To Satisfy Impatient Web surfers is Hard

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

To Satisfy Impatient Web surfers is Hard*

F.V. Fomin[1], F. Giroire[2], A. Jean-Marie[3], D. Mazauric[2], and N. Nisse[2]

[1] Department of Informatics, University of Bergen, Norway.
[2] MASCOTTE, INRIA, I3S(CNRS/Univ. Nice Sophia Antipolis), France.
[3] MAESTRO, INRIA and LIRMM, Univ. Montpellier 2, France.

Abstract. Prefetching is a basic mechanism for faster data access and efficient computing. An important issue in prefetching is the tradeoff 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 surfer 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 that ensures that all documents accessed by the Web surfer have previously been loaded in the cache.

We model this problem as a two-players game similar to Cops and Robber Games in graphs. The first player, a *fugitive*, starts on a marked vertex of a (di)graph G. The second player, an *observer*, marks $k \geq 1$ vertices, then the fugitive moves along one edge/arc of G to a new vertex, then the observer marks k vertices, etc.
The observer wins if he prevents the fugitive to reach an unmarked vertex. The fugitive wins otherwise, i.e., if she succeed to enter an unmarked vertex. The *surveillance number* of a (di)graph is the minimum $k \geq 1$ allowing the observer to win against any strategy of the fugitive.

We study the computational complexity of the game. 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 computing surveillance numbers of trees and interval graphs. Moreover, in the case of trees, we establish a combinatorial characterization, related to isoperimetry, of the surveillance number.
Keywords: Prefetching, Cops and robber games, PSPACE-complete.

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

* Due to lack of space, some proofs have been omitted or sketched, and can be found in [3]. This work has been done during the visit of Fedor V. Fomin at the INRIA team-project MASCOTTE, INRIA Sophia-Antipolis, France.

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 [1]. For this reason, link prefetching has been proposed as a draft Internet standard by Mozilla [7]. 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 [8], 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 [6, 9]. 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 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 [6, 9].

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).

Our results. We model the above prefetching problem as a Cop and Robber game (e.g., see [4, 2]). Using this framework, we prove that deciding whether the surveillance number of a chordal graph equals 2 is NP-hard. 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 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.

2 Preliminaries

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. For $S \subseteq V$, 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 $N[\{v\}]$.

2.1 The Surveillance Game

The *surveillance problem* deals with the following two players game in an n-node (di)graph $G = (V, E)$ with a given starting vertex $v_0 \in V$. 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 reaches an unmarked vertex and the observer wins otherwise. That is, 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 $M \subseteq V$, $f \in M$. The *surveillance number* of G, denoted by $sn(G, v_0)$, is the least k such that there is a winning k-strategy in G starting from v_0.

2.2 Restriction to induced paths

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. Due to lack of space, the proof of Theorem 1 is omitted and can be found in [3].

Theorem 1. *For any (di)graph G, $v_0 \in V(G)$, $sn(G, v_0) = msn(G, v_0)$ [3].*

In other words, if the fugitive follows induced paths and moves at every step, 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 obeys these restrictions.

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

Proof. We use a reduction from the 3-Hitting Set Problem. In the 3-Hitting Set Problem, we are given a set $\mathcal{I}$ of elements, a set $\mathcal{S}$ of subsets of size 3 of $\mathcal{I}$ and $k \in \mathbb{N}$ as an input. The question is to decide whether there exists a set $H \subseteq \mathcal{I}$ 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 [5].

Let $(\mathcal{I} = \{e_1, \cdots, e_n\}, \mathcal{S} = \{S_1, \cdots, S_m\})$ 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 each $i \leq n$ 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 $H \subseteq \mathcal{I}$ of size k such that $H \cap S \neq \emptyset$ for all $S \in \mathcal{S}$, then $sn(G, v_0) \leq 2$. 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 $e_1, \ldots, e_n$ 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 winning 2-strategy.

Now, assume that, for any $H \subseteq \mathcal{I}$ 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 either would have won earlier, or manage to reach 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 vertex v_{m+k-2}, there is $S_i \in V(K_m)$ with $H \cap S_i = \emptyset$, i.e., all three neighbors of S_i are unmarked. Then, the fugitive goes to S_i. The observer marks at most 2 of the neighbors of S_i, and the fugitive can reach an unmarked vertex. Hence, $sn(G, v_0) > 2$. □

The proof of the next Theorem is similar to the previous one. It is omitted due to lack of space and can be found in [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. A split graph is chordal.

Theorem 3. *The problem of deciding whether $sn(G, v_0) \leq k$ is NP-hard in split graphs (k is part of the input). Moreover, in this class of graphs, the game consists of at most 2 steps [3].*

For a set of boolean variables $x_0, y_0, x_1, y_1, \cdots, x_n, y_n$ and a boolean formula $F = C_1 \wedge \cdots \wedge C_m$, ($C_j$ is a 3-clause), 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 [5]. Due to lack of space, we sketch the proof of next theorem. The proof of Theorem 4 can be found in [3].

Theorem 4. *The problem of deciding whether $sn(G, v_0) \leq 4$ is PSPACE-complete in Directed Acyclic Graphs [3].*

Sketch of the 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 $0 \leq i \leq n$, $k \leq m-1$,

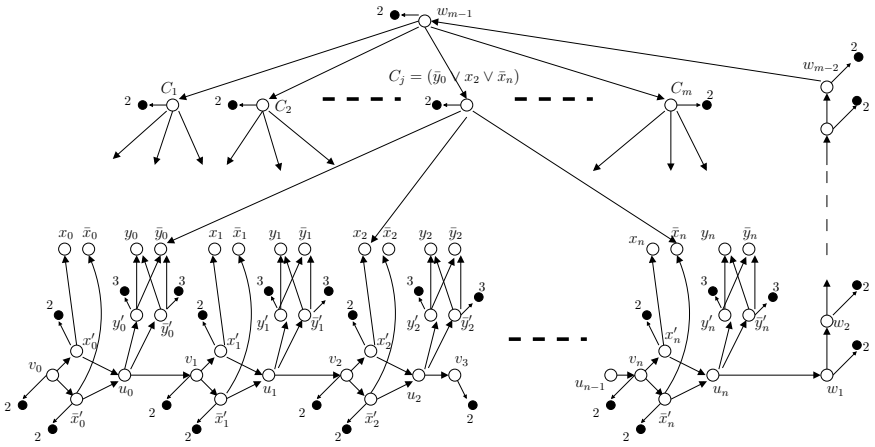

Fig. 1. Reduction in proof of Th. 4. A small black node with an integer i beside and that is the out-neighbor of a vertex v corresponds to i leaves that are in $N^+(v)$.

$j \leq m$ 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 is depicted in Figure 1.

Since $|N^+(v_0)| = 4$, $sn(D, v_0) \geq 4$ and the first step of the observer, allowed to mark 4 vertices per step, consists in marking the 4 out-neighbors of v_0. We sketch the proof that $sn(D, v_0) = 4$ if and only if Φ is true.

In [3], we show that the only way for the fugitive to win against an observer who can mark 4 vertices at each step is by following the path $P = (v_0, f_0, u_0, \cdots, v_i, f_i, u_i, \cdots, v_n, f_n, u_n, w_1, w_2, \cdots, w_{m-1})$, where $f_i \in \{x'_i, \bar{x}'_i\}$ for any $0 \leq i \leq n$. Moreover, during this game, the observer must have marked a_i where $a_i = x_i$ if $f_i = x'_i$ and $a_i = \bar{x}_i$ if $f_i = \bar{x}'_i$ (otherwise the fugitive would have won before by going to a_i). On the other hand, we prove that, during the game, the observer can have marked exactly one vertex b_i in $\{y_i, \bar{y}_i\}$. Finally, after the $(3n+m)$-th step of the observer, the fugitive stands on w_{m-1}, all vertices in $H = \{C_1, \cdots, C_m\}$ are marked while the set of marked vertices in the out-neighbors of H is exactly $\{a_0; b_0, \cdots, a_n, b_n\}$. 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.

Hence, deciding whether $sn(G, v_0) \leq 4$ is PSPACE-hard in DAGs. The proof that this problem is in PSPACE can be found in [3]. □

The following theorem provides an exponential algorithm computing $sn(G, v_0)$. Here, 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 5. *$sn(G, v_0)$ 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 $v_0, f \in M \subseteq V(G)$, $N[f] \subseteq M$ 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 $|V(\mathcal{G})| \leq n \sum_{i=1}^{\lceil \frac{n-1}{k} \rceil} \binom{n}{ki+1} \leq 2^n n$ and that the amount of arcs in $\mathcal{G}$ is $O^*(2^n)$.

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 other configurations are labeled with ∞. Iteratively, a configuration (M, f) with $|M| = ki + 1$ is labeled i if, for any $f' \in N_G(f)$, then $f' \in M$ and there is an out-neighbor (M', f') of (M, f) and that is labeled at most $i + 1$. We show that $sn(G, v_0) \leq k$ if and only if 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. By reverse induction on i, the labelling process labels $(M(W), v_i)$ with $i + 1$. This shows that $(\{v_0\} \cup \sigma(\{v_0\}, v_0), v_0)$ is labeled with 1.

For each k, the algorithm runs in time proportional to the size of $\mathcal{G}$, i.e. $2^n nk$, and thus the total running time of the algorithm is $O^*(2^n)$. □

4 Polynomial-time Algorithms in some graph classes

In this section, we give polynomial-time algorithms to compute the surveillance number of trees and interval graphs.

4.1 Keeping tree under surveillance

We first present a polynomial-time algorithm to compute $sn(T, v_0)$ 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 1. *Let T be a tree rooted in v_0, $f_k(v_0) = 0$ iff $sn(T, v_0) \leq k$.*

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, 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 $\leq k$ vertices and wins. 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 has marked 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 rooted in v_0 and of height $h + 1$, we show 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. 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 wins in T_i against an observer marking $\leq k$ vertices per step.

In T, if $f_k(v_0)$ vertices can be marked initially in $V(T) \setminus \{v_0\}$, then a 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 $|J| \leq 1 + f_k(v_0) + k$. 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.

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 6. *For any tree T rooted in v_0, $sn(T, v_0)$ can be computed in time $O(n \cdot \log n)$.*

We now give a combinatorial characterization of $sn(T, v_0)$.

Lemma 2. *For any tree T rooted in v_0 and $k < sn(T, v_0)$, 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 1, $f_k(v_0) > 0$. 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, $f_k(v) = 0$.

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, $f_k(v_0) = d - k + \sum_{1 \leq i \leq r} f_k(v_i) = d - k + \sum_{1 \leq i \leq r}(|N[S_i]| - 1 - k|S_i|) = d - k + (|N[S]| - 1 - (d - r)) - r - k(|S| - 1) = |N[S]| - 1 - k|S|$. □

Lemma 3. *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 1, 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, we mark the vertices in $\sigma(t)$.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 $V(T) \backslash \{v_0\}$ 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 7. *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 subtrees S of T containing v_0.*

4.2 To keep an Interval Graph under surveillance

An *interval graph* G is the intersection graph of a set of real intervals. The proof of the following theorem is omitted and can be found in [3].

Theorem 8. *$sn(G, v_0)$ can be computed in time $O(n \cdot \Delta^3)$ in the class of n-node interval graphs with maximum degree Δ. [3]*

5 Conclusion and further work

In [3], we define a variant of the surveillance game by introducing an extra natural constraint. In the *connected* variant of the surveillance game, the observer is constrained to mark only vertices that have neighbors already marked, i.e., the set of marked vertices must always induce a connected subgraph. We then define $csn(G, v_0)$ as the smallest k such that there is a winning connected k-strategy in G when the fugitive starts from v_0.

In [3], we show that there are graphs G and starting vertex v_0 for which $csn(G, v_0) > sn(G, v_0)$. However, we prove that all results of this paper hold for the connected variant. In particular, in any graph G that is an interval graph or a tree, and for any $v_0 \in V(G)$, $csn(G, v_0) = sn(G, v_0)$. Moreover, in all graphs used for the complexity reductions in this paper, the surveillance number equals its connected counterpart.

The connected version of the game seems 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.

To conclude, we ask some open 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 $v_0 \in V(G)$, $sn(G, v_0) = \max_S \left\lceil \frac{|N[S]|-1}{|S|} \right\rceil$ where S is taken among all subsets of $V(G)$ containing v_0 and inducing a connected subgraph?

References

1. `http://www.phdcomics.com/comics/archive.php?comicid=1456`.
2. B. Alspach. Searching and sweeping graphs: a brief survey. In *Le Matematiche*, pages 5–37, 2004.
3. F.V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To satisfy impatient web surfers is hard. Technical Report INRIA-7740, INRIA, 2011. `http://hal.inria.fr/inria-00625703/fr/`.
4. F.V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.
5. 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.
6. R. Grigoras, V. Charvillat, and M. Douze. Optimizing hypervideo navigation using a Markov decision process approach. In *ACM Multimedia*, pages 39–48, 2002.
7. Zona Research Inc. The economic impacts of unacceptable web-site download speeds. White paper, Redwood City, CA, April 1999. `www.webperf.net/info/wp_downloadspeed.pdf`.
8. D. Joseph and D. Grunwald. Prefetching using Markov predictors. In *ISCA*, pages 252–263, 1997.
9. O. Morad and A. Jean-Marie. Optimisation en temps-réel du téléchargement de vidéos. In *Proc. of 11th Congress of the French Operations Research Soc.*, 2010.