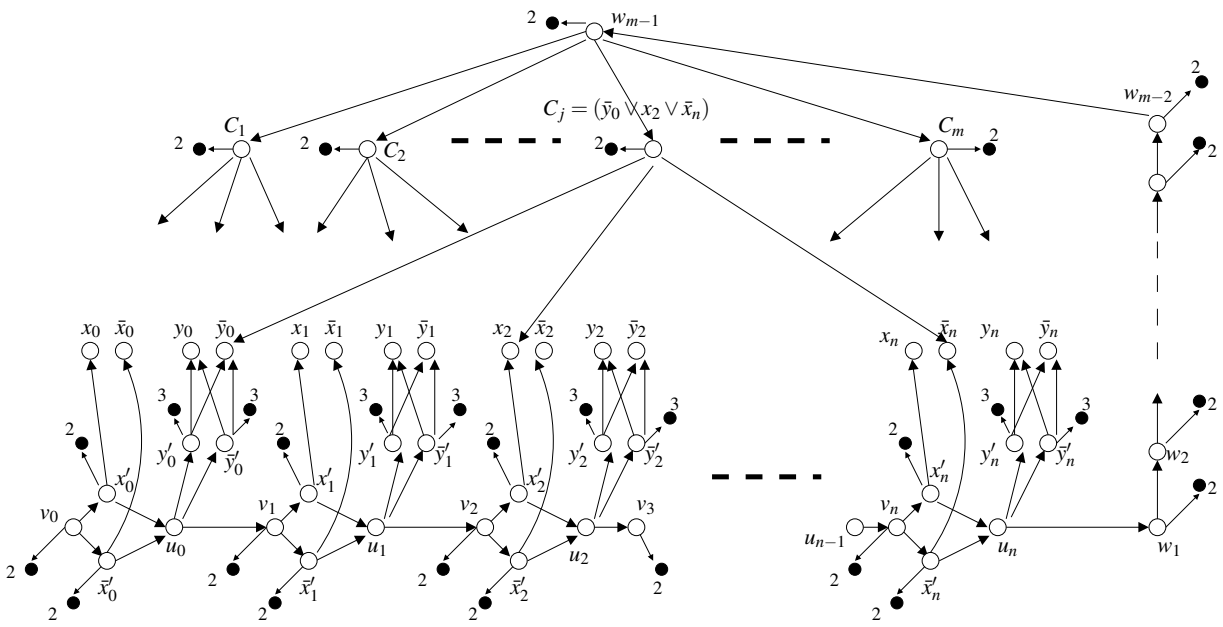

Figure 2: Example of the reduction in the proof of Lemma 6. A small black node with an integer i beside and that is the out-neighbor of a vertex v corresponds to i leaves that are in $N^+(v)$.

Since v_0 has out-degree 4, $csn(D,v_0) \geq sn(D,v_0) \geq 4$ and the first step of the observer, endowed with 4 marks, consists in marking the 4 out-neighbors of v_0 (the two leaves and x'_0 and $\bar{x}'_0$). We now show that $sn(D,v_0) = 4$ (and $csn(D,v_0) = 4$) if and only if Φ is true.

Assume by induction on $0 \leq i \leq n$ that, after the $(3i+1)^{th}$ step of the observer,

1. the fugitive occupies vertex v_i after having followed the directed path $P = (v_0, a'_0, u_0, v_1, a'_1, u_1, \cdots, v_{i-1}, a'_{i-1}, u_{i-1}, v_i)$ where $a'_j \in \{x'_j, \bar{x}'_j\}$ for any $0 \leq j < i$, and
2. at this step, the set of vertices marked consists of all the vertices in P and the outneighbors of the vertices in P plus, for any $j < i$, exactly one of y_j or $\bar{y}_j$, and
3. for any $j < i$, the choice of which vertex has been marked in $\{y_j, \bar{y}_j\}$ depends only on the observer (not on the path followed by the fugitive).

All these assumptions are satisfied for $i = 0$. If $i < n$, we will show it remains true for $i+1$.

Moreover, if $i = n$, we will show that,

Claim 2. *After the $3(n+1)^{th}$ move of the fugitive,*

1. *the fugitive occupies vertex w_1 after having followed the directed path $P = (v_0, a'_0, u_0, v_1, a'_1, u_1, \cdots, v_n, a'_n, u_n, w_1)$ where $a'_j \in \{x'_j, \bar{x}'_j\}$ for any $0 \leq j \leq n$, and*
2. *at this step, the set of vertices marked consists of all the vertices of P and the outneighbors of the vertices in P but $N^+(w_1)$, plus, for any $j \leq n$, exactly one of y_j and $\bar{y}_j$, and*
3. *for any $j \leq n$, the choice of which vertex has been marked in $\{y_j, \bar{y}_j\}$ depends only on the observer (not on the path followed by the fugitive).*

Note that, by the induction hypothesis, the 4 out-neighbors of v_i (two of which are leaves) are marked. Since it is useless for the fugitive to remain at v_i (by Theorem 3), then it goes to $a'_i \in \{x'_i, \bar{x}'_i\}$. a'_i has 4 out-neighbors that are, by the induction hypothesis, unmarked. Hence, the observer must mark these 4 vertices. In particular, if $a'_i = x'_i$ (resp., if $a'_i = \bar{x}'_i$) then the observer marks $a_i = x_i$ (resp., $a_i = \bar{x}_i$) while $\bar{x}_i$ (resp., x_i) remains unmarked. Then, the fugitive must go to u_i since the other 3 out-neighbors of a'_i have no out-neighbors.

Since u_i has 3 out-neighbors ($y'_i, \bar{y}'_i$ and v_{i+1}), the observer must mark these three vertices. Moreover, assume that the fourth vertex marked by the observer at this step is neither y_i nor $\bar{y}_i$, then the fugitive goes to the vertex in $\{y'_i, \bar{y}'_i\}$ with still its 5 out-neighbors unmarked, and then the fugitive will win at the next step. Hence, the observer must mark b_i that is either y_i or $\bar{y}_i$. It is important to note that the choice of which of these two vertices is marked is completly free for the observer. After this step of the observer, both y'_i and $\bar{y}'_i$ have 4 unmarked out-neighbors and all the 5 out-neighbors of y'_i and $\bar{y}'_i$ have no out-neighbors themselves. Hence, the fugitive would loose if she went to y'_i or $\bar{y}'_i$. Hence, the fugitive must go to v_{i+1} (where v_{n+1} is set to be w_1).

If $i < n$, then v_{i+1} has exactly 4 out-neighbors that must be marked by the observer, and then the induction hypothesis is satisfied for $i+1$. If $i = n$, then $v_{i+1} = w_1$ and the Claim holds.

Let X be the set of vertices consisting of $\{w_2, \cdots, w_{m-1}, C_1, \cdots, C_m\}$ plus the $2(m-1)$ leaves adjacent to $w_1, \cdots, w_{m-1}$. Let Y be the set of vertices consisting of $\{x_0, \bar{x}_0, y_0, \bar{y}_0, \cdots, x_n, \bar{x}_n, y_n, \bar{y}_n\}$ plus the $2m$ leaves adjacent to the C_j's.

By the above Claim, after the $3(n+1)^{th}$ move of the fugitive, no vertices in X are marked. Moreover, the set of marked vertices in Y is $\{a_0, b_0, \cdots, a_n, b_n\}$ where, for any $i \leq n$, $a_i \in \{x_i, \bar{x}_i\}$ has been imposed by the fugitive and $b_i \in \{y_i, \bar{y}_i\}$ has been chosen by the observer. In particular, if Φ is true, the observer can choose the b_i's such that $F(a_0, b_0, \cdots, a_n, b_n)$ is true whatever be the choices of the fugitive. On the other hand, if Φ is false, the fugitive can choose the a_i's such that $F(a_0, b_0, \cdots, a_n, b_n)$ is false whatever be the choices of the observer.

Now, from the $(3n+1)^{th}$ step of the observer to the end of its $(3n+m)^{th}$ step, the observer can mark at most $4(m-1) = |X|$ vertices. Moreover, between these steps, the fugitive must follow the path $Q = (w_1, \cdots, w_{m-1})$ (all other vertices the fugitive can access having no out-neighbors). Hence, the only choice for the observer is to successively mark all vertices in X otherwise, at some step along the path Q or just after the $(3n+m)^{th}$ step of the observer, the fugitive could have reached an unmarked vertex. Note that the marking process can be done in a connected way.

Finally, after the $(3n+m)^{th}$ step of the observer, the fugitive stands on w_{m-1}, all vertices in $\{C_1, \cdots, C_m\}$ are marked while the set of marked vertices in Y is $\{a_0; b_0, \cdots, a_n, b_n\}$. Now, if Φ is false, by the choice of the a_i's by the fugitive, there is a clause C_j with its 5 out-neighbors unmarked: the fugitive goes to C_j and will win at the next step. On the other hand, if Φ is true, by the choice of the b_i's by the observer, all C_j's have at most 4 unmarked out-neighbors. Whatever be the next moves of the fugitive, she will reach a marked vertex without out-neighbors. □

Lemma 7. *For every $k \geq 1$, the problem of deciding whether $sn(G, v_0) \leq k$ (resp., $csn(G, v_0) \leq k$) is in PSPACE.*

Proof. Let n be the number of vertices in graph G. Every game lasts at most n rounds. At each round, the configuration (M, f) can be encoded within polynomial space. This means that the problem is in NPSPACE (nondeterministic polynomial space)—a nondeterministic Turing machine deciding the problem uses polynomial space on every branch of its computation. By Savitch's theorem [Sav70], the problem is in PSPACE. □

Theorem 6. *The problem of deciding whether $sn(G, v_0) \leq 4$ (resp., $csn(G, v_0) \leq 4$) is PSPACE-complete in DAGs.*

The following theorem provides an exponential algorithm computing $sn(G, v_0)$. In the statement of the theorem we use a modified big-Oh notation that suppresses all polynomially bounded factors. For functions f and g we write $f(n) = O^*(g(n))$ if $f(n) = O(g(n)poly(n))$, where $poly(n)$ is a polynomial.

Theorem 7. *$sn(G, v_0)$ (resp., $csn(G, v_0)$) can be computed in time $O^*(2^n)$ on n-node graphs.*

Proof. For each $k \geq 1$, we decide if $sn(G, v_0) \leq k$. We consider the *arena digraph* $\mathcal{G}$ whose vertices are configurations of the game, i.e., the pairs (M, f) where $M \subseteq V(G)$, $f \in M$ and $|M \setminus \{v_0\}| = ki$ for some $i > 0$ (or $M = V(G)$). Moreover, there is an arc from (M, f) to (M', f') if $f' \in N(f)$ and $M \subset M'$ and $|M'| = |M| + k$ (or $|V(G) \setminus M| \leq k$ and $M' = V(G)$). Note that $|V(\mathcal{G})| \leq n \sum_{i=1}^{\lceil \frac{n-1}{k} \rceil} \binom{n}{ki+1} \leq 2^n n$. The amount of arcs in $\mathcal{G}$ is at most $2^n nk$.

We consider the following labelling process. Initially, all configurations $(V(G), v)$, for any $v \in V(G)$, are labeled with $\lceil \frac{n-1}{k} \rceil$, and all remaining configurations are labeled with ∞. Then, iteratively, a configuration (M, f) with $|M| = ki + 1$ is labeled i if, for any $f' \in N_G(f)$, there is a configuration (M', f') that is an out-neighbor of (M, f) and that is labeled at most $i + 1$.

Claim 3. *$sn(G, v_0) \leq k$ iff there is a configuration (M, v_0), $|M| = k + 1$, labeled with 1.*

We first show by induction on i, that the observer can win starting from any configuration labeled with $\lceil \frac{n-1}{k} \rceil - i$. If $i = 0$, the result holds trivially. Assume that the result holds for $\lceil \frac{n-1}{k} \rceil - 1 > i > 0$. Let (M, f) be a configuration labeled with $\lceil \frac{n-1}{k} \rceil - (i + 1)$. For any $f' \in N(f)$, by definition of the labelling process, there is a configuration (M', f') out-neighbor of (M, f) and labeled with $\lceil \frac{n-1}{k} \rceil - i$. If the fugitive goes from f to f', then the observer marks the vertices in $M' \setminus M$ and the game reaches the configuration (M', f'). Hence, by the induction hypothesis, the observer wins. So, applying the result for $i = \lceil \frac{n-1}{k} \rceil - 1$, the observer wins starting from any configuration (M, v_0), $|M| = k + 1$, labeled 1. To reach this configuration, the first step of the observer is to mark the k vertices in $M \setminus \{v_0\}$. Therefore, $sn(G, v_0) \leq k$.

Now, assume that $sn(G, v_0) \leq k$. Let σ be a winning k-strategy for the observer. For any walk $W = (v_0, v_1, \cdots, v_i)$ followed by the fugitive, let $M(W)$ be the set of vertices marked by the observer (using σ) after the fugitive has followed W until v_i and when it is the turn of the fugitive. It follows by reverse induction on i that the labelling process labels $(M(W), v_i)$ with $i + 1$. In particular, this shows that $(\{v_0\} \cup \sigma(\{v_0\}, v_0), v_0)$ is labeled with 1. ◇

To obtain the same result with $csn(G, v_0)$, it is sufficient to modify the definition of a configuration (M, f) by imposing that M must induce a connected subgraph. □

4 Polynomial-time Algorithms in some graphs'classes

In this section, we give polynomial-time algorithms to compute the surveillance number of trees and interval graphs. Moreover, in both these classes of graphs, we show that connectedness does not cost, i.e., the surveillance number equals the connected surveillance number.

4.1 Keeping tree under surveillance

In this section, we first present a polynomial-time algorithm to compute $sn(T, v_0) = csn(T, v_0)$ for any tree $T = (V, E)$ rooted at $v_0 \in V$. Let $k \geq 0$. We define the function $f_k : V(T) \to \mathbb{N}$ in the following recursive way:

- $f_k(v) = 0$ for any leaf v of T;
- for any $v \in V(T)$ with d children, $f_k(v) = \max\{0, d + \sum_{w \in C} f_k(w) - k\}$ where C is the set of children of v.

Lemma 8. *For any tree T rooted in v_0, if $f_k(v_0) > 0$ then $sn(T, v_0) > k$. Moreover, if $f_k(v_0) = 0$ then $csn(T, v_0) \leq k$.*

Proof. The result holds if T is reduced to one vertex. So we may assume that T has height at least 1. Recall that the height of T is the maximum length (number of edges) of a path between the root v_0 and a leaf of T.

We prove by induction on the height of T that the observer cannot win the game marking at most k vertices per step, even if at most $f_k(v_0) - 1$ vertices in $V(T) \setminus \{v_0\}$ are initially marked. Moreover, we prove that the observer

can win in a connected way, marking at most k vertices per step, if at most $f_k(v_0)$ vertices plus v_0 are initially marked.

If T has height 1 and v_0 has degree d, then $f_k(v_0) = \max\{0, d-k\}$ and the result holds. Indeed, if v_0 and $f_k(v_0)$ other vertices are initially marked, then during its first step, the observer marks all remaining vertices and wins in a connected way. On the other hand, if v_0 and at most $f_k(v_0)-1$ vertices are marked, then after the first step of the observer (when he marks k other vertices), at least one neighbor of v_0 is still unmarked and the fugitive can go to it and wins.

Now, assume that the result holds for any tree of height $h \geq 1$. Let T be rooted in v_0 and of height $h+1$, we show that the result holds.

Let $(v_1, \cdots, v_r)$ be the children of v_0 and let T_i be the subtree of T rooted in v_i, $1 \leq i \leq r$. By the induction hypothesis, for any $1 \leq i \leq r$, there is a set $I_i \subseteq V(T_i) \setminus \{v_i\}$ of $f_k(v_i)$ vertices such that, if the vertices of I_i and v_i are initially marked in T_i, then the observer can win in T_i starting from v_i, marking at most k vertices per step and in a connected way. In particular, $I_i \cup \{v_i\}$ induces a (connected) subtree of T_i. On the contrary, if strictly less than $f_k(v_i)$ vertices are initially marked in $V(T_i) \setminus \{v_i\}$, then the fugitive will win in T_i against an observer marking at most k vertices per step.

Then, in T, if at least $f_k(v_0)$ vertices can be marked initially in $V(T) \setminus \{v_0\}$, then a connected k-strategy consists of the following. The set of vertices that are initially marked union the vertices marked during the first step of the observer is $J = N[v_0] \cup (\bigcup_{1 \leq i \leq r} I_i)$. It is possible since $|J| \leq 1 + f_k(v_0) + k$ and J induces a (connected) subtree of T. Then the fugitive moves to some child v_i ($1 \leq i \leq r$) of v_0. Since the vertices of I_i and v_i are already marked, the observer will win in T_i in a connected way.

On the contrary, if strictly less than $f_k(v_0)$ vertices can be marked initially in $V(T) \setminus \{v_0\}$, then there is at least one child v_i ($1 \leq i \leq r$) such that either v_i is not marked after the first step of the observer, or at most $f_k(v_i) - 1$ vertices in $V(T_i) \setminus \{v_i\}$ are marked after the first step of the observer. In both cases, the fugitive will win in T_i. □

Theorem 8. *For any tree T rooted in v_0, $sn(T, v_0) = csn(T, v_0)$ and can be computed in time $O(n \cdot \log n)$.*

Proof. By Lemma 8, $sn(T, v_0) = csn(T, v_0) = \min\{k \ : \ f_k(v_0) = 0\}$ (Note that, v being fixed, $f_k(v)$ is a decreasing function of k).The result comes from the fact that $f_k(v_0)$ can be computed in linear time and so, the minimum k such that $f_k(v_0) = 0$ can be searched using dichotomy. □

We now give a combinatorial characterization of $sn(T, v_0)$ of any tree T rooted in v_0.

Lemma 9. *For any tree T rooted in v_0, and for any $k < sn(T, v_0)$, there is $S \subseteq V(T)$ inducing a subtree of T containing v_0 such that $\left\lceil \frac{|N[S]|-1}{|S|} \right\rceil > k$.*

Proof. Let $k < sn(T, v_0)$. By Lemma 8, $f_k(v_0) > 0$. Let S be the inclusion-maximal subtree of T containing v_0 and such that $f_k(v) > 0$ for all vertices in S. We show by induction on the height of S that $f_k(v_0) = |N[S]| - 1 - k|S|$.

If $S = \{v_0\}$ and v_0 has degree d, then $f_k(v_0) = d - k = |N[S]| - 1 - k|S| > 0$ because for any child v of v_0, $f_k(v) = 0$. So the result holds for height 0.

Assume that the result holds for any subtree of height $h \geq 0$ and assume that S has height $h+1$. Let d be the degree of v_0 and let $v_1, \cdots, v_r$, $1 \leq r \leq d$, be the children of v_0 with $f_k(v_i) > 0$. Let S_i be the subtree of S rooted in v_i, $1 \leq i \leq r$, and let $N[S_i]$ be the vertices of S_i or in the neighborhood of S_i in the subtree of T rooted in v_i. By the induction hypothesis, $f_k(v_i) = |N[S_i]| - 1 - k|S_i|$ for any $1 \leq i \leq r$. Now, $f_k(v_0) = d - k + \sum_{1 \leq i \leq r} f_k(v_i) = d - k + \sum_{1 \leq i \leq r}(|N[S_i]| - 1 - k|S_i|) = d - k + (|N[S]| - 1 - (d-r)) - r - k(|S| - 1) = |N[S]| - 1 - k|S|$.

Hence, we have $0 < f_k(v) = |N[S]| - 1 - k|S|$. Therefore, $\left\lceil \frac{|N[S]|-1}{|S|} \right\rceil > k$. □

Lemma 10. *For any tree T rooted in v_0, for any $k \geq sn(T, v_0)$, for any $S \subseteq V(T)$ inducing a subtree of T containing v_0, we have $\left\lceil \frac{|N[S]|-1}{|S|} \right\rceil \leq k$.*

Proof. We consider the following game. Initially, an unbounded number of fugitives are in v_0 which is initially marked. Then, at most k vertices of $T \setminus \{v_0\}$ are marked. At each turn, each fugitive can move along an edge of the tree, and then, for each vertex v that is reached for the first time by a fugitive, at most k vertices can be marked in T_v the subtree of T rooted in v. The fugitives win if at least one fugitive reaches an unmarked vertex, they loose otherwise.

We first show that if $k \geq sn(T, v_0)$ then the fugitives loose in this game. Assume that $k \geq sn(T, v_0)$. Then there is a winning k-strategy σ for the "normal" surveillance game in T starting from v_0. Recall that by Theorem 3, we can restrict the fugitive to follow an induced path. Since for any $t \in V(T)$, there is a unique induced path from v_0 to t, σ can be defined uniquely by the position of the fugitive. That is, in the case of trees, we can define a k-strategy as a function that assigns a subset $\sigma(t) \subseteq V(T_t)$ (of size at most k) to any vertex $t \in V(T)$. Now, in the game with several fugitives, we consider the following strategy: each time a vertex t is reached for the first time by a fugitive, then we mark the vertices in $\sigma(t)$. It is easy to check that the fugitives cannot win against such a strategy.

Finally, we show that if there is a subtree S containing v_0 such that $\left\lceil \frac{|N[S]|-1}{|S|} \right\rceil > k$, then the fugitives win the new game. Indeed, the fugitives first occupy all vertices of S. At this step, at most $k \cdot |S| + 1$ vertices have been marked (because S is connected and v_0 is marked and for each vertex in S at most k vertices in $V(T) \setminus \{v_0\}$ are marked). Since $|N[S]| > k \cdot |S| + 1$, at least one unmarked vertex in $N[S]$ will be reached by some fugitive during the next step.

Hence, $sn(T, v_0) \geq \max \left\lceil \frac{|N[S]|-1}{|S|} \right\rceil$ where the maximum is taken over all $S \subseteq V(T)$ inducing a subtree of T containing v_0. □

Theorem 9. *For any tree T rooted in v_0, $sn(T, v_0) = \max \left\lceil \frac{|N[S]|-1}{|S|} \right\rceil$ where the maximum is taken over all $S \subseteq V(T)$ inducing a subtree of T containing v_0.*

4.2 To keep an Interval Graph under surveillance

We recall that an *interval graph* G is the intersection graph of a set of real intervals. This set of real intervals is a *realization* of G. In this section, we give a polynomial-time algorithm for computing the surveillance number in interval graphs. Moreover, we show that it is equal to the connected variant in this class of graphs. It is important to recall that, by Theorem 3, we do not help the observer when forcing the fugitive to move at each step and to follow induced paths. Hence, in this section, we assume that the fugitive obeys these restrictions.

Let G be a connected interval graph and $v_0 \in V(G)$. We consider any realization I of G such that, no two intervals have a common end (such a realization clearly always exists). Let us say that $v \prec_L w$ if the left (smallest) end of (the interval of) v is smaller than the left end of w, and $v \succ_R w$ if right (largest) end of v is larger than the right end of w.

We partition $V(G)$ into several subsets. Let $\mathcal{C}$ be the subset of vertices the interval of which contains the interval of v_0. Note that $v_0 \in \mathcal{C} \subseteq N(v_0)$ and that $\mathcal{C}$ induces a clique. Since G is an interval graph, $V \setminus N[v_0]$ induces a subgraph H the connected components of which are interval graphs. Let $\mathcal{L}$ be the vertices of the components of H with their interval more to the left than the interval of v_0, i.e., the greatest end of an interval in $\mathcal{L}$ is strictly smaller than the smallest end of the interval of v_0. Respectively, $\mathcal{R}$ be the vertices of the components of H with their interval more to the right than the interval of v_0. Note that $\mathcal{R}$ and $\mathcal{L}$ are disjoint and are separated by $N[v_0]$ because G is an interval graph (no interval can be both more to the left and more to the right than the interval of v_0). Let $\mathcal{C}_L$ be the vertices in $N(v_0) \setminus \mathcal{C}$ with neighbors in $\mathcal{L}$ and let $\mathcal{C}_R$ be the vertices in $N(v_0) \setminus \mathcal{C}$ with neighbors in $\mathcal{R}$. Finally, let $\mathcal{C}' = V(G) \setminus (\mathcal{L} \cup \mathcal{C}_L \cup \mathcal{C} \cup \mathcal{C}_R \cup \mathcal{R})$. Note that, for any $v \in \mathcal{C}'$, $v_0 \in N(v) \subseteq N[v_0]$.

Claim 4. *$(\mathcal{L}, \mathcal{C}_L, \mathcal{C}, \mathcal{C}', \mathcal{C}_R, \mathcal{R})$ is a partition of $V(G)$.*

Proof. Since, $N[v_0] = \mathcal{C} \cup \mathcal{C}' \cup \mathcal{C}_R \cup \mathcal{C}_L$, $(\mathcal{L}, \mathcal{C}_L, \mathcal{C}, \mathcal{C}_R, \mathcal{R})$ covers $V(G)$. It only remains to prove that $\mathcal{C}_R \cap \mathcal{C}_L = \emptyset$. Indeed, if $v \in \mathcal{C}_R \cap \mathcal{C}_L$, then v must be adjacent to a vertex in $\mathcal{L}$ and to a vertex in $\mathcal{R}$. However, it means that the interval of v contains the one of v_0 and $v \in \mathcal{C}$, a contradiction. □

Recall that the fugitive is forced to follow an induced path. We now describe the structure of induced paths in G. Roughly, the next lemma says that once the fugitive has chosen a "side" (left or right) it has to remain on this side, and the choice occurs after one or two moves. Moreover, once the fugitive has chosen a side, it must go "further" into this side or it should stop.

Lemma 11. *Let $P = (v_0, v_1, \cdots, v_p)$ be an induced path starting from v_0 in any connected interval graph G. Let $\mathcal{L}, \mathcal{C}_L, \mathcal{C}, \mathcal{C}', \mathcal{C}_R, \mathcal{R}$ be defined as above. Then, there are three possible cases:*

1. *Either $v_1 \in \mathcal{C}'$ and then $p = 1$;*
2. *Either $v_2 \in \mathcal{L}$, and then*
   - *for any $i > 1$, $v_i \in \mathcal{L}$ and $v_1 \in \mathcal{C}_L \cup \mathcal{C}$;*
   - *for any $0 < i < p-1$, $v_{i+1} \prec_L v_i$;*
   - *if $v_{p-1} \prec_L v_p$ then $N(v_p) \setminus \cup_{i<p} N(v_i) = \emptyset$.*
3. *Or $v_2 \in \mathcal{R}$, and then*
   - *for any $i > 1$, $v_i \in \mathcal{R}$ and $v_1 \in \mathcal{C}_R \cup \mathcal{C}$;*
   - *for any $0 < i < p-1$, $v_{i+1} \succ_R v_i$;*
   - *if $v_{p-1} \succ_R v_p$ then $N(v_p) \setminus \cup_{i<p} N(v_i) = \emptyset$.*

Proof. Clearly, v_2 cannot be in $N[v_0] = \mathcal{C} \cup \mathcal{C}' \cup \mathcal{C}_R \cup \mathcal{C}_L$ because P is induced. Hence, $v_2 \in \mathcal{R} \cup \mathcal{L}$. In particular, if $v_1 \in \mathcal{C}'$, then all neighbors of v_1 are in $N[v_0]$ and thus $p = 1$. Let us assume that $v_2 \in \mathcal{L}$. The case $v_2 \in \mathcal{R}$ can be delt with similarly.

By the claim above, v_2 cannot be adjacent to some vertex in $\mathcal{C}_R \cup \mathcal{C}'$. Hence, $v_1 \in \mathcal{C}_L \cup \mathcal{C} = N[v_0] \setminus \mathcal{C}_R$. Moreover, for any $i > 1$, $v_i \notin N[v_0]$ because P is induced. Since $N[v_0]$ separates $\mathcal{R}$ and $\mathcal{L}$, and since $v_2 \in \mathcal{L}$, then for any $i > 1$, $v_i \in \mathcal{L}$.

Let us assume that $v_i \prec_L v_{i+1}$ for some $0 < i < p$ such that i is minimum for this property. We show that $i = p-1$ and $N(v_p) \setminus \cup_{j<p} N(v_j) = \emptyset$.

We first consider the case when the interval of v_i does not contain the interval of v_{i+1}. Since $v_i \prec_L v_{i+1}$, we get that $v_{i+1} \succ_R v_i$. Since P is induced, $v_{i+1} \notin N(v_{i-1})$. This implies, since $v_i \in N(v_{i-1})$, that $v_i \succ_R v_{i-1}$. Since, by minimality of i, $v_i \prec_L v_{i-1}$, then the interval of v_{i-1} must be contained into the interval of v_i. Therefore, if $i > 1$, then $v_{i-2} \in N(v_{i-1}) \setminus N(v_i) = \emptyset$, a contradiction. Then $i = 1$ and the interval of $v_{i-1} = v_0$ is strictly more to the left than the interval of $v_{i+1} = v_2$ which contradicts the fact that $v_2 \in \mathcal{L}$.

Therefore, the interval of v_{i+1} must be contained into the interval of v_i. Then $N(v_{i+1}) \subseteq N(v_i) \subseteq \cup_{j \leq i} N(v_j)$. If $i < p-1$, then $v_{i+2} \in N(v_{i+1}) \cap N(v_i)$ contradicting the fact that P is induced. Hence $i = p-1$, and we get that $N(v_p) \setminus \cup_{j<p} N(v_j) = \emptyset$. □

The next lemma shows that, in interval graphs, we can define few particular induced paths that "dominate" all paths. That is, if the observer is able to win when the fugitive is constrained to follow one of these particular paths, then the observer always win.

Let $v_L \in \mathcal{C}_L$ be smallest vertex of $\mathcal{C}_L$ according to $\prec_L$, i.e., $v \prec_L w$ for any $w \in \mathcal{C}_L \setminus \{v\}$. For any $v_1 \in \mathcal{C} \cup \{v_L\}$, let us define $P_L(v_1)$ as the longest induced path $(v_0, v_1, v_2, \cdots, v_p)$ such that, for any $i > 1$, v_{i+1} is the smallest vertex of $N(v_i)$ according to $\prec_L$, i.e., $v_{i+1} \prec_L w$ for any $w \in N(v_i)$. Intuitively, except for the first move, we choose as next vertex the neighbor with leftest left end.

By symmetry, let $v_R \in \mathcal{C}_R$ be greatest vertex of $\mathcal{C}_R$ according to $\succ_R$, i.e., $v \succ_R w$ for any $w \in \mathcal{C}_R \setminus \{v\}$. For any $v_1 \in \mathcal{C} \cup \{v_R\}$, let us define $P_R(v_1)$ as the longest induced path $(v_0, v_1, v_2, \cdots, v_p)$ such that, for any $i > 1$, v_{i+1} is the largest vertex of $N(v_i)$ according to $\succ_R$, i.e., $v_{i+1} \succ_R w$ for any $w \in N(v_i) \setminus \{v_i\}$. Except for the first move, we choose as next vertex the neighbor with rightest right end.

Finally, for any path $P = (v_0, \cdots, v_p)$ an for any $i \leq p$, let $P^i = N[\{v_0, \cdots, v_i\}]$ the set of the vertices that are in $\{v_0, \cdots, v_i\}$ or that have a neighbor in $\{v_0, \cdots, v_i\}$.

Lemma 12. *Let G be a connected interval graph and let $P = (v_0, \cdots, v_p)$ be an induced path starting from v_0 and $p > 1$. For any $i \leq p$, we have $P^i \subseteq Q^{i'}$ where $i' = \min\{i, |Q|\}$ and Q is:*

- *$P_L(v_L)$ if $v_1 \in \mathcal{C}_L$;*
- *$P_R(v_R)$ if $v_1 \in \mathcal{C}_R$;*
- *$P_L(v_1)$ if $v_1 \in \mathcal{C}$ and $v_2 \in \mathcal{L}$, and*
- *$P_R(v_1)$ if $v_1 \in \mathcal{C}$ and $v_2 \in \mathcal{R}$.*

Proof. By Lemma 11, P must satisfy one of the four cases.

Let us first assume that $v_1 \in \mathcal{C}_L$, and let $Q = P_L(v_L) = (v_0, v_L, q_2, \cdots, q_p)$. By Lemma 11, $v_i \in \mathcal{L}$ for any $i \geq 2$. Hence, since no vertices of $\mathcal{L} \cup \mathcal{C}_L$ has a neighbor in $\mathcal{R}$, we have $N[P] \subseteq \mathcal{L} \cup N[v_0]$.

Similarly, $N[Q] \subseteq \mathcal{L} \cup N[v_0]$. We show that $N[Q] = \mathcal{L} \cup N[v_0]$. Clearly, $N[v_0] \subseteq N[Q]$. Let $v \in \mathcal{L} \setminus Q$. Note that $\mathcal{L} \cup N[v_0]$ is connected and let $R = (v_0, a_1, a_2, \cdots, a_i, v)$ $(i \geq 1)$ be a shortest path from v_0 to v. Since, $v \in \mathcal{L}$, we must have $v \prec_L v_0$, and since R is an induced path, we get $v \prec_L a_i \prec_L a_{i-1} \prec_L \cdots \prec_L a_1 \prec_L v_0$. By induction on $j \leq i$, we show that $(v_0, v_L, q_2, \cdots, q_j, a_{j+1}, \cdots, a_i, v)$ is a shortest path from v_0 to v and therefore $v \in N[Q]$. Since v_L is the smallest neighbor of v_0 according to $\prec_L$, then $a_2 \in N[v_L]$ (or $v \in N[v_L]$ if $i = 1$) and then $(v_0, v_L, a_2, \cdots, a_i, v)$ is a shortest path from v_0 to v. Assume $(v_0, v_L, q_2, \cdots, q_j, a_{j+1}, \cdots, a_i, v)$ is a shortest path from v_0 to v for some $j < i$. Then, $j < p$ because otherwise, Q would not be a maximal induced path. Moreover, since q_{j+1} is the smallest neighbor of q_j according to $\prec_L$, then $a_{j+2} \in N[q_{j+1}]$ (or $v \in N[q_{j+1}]$ if $j = i-1$) and then $(v_0, v_L, q_2, \cdots, q_j, q_{j+1}, a_{j+2}, \cdots, a_i, v)$ is a shortest path from v_0 to v. Hence, $\mathcal{L} \cup N[v_0] \subseteq N[Q]$ and so $N[Q] = \mathcal{L} \cup N[v_0]$.

Therefore, for any $i \geq p = |Q|$, $P^i \subseteq \mathcal{L} \cup N[v_0] = N[Q] = Q^p$.

Now, we show by induction on $i < p$ that $P^i \subseteq Q^i$. Since $P^0 = Q^0 = N[v_0]$, the result holds for $i = 0$. Let $0 \leq i < p-1$ and assume that $P^i \subseteq Q^i$. Let $v \in P^{i+1} \setminus P^i$. Then, $(v_0, v_1, \cdots, v_{i+1}, v)$ is an induced path from v_0 to v. As previously, we show that $(v_0, v_L, q_2, \cdots, q_{i+1}, v)$ is an induced path from v_0 to v. Therefore, $v \in Q^{i+1}$ and the result holds.

The case $v_1 \in \mathcal{C}_R$ can be handled similarly by symmetry.

Now, if $v_1 \in \mathcal{C}$ and $v_2 \in \mathcal{L}$, we can prove in a similar way that $N[P] \subseteq \mathcal{L} \cup N[v_1] = N[P_L(v_1)]$. Hence, for any $i \geq p = |P_L(v_1)|$, $P^i \subseteq \mathcal{L} \cup N[v_0] = N[P_L(v_1)] = Q^p$. Moreover, a similar induction on $i < p$ allows to prove that $P^i \subseteq Q^i$. The case $v_1 \in \mathcal{C}$ and $v_2 \in \mathcal{R}$ is symmetric. □

The previous two lemmas roughly say that the fugitive can only choose five kinds of induced paths: the ones with second vertex in $\mathcal{C}'$ (such induced paths have only one edge), the ones with second vertex in $\mathcal{L}$ and going through $\mathcal{L}$, those with second vertex in $\mathcal{R}$ and going through $\mathcal{R}$, and the paths with second vertex $x \in \mathcal{C}$ and then either going through $\mathcal{L}$ or through $\mathcal{R}$. Moreover, once the observer knows which kind of path has been chosen, it is sufficient for the observer to protect one particular path. However, during the first step (before the first move of the fugitive), the first set S_0 of marked vertices must be chosen by the observer independently of what will choose the fugitive. Similarly, if the fugitive first goes to $x \in \mathcal{C}$, the observer cannot yet guess on which side the fugitive will go. Hence, the set S_x of marked vertices during the second step (before the second move of the fugitive) must be independent of the next choice of the fugitive (S_x may only depends on x). The next theorem formalizes this.

Now, we order the vertices of $V(G)$ in the following way. The vertices in $N[v_0]$ are ordered in any arbitrary ordering, any vertex in $\mathcal{L}$ is smaller than a vertex in $N[v_0]$ and any vertex in $\mathcal{R}$ is greater than the vertices in $N[v_0]$. Finally, vertices in $\mathcal{L}$ are ordered according to $\succ_R$, i.e., for any $v,w \in \mathcal{L}$, $v < w$ iff $w \succ_R v$. Symmetrically, vertices in $\mathcal{R}$ are ordered according to $\prec_L$. We say that a set $S \subseteq V(G)$ is *contiguous* if for any $a,b \in S$ and $a < w < b$, then $w \in S$. Note that a contiguous subset of a connected interval graph induces a connected subgraph.

Theorem 10. *Let G be an interval graph and $v_0 \in V(G)$. Let k be the smallest integer s.t.:*

- *there exists $S_0 \subseteq V(G)$ with $N[v_0] \subseteq S_0$, $|S_0 \setminus \{v_0\}| \leq k$, and*
- *for any $i > 0$, $|P_L^i(v_L) \setminus S_0| \leq i \cdot k$ and $|P_R^i(v_R) \setminus S_0| \leq i \cdot k$, and*
- *for any $x \in \mathcal{C} \setminus \{v_0\}$, there is $S_x \subseteq V(G) \setminus S_0$ with $N[v_0] \cup N[x] \subseteq S_0 \cup S_x$ and $|S_x| \leq k$, and*
- *for any $i > 1$ and any $x \in \mathcal{C} \setminus \{v_0\}$, $|P_L^i(x) \setminus (S_0 \cup S_x)| \leq (i-1) \cdot k$ and $|P_R^i(x) \setminus (S_0 \cup S_x)| \leq (i-1) \cdot k$.*

Then, $sn(G,v_0) = csn(G,v_0) = k$. Moreover, we show that S_0 can be chosen contiguous and the sets S_x can be chosen such that $S_0 \cup S_x$ is contiguous for any $x \in \mathcal{C}$ (without increasing k).

Proof. Let k be the smallest integer defined as in the statement of the theorem. We first show that $k' = sn(G,v_0)$ is at least k.

Claim 5. $sn(G,v_0) = k' \geq k$

Let σ be any optimal winning strategy for the observer, i.e., marking k' vertices at each step. Let $S_0 = \sigma(\{v_0\},v_0) \cup \{v_0\}$ and, for any $x \in \mathcal{C}$, let $S_x = \sigma(S_0,x)$. Obviously, $|S_0| \leq k'+1$, $|S_x| \leq k'$, $N[v_0] \subseteq S_0$ and $N[x] \cup N[v_0] \subseteq S_0 \cup S_x$.

Now, let us assume that the fugitive follows the induced path $P_L(x) = (v_0, x, v_2, \cdots, v_p)$ for some $x \in \mathcal{C}$. At step $1 < i \leq p$, when it is the turn of the fugitive that stands at v_i, the observer must have marked at least the vertices in $N[\{v_0, x, v_2, \cdots, v_i\}]$. Moreover, during the first two steps of the strategy, the observer has marked the vertices of S_0 and S_x by definition of σ. Hence, during the $i-1$ steps after the first two steps, the observer must have marked the vertices in $N[\{v_0, x, v_2, \cdots, v_i\}] \setminus (S_0 \cup S_x) = P_L^i(x) \setminus (S_0 \cup S_x)$ which proves that $|P_L^i(x) \setminus (S_0 \cup S_x)| \leq (i-1) \cdot k'$.

The other properties can be proved in the same way and thus, $k' \geq k$. ◇

Claim 6. *There exist S_0^* and S_x^* ($x \in \mathcal{C}$) that are contiguous sets and that satisfy the same properties as S_0 and S_x ($x \in \mathcal{C}$). That is, S_0 and S_x ($x \in \mathcal{C}$) may be chosen contiguous.*

Recall that to define contiguous sets, we have ordered the vertices in $V(G)$.

Let $\ell = |S_0 \cap \mathcal{L}|$ and $r = |S_0 \cap \mathcal{R}|$. Let S_0^* be the set obtained from the union of $N[v_0]$, the ℓ greatest vertices in $\mathcal{L}$ and the r smallest vertices in $\mathcal{R}$. Note that S_0^* is contiguous and that $S_0 = S_0^*$ iff S_0 is contiguous.

Similarly, for any $x \in \mathcal{C}$, let $\ell_x = |S_x \cap \mathcal{L}|$ and $r_x = |S_x \cap \mathcal{R}|$. Let S_x^* be the set obtained from the union of the ℓ_x greatest vertices in $\mathcal{L} \setminus S_0^*$ and the r_x smallest vertices in $\mathcal{R} \setminus S_0^*$. Note that $S_0^* \cup S_x^*$ is contiguous and that $S_x = S_x^*$ iff $S_0 \cup S_x$ is contiguous.

We claim that S_0^* and the sets S_x^*, $x \in \mathcal{C}$, satisfy the desired properties.

Indeed, $(N[v_0], S_0 \cap \mathcal{L}, S_0 \cap \mathcal{R})$ is a partition of S_0 hence, $k+1 = |N[v_0]| + r + \ell$ and then $|S_0^*| = k+1$ and $N[v_0] \subseteq S_0^*$. Moreover, $|S_x^*| = \ell_x + r_x = |S_x| \leq k$. Since $N[v_0] \cup N[x] \subseteq S_0 \cup S_x$, $N[x] \cap \mathcal{L} \subseteq (S_0 \cup S_x) \cap \mathcal{L}$. Hence, $|N[x] \cap \mathcal{L}| \leq \ell + \ell_x$. Moreover, $N[x] \cap \mathcal{L}$ must be contiguous. Therefore, $N[x] \cap \mathcal{L} \subseteq (S_0^* \cup S_x^*) \cap \mathcal{L}$. By symmetry, $N[x] \cap \mathcal{R} \subseteq (S_0^* \cup S_x^*) \cap \mathcal{R}$, and then $N[v_0] \cup N[x] \subseteq S_0^* \cup S_x^*$.

Let $i > 1$ and $x \in \mathcal{C} \setminus \{v_0\}$. We have $|P_L^i(x) \setminus (S_0 \cup S_x)| \leq (i-1) \cdot k$. Moreover, $P_L^i(x) \setminus (S_0 \cup S_x) \subseteq \mathcal{L}$, hence $|P_L^i(x) \cap \mathcal{L}| \leq (i-1) \cdot k + \ell + \ell_x$. Also, $P_L^i(x) \cap \mathcal{L}$ must be contiguous, so either $P_L^i(x) \subseteq S_0^* \cup S_x^*$ in which case the result is trivial, or $(S_0^* \cup S_x^*) \cap \mathcal{L} \subseteq P_L^i(x) \cap \mathcal{L}$. In the latter case, $|P_L^i(x) \setminus (S_0^* \cup S_x^*)| = |P_L^i(x) \cap \mathcal{L}| - |(S_0^* \cup S_x^*) \cap \mathcal{L}| \leq (i-1) \cdot k$.

The other properties can be checked in a similar way. ◇

Now, we describe a k-winning connected strategy for the observer. By previous claim, we may assume that the sets S_0 and S_x ($x \in \mathcal{C}$) are contiguous.

Claim 7. *$csn(G, v_0) \leq k$.*

We define a k-winning connected strategy for the observer when the fugitive is constrained to follow induced paths. By Theorem 3, it is sufficient to prove the claim.

Initially, the observer marks the vertices in S_0. Let $P = (v_0, v_1, \cdots)$ be an induced path followed by the fugitive starting from v_0. If $v_1 \in \mathcal{C}'$ then the fugitive must stop there and loose. So assume that $v_1 \notin \mathcal{C}'$. By Lemma 11, $v_1 \in \mathcal{C} \cup \mathcal{C}_R \cup \mathcal{C}_L$ and $v_2 \in \mathcal{L} \cup \mathcal{R}$. If $v_1 \in \mathcal{C}$, at the second step, the observer marks the vertices in S_{v_1}. Then, if $v_1 \in \mathcal{C}_L$ or $v_2 \in \mathcal{L}$, then at each step (but the second one if $v_1 \in \mathcal{C}$), the observer marks the k greatest vertices unmarked in $\mathcal{L}$. Then, if $v_1 \in \mathcal{C}_R$ or $v_2 \in \mathcal{R}$, then at each step (but the second one if $v_1 \in \mathcal{C}$), the observer marks the k smallest vertices unmarked in $\mathcal{R}$. Such a strategy is connected since, at each step, the set of marked vertices is contiguous and connected to the set of previously marked vertices.

Let us show that the strategy is winning. Clearly, it marks at most k vertices at each steps, because $|S_0 \setminus \{v_0\}| \leq k$ and $|S_{v_1}| \leq k$. Let us assume $v_1 \in \mathcal{C}$ and $v_2 \in \mathcal{L}$, the other cases can be handled in a similar way. Clearly, the fugitive cannot win during the first two steps since $N[v_0] \subseteq S_0$ and $N[v_0] \cup N[v_1] \subseteq S_0 \cup S_{v_1}$. Now, after its i^{th} step, $i > 2$, the observer has marked the vertices in $S_0 \cup S_{v_1}$ and the vertices in the set M of the $(i-2) \cdot k$ greatest vertices in $\mathcal{L} \setminus (S_0 \cup S_{v_1})$. Since M is contiguous, $P_L^{i-1}(v_1) \setminus (S_0 \cup S_{v_1})$ is also contiguous and $|P_L^{i-1}(v_1) \setminus (S_0 \cup S_{v_1})| \leq (i-2) \cdot k$, we get that $P_L^{i-1}(v_1) \setminus (S_0 \cup S_{v_1}) \subseteq M$. Finally, by Lemma 12, $P^{i-1} = N[\{v_0, v_1, \cdots, v_{i-1}\}] \subseteq P_L^{i-1}(v_1)$. Therefore, $P^{i-1} \subseteq M \cup (S_0 \cup S_{v_1})$. Hence, all neighbors of the current position v_{i-1} of the fugitive are marked and the fugitive cannot escape during its next move.

Hence, $csn(G, v_0) \leq k$. ◇ □

Theorem 11. *$sn(G, v_0)$, resp., $csn(G, v_0)$, can be computed in time $O(n \cdot \Delta^3)$ in the class of n-node interval graphs with maximum degree Δ.*

Proof. By Theorem 10, it is sufficient to prove that the smallest integer k defined in Theorem 10 can be computed in polynomial time. An exhaustive check is sufficient: k being fixed, it can be be checked in polynomial-time whether k satisfies the properties. Indeed, by Theorem 10, the sets S_0 and S_x ($x \in \mathcal{C}$) that must be checked can be restricted to the contiguous sets.

In particular, since S_0 has $k+1$ vertices and must contain v_0, there are at most k such sets. Then, for any $x \in \mathcal{C}$, S_0 being fixed, there are at most k sets S_x since $S_0 \cup S_x$ must be contiguous. Moreover, given $x, x' \in \mathcal{C}$, S_x and $S_{x'}$ can be checked independently.

Hence, for any integer k, we have to check at most k sets S_0 and k sets S_x for each $x \in \mathcal{C}$. Since each test can be done in linear time in n, $\mathcal{C} \subseteq N[v_0]$ and $k \leq \Delta$. The algorithm takes at most $O(n \cdot \Delta^3)$ □

5 Conclusion and further work

Before concluding, we consider the case of an invisible fugitive. Generally, in cops and robber games, both visible and invisible robbers are difficult to handle. In this game, the invisible case is trivial. In the case of an invisible fugitive, a winning k-strategy for the observer is a sequence $(X_1, \cdots, X_r)$ of subsets of vertices of G such that $|X_i| \leq k$ for any $i \leq r$ and for any walk W starting from v_0 and of length i followed by the fugitive, $W \subseteq \bigcup_{j<i} X_j \cup \{v_0\}$. The strategy is connected if $\bigcup_{j<i} X_j \cup \{v_0\}$ induces a connected subgraph for any $i \leq r$. Let $vsn(G, v_0)$ ($cvsn(G, v_0)$) be the smallest k such that there exists a (connected) winning k-strategy for the observer. It is straightforward that:

Theorem 12. *For any connected (di)graph G and $v_0 \in V(G)$, $vsn(G,v_0) = cvsn(G,v_0)$ and equals the smallest k such that, for any $i \geq 1$, $|V_i| \leq ki+1$ where V_i is the set of vertices at distance at most i from v_0. Moreover, it can be computed in linear time.*

To conclude, we recall the questions we ask throughout the paper and add some new questions:

- Does there exist a constant bounding the ratio (resp., the difference) between *csn* and *sn* in any graph?
- What is the complexity of computing the surveillance number in the class of graphs with maximum degree 4? With bounded degree? With bounded treewidth?
- Does there exists a constant $c < 2$ and an algorithm that computes $sn(G,v_0)$ in time $O(c^n)$ in general graphs G?
- Is that true that, for any graph G and $v_0 \in V(G)$, $sn(G,v_0) = \max_S \lfloor \frac{|N[S]|}{|S|} \rfloor$ where S is taken among all subsets of $V(G)$ containing v_0 and inducing a connected subgraph?

We believe that the connected version of the game is interesting since it is closer to the more realistic *online* version of the prefetching problem. In an online version, the observer has no global knowledge of the graph anymore but discovers progressively the neighbors of the vertices she marks.

References

[AF84] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8:1–12, 1984.

[Als04] Brian Alspach. Searching and sweeping graphs: a brief survey. In *Le Matematiche*, pages 5–37, 2004.

[AM10] N. Alon and A. Mehrabian. On a generalization of meyniel's conjecture on the cops and robbers game. Research report, 2010.

[And86] T. Andreae. On a pursuit game played on graphs for which a minor is excluded. *J. Comb. Theory, Ser. B*, 41(1):37–47, 1986.

[BCP10] A. Bonato, E. Chiniforooshan, and P. Pralat. Cops and robbers from a distance. *Theor. Comput. Sci.*, 411(43):3834–3844, 2010.

[BKL08] B. Bollobas, G. Kun, and I. Leader. Cops and robbers in random graphs. Research report, 2008. arXiv:0805.2709v1.

[CCNV11] J. Chalopin, V. Chepoi, N. Nisse, and Y. Vaxès. Cop and robber games when the robber can hide and ride. *SIAM J. Discr. Math.*, 25(1):333–359, 2011.

[FGG+10] F. V. Fomin, S. Gaspers, P. A. Golovach, D. Kratsch, and S. Saurabh. Parameterized algorithm for eternal vertex cover. *Inf. Proc. Lett.*, 110(16):702–706, 2010.

[FGH+08] F. V. Fomin, P. A. Golovach, A. Hall, M. Mihalák, E. Vicari, and P. Widmayer. How to guard a graph? In *19th International Symposium on Algorithms and Computation (ISAAC)*, volume 5369 of *LNCS*, pages 318–329. Springer, 2008.

[FGK+10] F. V. Fomin, P. A. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theor. Comput. Sci.*, 411(7-9):1167–1181, 2010.

[FGL09] F. V. Fomin, P. A. Golovach, and D. Lokshtanov. Guard games on graphs: Keep the intruder out! In *7th International Workshop on Approximation and Online Algorithms (WAOA)*, volume 5893 of *LNCS*, pages 147–158. Springer, 2009.

[FGL10] F. V. Fomin, P. A. Golovach, and D. Lokshtanov. Cops and robber game without recharging. In *12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, volume 6139 of *Lecture Notes in Computer Science*, pages 273–284, Bergen, Norway, June 2010. Springer.

[Fra87] P. Frankl. Cops and robbers in graphs with large girth and cayley graphs. *Discrete Applied Mathematics*, 17:301–305, 1987.

[FT08] Fedor V. Fomin and Dimitrios M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.

[GCD02] Romulus Grigoras, Vincent Charvillat, and Matthijs Douze. Optimizing hypervideo navigation using a Markov decision process approach. In *ACM Multimedia*, pages 39–48, 2002.

[GJ90] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[GK08] J. L. Goldwasser and W. Klostermeyer. Tight bounds for eternal dominating sets in graphs. *Discrete Mathematics*, 308(12):2589–2593, 2008.

[GR95] A. S. Goldstein and E. M. Reingold. The complexity of pursuit on a graph. *Theor. Comput. Sci.*, 143(1):93–112, 1995.

[JG97] Doug Joseph and Dirk Grunwald. Prefetching using Markov predictors. In *ISCA*, pages 252–263, 1997.

[Meh11] A. Mehrabian. Lower bounds for the cop number when the robber is fast. *Combinatorics, Probability & Computing*, 20(4):617–621, 2011.

[MJM10] O. Morad and A. Jean-Marie. Optimisation en temps-réel du téléchargement de vidéos. In *Proceedings of 11th Congress of the French Operations Research Society (ROADEF)*, 2010.

[NW83] R. J. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43:235–239, 1983.

[phd] `http://www.phdcomics.com/comics/archive.php?comicid=1456`.

[Qui83] A. Quilliot. *Problèmes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes*. Thèse de doctorat d'état, Université de Paris VI, France, 1983.

[Sav70] W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970.

[Sch01] B. S. W. Schröder. The copnumber of a graph is bounded by $\lfloor \frac{3}{2} genus(g) \rfloor + 3$. *Categorical perspectives (Kent, OH, 1998), Trends in Maths.*, pages 243–263, 2001.

Centre de recherche INRIA Sophia Antipolis – Méditerranée
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399