Further results on the Hunters and Rabbit game through monotonicity

Thomas Dissaux, Foivos Fioravantes, Harmender Galhawat, Nicolas Nisse

Further results on the Hunters and Rabbit game through monotonicity

Thomas Dissaux[1], Foivos Fioravantes[1,2], Harmender Galhawat[3], and Nicolas Nisse[1]

[1]Université Côte d'Azur, Inria, CNRS, I3S, France
[2]Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic
[3]Ben-Gurion University, Beersheba, Israel

February 28, 2023

Abstract

The Hunters and Rabbit game is played on a graph G where the Hunter player shoots at k vertices in every round while the Rabbit player occupies an unknown vertex and, if it is not shot, must move to a neighbouring vertex after each round. The Rabbit player wins if he can ensure that its position is never shot. The Hunter player wins otherwise. The hunter number $h(G)$ of a graph G is the minimum integer k such that the Hunter player has a winning strategy (i.e., allowing him to win whatever be the strategy of the Rabbit player). This game has been studied in several graph classes, in particular in bipartite graphs (grids, trees, hypercubes...), but the computational complexity of computing $h(G)$ remains open in general graphs and even in more restricted graph classes such as trees. To progress further in this study, we propose a notion of monotonicity (a well-studied and useful property in classical pursuit-evasion games such as graph searching games) for the Hunters and Rabbit game imposing that, roughly, a vertex that has already been shot "must not host the rabbit anymore". This allows us to obtain new results in various graph classes.

More precisely, let the monotone hunter number $mh(G)$ of a graph G be the minimum integer k such that the Hunter player has a monotone winning strategy. We show that $pw(G) \leq mh(G) \leq pw(G) + 1$ for any graph G with pathwidth $pw(G)$, which implies that, unless $\mathsf{P} = \mathsf{NP}$, there does not exist a polynomial-time algorithm for computing $mh(G)$. Then, we show that $mh(G)$ can be computed in polynomial time in split graphs, interval graphs, cographs and trees. These results go through structural characterizations which allow us to relate the monotone hunter number with the pathwidth in some of these graph classes. In all cases, this allows us to specify the hunter number or to show that there may be an arbitrary gap between h and mh, i.e., that monotonicity does not help. In particular, we show that, for every $k \geq 3$, there exists a tree T with $h(T) = 2$ and $mh(T) = k$. We conclude by proving that computing h (resp., mh) is FPTparameterized by the minimum size of a vertex cover.

1 Introduction

The Hunters and Rabbit game is played on a graph G and with a fixed integer k (the number of hunters), where the Hunter player shoots at k vertices in every round while the Rabbit player occupies an unknown vertex and, if it is not shot, must move to a neighbouring vertex after each round. The Rabbit player wins if he can ensure that its position is never shot. The Hunter

player wins otherwise. The HUNTERS AND RABBIT game was first introduced in [8], in the case $k = 1$, where it was shown that the Hunter player wins in a tree T if and only if T does not contain as subgraph any tree obtained from a star with 3 leaves by subdividing each edges twice. This result was also observed in [18], where the authors also consider the minimum number of rounds needed for the Hunter player to win. The version where $k > 1$ was first considered in [1]. Observe that, if $k = |V(G)| - 1$, the Hunter player can win in any graph G (in two rounds) by shooting twice a subset of k vertices of G. Hence, let the *hunter number* of G, denoted by $h(G)$, be the minimum integer k such that k hunters can win in G whatever be the rabbit strategy. The exact value of $h(G)$ has been determined for several specific families of graphs G. For any $n \geq 2$, $h(P_n) = 1$ where P_n is the path with n vertices [1] (because the rabbit is forced to move at every round, $h(P_1) = 0$). For any $n \geq 3$, $h(C_n) = 2$ and $h(K_n) = n - 1$, where C_n and K_n are the cycle and complete graph on n vertices respectively [1]. Moreover, $h(G_{n\times m}) = \lfloor \frac{\min\{n,m\}}{2} \rfloor + 1$ [1] and $h(Q^n) = 1 + \Sigma_{i=0}^{n-2} \binom{i}{\lfloor i/2 \rfloor}$ [6], where $G_{n\times m}$ is the $n \times m$ grid and Q^n is the hypercube with dimension n. By taking advantage of the bipartiteness of trees, it was proven that, for any tree T, $h(T) \leq \lceil \frac{1}{2} \log_2(|V(T)|) \rceil$ [15]. Surprisingly, the computational complexity of the problem that takes a graph G and an integer k as inputs and aims at deciding whether $h(G) \leq k$ is still open, even if G is restricted to be a tree.

In this paper, we progress further in this research direction by exhibiting new classes of graphs G where $h(G)$ can be determined in polynomial time. We also define some *monotone* variants of the game which allow us to get new results on the initial game.

Graph searching games. The HUNTERS AND RABBIT game takes place in the larger class of Graph Searching games initially introduced in [7, 25]. In these pursuit-evasion games, one player plays with a team of searchers (also called cops, hunters, etc.) that must track a fugitive (or robber, rabbit, etc.) moving in a graph. There are many games that can fall under this framework, each one specifying its own rules on, for example, the available moves of the searchers, the speed of the fugitive, whether the fugitive is visible or not, and so on. Several variations of graph searching games have been studied in the literature due to their numerous applications in artificial intelligence [20], robot motion planning [9], constraint satisfaction problems and database theory [14], and distributed computing [24]. Graph Searching games have mostly been studied for their significant implications in graph theory and algorithms. In particular, many variants of these games provide algorithmic interpretations of several width measures of graphs like treewidth [26], pathwidth [25], tree-depth [13], hypertree-width [2], cycle-rank [13], and directed tree-width [21]. The connection between Graph Searching games and structural parameters, such as the treewidth or the pathwidth, is based on the notion of *monotonicity* [4, 26, 23, 19]. In short, a searchers' strategy is *monotone* if it ensures that the fugitive can never "recontaminate" a vertex, i.e., it can never access a vertex that has already been "visited" (or "searched") by a searcher. The main question is then, given a game, whether "recontamination does not help in this game" [22], i.e., whether there always exists, in this game, an optimal (in terms of number of searchers) monotone winning strategy for the searchers. In particular, the monotonicity played a central role in the proof that the minimum number of searchers to capture an invisible (resp., visible) fugitive in the node-searching game played in a graph G equals its pathwidth plus one [4] (resp., treewidth plus one [26]).

Not surprisingly, the HUNTERS AND RABBIT game has also a close relationship with the pathwidth of graphs. Precisely, the hunter number of any graph is at most its pathwidth plus one [1]. In this paper, we investigate further this relationship and, for this purpose, we define a notion of monotonicity adapted to the HUNTERS AND RABBIT game and study the monotone variant of the game.

Our contribution. In Section 2, we first give the main notation and definitions used through-

out this paper, and we prove (or recall from previous works) several basic properties of the hunter number of graphs. In Section 3, we introduce the notion of monotonicity for the Hunters and Rabbit game. As discussed in Section 3, some peculiar behaviors of the Hunters and Rabbit game makes the definition of monotone hunter strategies not as straightforward as in classical Graph Searching games. We then prove, in Section 3.1, some technical properties (used later) of the monotone hunter number $mh(G)$ of a graph G, i.e., the minimum number of hunters needed by a monotone strategy to win against the rabbit whatever it does in G. In Section 3.2, we prove that $mh(G) \in \{pw(G), pw(G)+1\}$ in any graph G, which implies that, unless $\mathsf{P} = \mathsf{NP}$, there does not exist any polynomial-time algorithm to compute $mh(G)$ in any graph G. On the positive side, we give polynomial-time algorithms to determine $h(G)$ and/or $mh(G)$ in particular graph classes G in Section 4. Precisely, in Section 4.1, we show that $\omega(G) \leq h(G) \leq mh(G) \leq \omega(G)+1$ in any split graph G with maximum clique of size $\omega(G)$ and precisely characterise when each bound is reached. We also precisely characterise $mh(G)$ for any interval graph G. In Section 4.2, we design a linear-time algorithm that computes $mh(G)$ for any cograph G and give bounds for $h(G)$ in that case. In Section 4.3, we adapt the Parsons' Lemma [25] to the case of the monotone Hunters and Rabbit game which leads to a polynomial-time algorithm that computes $mh(T)$ for any tree T. In Section 5, we investigate the monotonicity property in the case of the "bipartite" variant of the Hunters and Rabbit game (see [1, 15]). In particular, this allows us to show that, for any $k \in \mathbb{N}$, there exist trees T such that $h(T) = 2$ and $mh(T) \geq k$. That is, "recontamination helps a lot" in the Hunters and Rabbit game. Finally, in Section 6, we show as a general positive result that the problem of deciding if $h(G) \leq k$, for some given integer k, is in FPT when parameterised by the vertex cover number of G. This is done through kernelisation. We close our study by providing directions for further research in Section 7.

2 Preliminaries

Unless mentioned otherwise, in this paper we will always deal with graphs $G = (V, E)$ that are non empty, finite, undirected, connected and simple. For any two adjacent vertices $x, y \in V$, let $xy \in E$ denote the edge between x and y. Given a set $S \subseteq V$, let $G[S]$ denote the subgraph of G induced by (the vertices in) S and let $G \setminus S$ denote the subgraph $G[V \setminus S]$. For any $v \in V$ and $X \subseteq V$, let $N_X(v) = \{u \in X \mid uv \in E\}$ be the *open neighbourhood* of v in X and let the *closed neighbourhood* of v in X be $N_X[v] = (N_X(v) \cup \{v\}) \cap X$. If $X = V$, we simply write $N(v)$ and $N[v]$ respectively. For any $S \subseteq V$, let $N(S) = \bigcup_{v \in S} N(v) \setminus S$ and $N[S] = N(S) \cup S$. The degree $d(v) = |N(v)|$ is the number of neighbours of v and let $\delta(G) = \min_{v \in V} d(v)$. An *independent set* of a graph $G = (V, E)$ is a subset I of V such that, for every $u, v \in I$, $uv \notin E$. A graph is *bipartite* if its vertex-set can be partitioned into two independent sets.

Hunters and Rabbit game. The Hunters and Rabbit game is played between two players, Hunter and Rabbit, on a non empty, finite, undirected, connected and simple graph $G = (V, E)$. Let $k \in \mathbb{N}^*$. The Hunter player controls k hunters and the Rabbit player controls a single rabbit. First, the Rabbit player places the rabbit at a vertex $r_0 \in V$. The rabbit is *invisible*, that is, the position of the rabbit is not known to the hunters. Then, the game proceeds in *rounds*. In each round $i \geq 1$, first, the Hunter player selects a non empty subset $S_i \subseteq V$ of at most k vertices of G (we say that the vertices in S_i are *shot* at round i). If the current position r_{i-1} of the rabbit is shot, i.e., if $r_{i-1} \in S_i$ (we say that the rabbit is shot), then the Hunter player wins, and the game stops. Otherwise, the rabbit must move from its current position r_{i-1} to a vertex $r_i \in N(r_{i-1})$, and the next round starts. The Rabbit player wins if they avoid being shot forever.

A *hunter strategy* in $G = (V, E)$ is a finite sequence $\mathcal{S} = (S_1, \ldots, S_\ell)$ of non empty subsets of vertices of G. Let $h(\mathcal{S}) := \max_{1 \leq i \leq \ell} |S_i|$ and let us say that $\mathcal{S}$ *uses* $h(\mathcal{S})$ hunters. A *rabbit trajectory in G starting from $W \subseteq V$* (W will always be assumed non empty) is any walk $(r_0, \ldots, r_\ell)$ starting from W, *i.e.*, $r_0 \in W$ and $r_i \in N_V(r_{i-1})$ for every $1 \leq i \leq \ell$. A hunter strategy is *winning with respect to W* if, for every rabbit trajectory $(r_0, \ldots, r_\ell)$ starting from W, there exists $0 \leq j < \ell$ such that $r_j \in S_{j+1}$, that is, the rabbit is eventually shot whatever be its trajectory starting from W. Given a hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$, a rabbit trajectory $(r_0, \ldots, r_\ell)$ starting from W is *winning against $\mathcal{S}$* if $r_i \notin S_{i+1}$ for every $0 \leq i < \ell$. A *winning hunter strategy* is any winning hunter strategy with respect to V and a *rabbit trajectory* is any rabbit trajectory starting from V.

The *hunter number of $G = (V, E)$ with respect to $W \subseteq V$*, denoted by $h_W(G)$, is the minimum integer k such that there exists a winning hunter strategy with respect to W and using k hunters. Let $h(G) = h_V(G)$ be the *hunter number* of G. Note that, for technical reasons, for a single vertex graph G, we set $h(G) = 0$. This goes in accordance with "the locating part" of the game since the rabbit is already located. The Rabbit player has a *strategy $\mathcal{R}$ starting from $W \subseteq V$ against $k \geq 1$ hunters* if, for every hunter strategy $\mathcal{S}$ using k hunters, there exists a rabbit trajectory $\mathcal{R}(\mathcal{S})$ that is winning against $\mathcal{S}$. Note that, if such a strategy $\mathcal{R}$ exists, then $h_W(G) > k$.

The following lemmas will be used throughout this paper. In [1], it is shown that the hunter number is closed under taking subgraphs. We first show that this result trivially extends to the case when the starting positions of the rabbit are restricted.

Lemma 1. *Let $G = (V, E)$ be any graph and let H be a subgraph of G, and let $W \subseteq V$ with $W \cap V(H) \neq \emptyset$. Then, $h_{W \cap V(H)}(H) \leq h_W(G) \leq h(G)$.*

Proof. By definition, $h_W(G) \leq h(G)$. Let us show the other inequality.

Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a winning hunter strategy in G with respect to W. Let $\mathcal{S}' = (S'_1, S'_2, \ldots, S'_\ell)$ be such that, for every $1 \leq i \leq \ell$, $S'_i = S_i \cap V(H)$ if $S_i \cap V(H) \neq \emptyset$ and S'_i consists of any vertex of $V(H)$ otherwise. Then, $\mathcal{S}'$ is a winning hunter strategy in H with respect to $W \cap V(H)$. Indeed, any rabbit trajectory $(r_0 \in W \cap V(H), r_1, \ldots, r_\ell)$ in H is also a trajectory starting from W in G. Since $\mathcal{S}$ is winning w.r.t. W, there exists $i < \ell$ such that $r_i \in S_{i+1} \cap V(H) \subseteq S'_{i+1}$, and so $\mathcal{S}'$ is winning w.r.t. $W \cap V(H)$. Moreover, $h(\mathcal{S}') \leq h(\mathcal{S})$. □

For any hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$, it will be convenient to identify the potential positions of a rabbit (starting in $W \subseteq V$) after each round. Precisely, let $\mathcal{Z}^W(\mathcal{S}) = (Z_0^W(\mathcal{S}), \ldots, Z_\ell^W(\mathcal{S}))$ be defined as follows. Let $Z_0^W(\mathcal{S}) = W$ and, for every $0 < i \leq \ell$, let $Z_i^W(\mathcal{S})$ be the set of vertices v such that there exists a rabbit trajectory $(r_0, r_1, \ldots, r_i = v)$ such that $r_0 \in W$ and, for every $0 \leq j < i$, $r_j \notin S_{j+1}$. Formally, for any $1 \leq i \leq \ell$, let $Z_i^W(\mathcal{S}) = \{x \in V(G) \mid \exists y \in (Z_{i-1}^W(\mathcal{S}) \setminus S_i) \wedge (xy \in E(G))\}$. Intuitively, $Z_i^W(\mathcal{S})$ is the set of vertices that the rabbit (starting from some vertex in W) can have reached at the end of the i^{th} round without having been shot. We will refer to the vertices in $Z_i^W(\mathcal{S})$ as the *contaminated* vertices after round i. Note that, if $\mathcal{S}$ is winning, then $Z_\ell^W(\mathcal{S}) = \emptyset$. In what follows, we write Z_i (resp., $Z_i(\mathcal{S})$) instead of $Z_i^W(\mathcal{S})$ when $\mathcal{S}$ and W (resp., when W) are clear from the context.

We now show that we can only consider hunter strategies that consist only of "useful shots". A hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ is said to be *parsimonious* if, for every $1 \leq i \leq \ell$, $S_i \subseteq Z_{i-1}(\mathcal{S})$. Note that, if $\mathcal{S}$ is parsimonious, then $Z_i \neq \emptyset$ for every $i < \ell$. Note that if $\mathcal{S}$ is parsimonious, then it can be retrieved only from the sequence $\mathcal{Z}(\mathcal{S}) = (Z_0, \ldots, Z_\ell)$ of the contaminated sets. Indeed, for any $1 \leq i \leq \ell$, $S_i = \{w \in Z_{i-1} \mid \exists x \in N(w) \setminus Z_i\}$.

In the following lemma, we establish that we can hunt the rabbit in a parsimonious manner without increasing the number of required hunters.

Lemma 2. *For any graph $G = (V, E)$ and any non empty subset $W \subseteq V$, there is a parsimonious winning hunter strategy in G with respect to W and that uses $h_W(G)$ hunters.*

Proof. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a winning hunter strategy with respect to $W \subseteq V$ using at most $k \geq 1$ hunters. Let $\mathcal{Z}(\mathcal{S}) = (Z_0(\mathcal{S}), \ldots, Z_\ell(\mathcal{S}))$ be the set of contaminated vertices for each round of $\mathcal{S}$. If there exists an integer $\ell' < \ell$ such that $Z_{\ell'}(\mathcal{S}) = \emptyset$, then $\mathcal{S} = (S_1, \ldots, S_{\ell'})$ is also a winning hunter strategy with respect to $W \subseteq V$ using at most k hunters. Hence, we may assume that $Z_i(\mathcal{S}) \neq \emptyset$ for every $0 \leq i < \ell$.

Moreover, if there exists an integer $1 \leq i \leq \ell$ such that $S_i \cap Z_{i-1}(\mathcal{S}) = \emptyset$, let h be the smallest such integer and let $v \in Z_{h-1}(\mathcal{S})$. Then, $\mathcal{S}' = (S_1, \ldots, S_{h-1}, \{v\}, S_{h+1}, \ldots, S_\ell)$ is also a winning strategy with respect to $W \subseteq V$ using at most $k \geq 1$ hunters (since $S_h \cap Z_{h-1}(\mathcal{S}) = \emptyset$). By repeating this process, we may assume that, for every $1 \leq i \leq \ell$, $S_i \cap Z_{i-1}(\mathcal{S}) \neq \emptyset$.

Let $\mathcal{S}' = (S'_1, S'_2, \ldots, S'_{\ell'})$ be such that, for every $1 \leq i \leq \ell'$, $S'_i = S_i \cap Z_{i-1}(\mathcal{S})$. It is easy to see that, for every $i \leq \ell'$, $Z_i(\mathcal{S}) = Z_i(\mathcal{S}')$, and then $\mathcal{S}'$ is parsimonious. Furthermore, $\mathcal{S}'$ is a winning hunter strategy with respect to W. Indeed, since $\mathcal{S}$ is winning w.r.t. W, for any rabbit trajectory $(r_0, r_1, \ldots, r_\ell)$, there exists an integer $j < \ell$ such that $r_j \in S_{j+1}$. Let i be the smallest such integer. By definition, $r_i \in Z_i \cap S_{i+1} = S'_{i+1}$ and so $\mathcal{S}'$ is winning w.r.t. W. Moreover, $h(\mathcal{S}') \leq h(\mathcal{S})$. □

It must be noticed that there exist graphs $G = (V, E)$ and hunter strategies $(S_1, \ldots, S_\ell)$ that are winning in G without shooting to all vertices, i.e., such that $V \setminus \bigcup_{1 \leq i \leq \ell} S_i \neq \emptyset$. For instance, in the graph G that consists of a single edge uv, the strategy $(\{u\}, \{u\})$ is a winning hunter strategy using one hunter and without shooting at v. Note that, in that example, there exists no winning parsimonious hunter strategy using one hunter and that shots to both u and v. The next lemma, that characterises the set of such unshot vertices, will be used throughout the paper.

Lemma 3. *Let H be any non-empty connected subgraph of any graph $G = (V, E)$. Let $W \subseteq V$ such that $W \cap V(H) \neq \emptyset$. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be any winning hunter strategy in G with respect to W. If $S_i \cap V(H) = \emptyset$ for all $1 \leq i \leq \ell$, then $|V(H)| = 1$.*

Proof. Let $x \in V(H) \cap W$. Towards a contradiction, assume that $|V(H)| \geq 2$. Let $y \in N_H(x)$ (it exists since H is connected). Note that since $S_i \cap V(H) = \emptyset$ for all $1 \leq i \leq \ell$, $\{x, y\} \cap \bigcup_{1 \leq i \leq \ell} S_i = \emptyset$. Thus, the rabbit can oscillate between x and y during the whole game without being shot. That is, $\mathcal{R} = (r_0 = x, r_1 = y, r_2 = x, \ldots, r_\ell)$ is a winning rabbit trajectory against $\mathcal{S}$ starting from $W \cap V(H)$. This contradicts that $\mathcal{S}$ is a winning hunter strategy in G with respect to W. □

In what follows, we will use the following result of [6]:

Lemma 4. *[6] For any graph G, $h(G) \geq \delta(G)$.*

The Hunters and Rabbit game has been particularly studied in bipartite graphs [1, 6, 15] and we continue this study in Section 5. In what follows, bipartite graphs are referred to as $G = (V_r \cup V_w, E)$ where (V_r, V_w) is implicitly a bipartition of $V(G)$ such that V_r and V_w are independent sets respectively. We refer to the vertices in V_r (resp., in V_w) as the *red* (resp., *white*) vertices.

In [1], it is shown that, in bipartite graphs, it is sufficient to consider winning hunter strategies with respect to one of the independent sets of the bipartition. For completeness and to further motivate some of our results, we briefly recall their result. Precisely:

Lemma 5. *[1] For any bipartite graph $G = (V_r \cup V_w, E)$, $h(G) = h_{V_r}(G) = h_{V_w}(G)$.*

Proof. By definition, $\max\{h_{V_r}(G), h_{V_w}(G)\} \leq h(G)$. To show that $h(G) \leq h_{V_r}(G)$ (resp., $h(G) \leq h_{V_w}(G)$), let $\mathcal{S}_r = (S_1, \ldots, S_\ell)$ be a winning hunter strategy in G with respect to V_r (resp., w.r.t. V_w). If ℓ is odd, then $(S_1, \ldots, S_\ell, S_1, \ldots, S_\ell)$ is a winning hunter strategy, and otherwise, $(S_1, \ldots, S_\ell, \{u\}, S_1, \ldots, S_\ell)$ where u is any arbitrary vertex is a winning hunter strategy. □

Note that, in most of the paper, we will consider hunter strategies with respect to V, but in section 5. More precisely, in Section 5, we will consider the Hunters and Rabbit game in bipartite graphs when the rabbit must start at some vertex of V_r. We will refer to this variant as the *red variant* of the game. The following remark will be widely used.

Remark. Let $G = (V_r \cup V_w, E)$ be a bipartite graph and $\mathcal{S}_r = (S_1, \ldots, S_\ell)$ be a parsimonious hunter strategy in G with respect to V_r. Then, for every $1 \leq i \leq \lceil \ell/2 \rceil$, $S_{2i-1} \subseteq Z_{2i-2} \subseteq V_r$ and (if $2i \leq \ell$) $S_{2i} \subseteq Z_{2i-1} \subseteq V_w$. Indeed, in a bipartite graph, if the rabbit starts at a vertex in V_r (resp., V_w), it must occupy a vertex of V_r at the end of every even (resp., odd) round and a vertex of V_w at the end of every odd (resp., even) round.

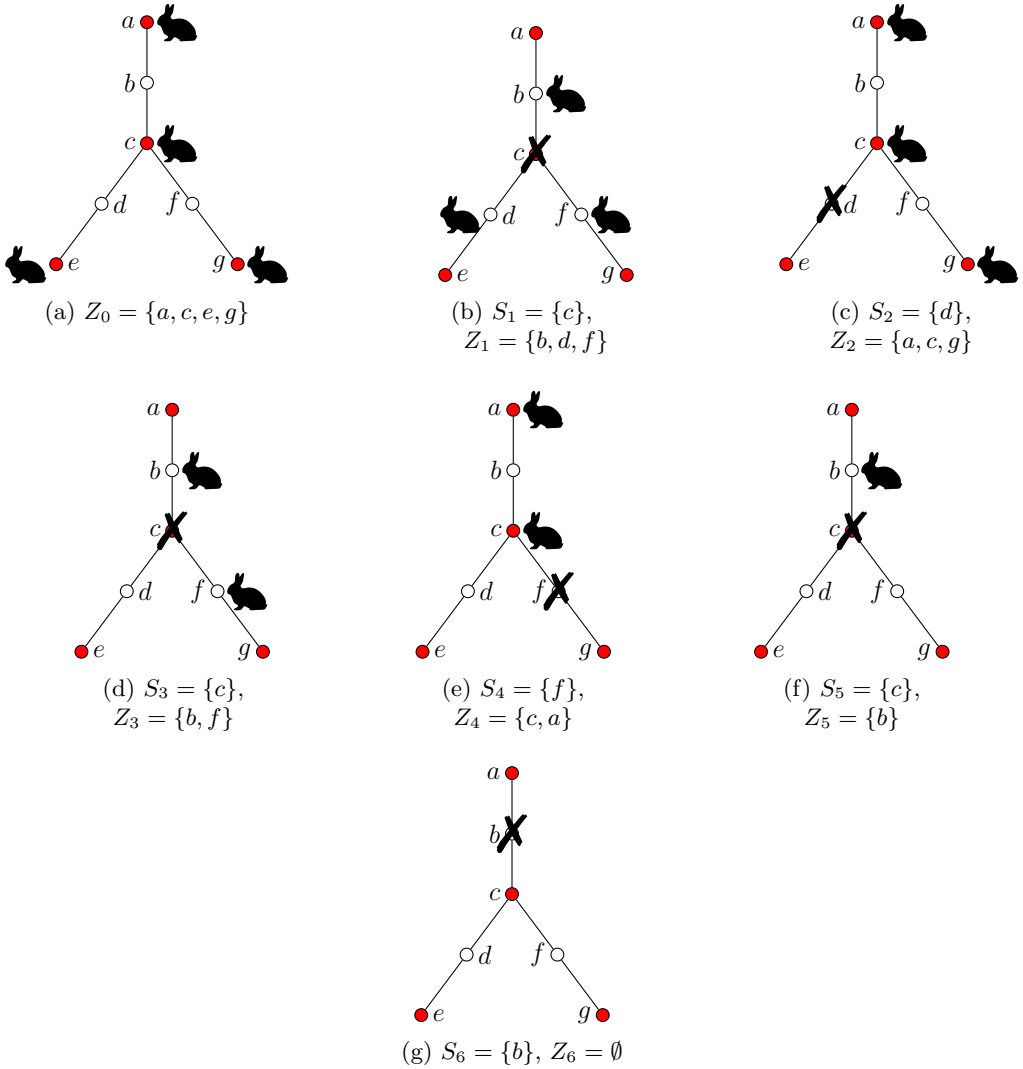
3 Monotonicity

In classical graph pursuit-evasion games, an important notion is that of *monotonicity*. Without going into the details, in these games, a strategy is *monotone* if the area reachable by the fugitive never increases. Said differently, in the particular case of graph searching games, a strategy is monotone if, once a searcher is removed from one vertex, it is never necessary to occupy this vertex during a subsequent round (note that, in some specific cases, for instance in directed graphs, these two definitions are not rigorously equivalent [3]). Monotone strategies have been widely studied [4, 28, 23] because, on the one hand, it is generally easier to design them and, on the other hand, monotone strategies have length polynomial in the size of the graph and, so, corresponding decision problems (is there a monotone strategy using k searchers?) can be proven to be in NP.

It is clear that such a definition is not suitable to the Hunters and Rabbit game. Indeed, consider the graph that consists of a single edge uv: the hunter must shoot at some vertex, say u, and, if the rabbit was at v, it will move to u, i.e., the vertex u is "recontaminated". Therefore, we propose to define monotonicity in the Hunters and Rabbit game as follows (see the formal definition below): once a vertex has been "cleared", if the rabbit can access it in a subsequent round, then the vertex must be shot immediately.

In classical graph searching games, a vertex being cleared at some round means that the searcher strategy ensures that the fugitive cannot occupy this vertex at this round. Being recontaminated can then be intuitively defined by the fact that a vertex can be reached by the fugitive while having been cleared in a previous round. This intuitive definition does not make any sense in the Hunters and Rabbit game and, in particular, in its red variant in bipartite graphs. Indeed, in such case, every red vertex is cleared at every odd round and so, looking for a strategy without recontamination would be meaningless. To overcome this difficulty, we propose to define the clearing of a vertex at some round by the fact that the actions of the hunters ensure that this vertex cannot be occupied by the rabbit at this round.

A related difficulty comes from the fact that, contrary to classical graph searching games, a vertex may be "cleared" without having been shot during the game. Recall, for instance, our previous discussion for the graph consisting of a single edge. As a less trivial example, consider a star with three leaves whose edges have been subdivided once each. Then, assuming that the leaves and the centre are red, in the red variant, it is possible for one hunter to win



(a) $Z_0 = \{a, c, e, g\}$

(b) $S_1 = \{c\}$, $Z_1 = \{b, d, f\}$

(c) $S_2 = \{d\}$, $Z_2 = \{a, c, g\}$

(d) $S_3 = \{c\}$, $Z_3 = \{b, f\}$

(e) $S_4 = \{f\}$, $Z_4 = \{c, a\}$

(f) $S_5 = \{c\}$, $Z_5 = \{b\}$

(g) $S_6 = \{b\}$, $Z_6 = \emptyset$

Figure 1: Example of a bipartite graph (where $V_r = \{a, c, e, g\}$ corresponds to the red part of the bipartition, illustrated by the red vertices in the figures) and of a parsimonious winning strategy with respect to V_r, such that no vertex in $\{a, e, g\}$ is ever shot. Each subfigure depicts the situation at the end of the corresponding round. In round 0, the rabbit occupies any vertex in $\{a, c, e, g\}$. Then, in round 1, the hunter shoots the vertex c (depicted as the cross over the corresponding vertex of subfigure (b)) and the rabbit moves to one of the vertices in $\{b, d, f\}$. The game continues until the end of round 6 (subfigure (g)), at which point the hunter is sure to shoot the rabbit in vertex b. Formally, we have $\mathcal{S} = (\{c\}, \{d\}, \{c\}, \{f\}, \{c\}, \{b\})$ and $\mathcal{Z}(\mathcal{S}) = (\{a, c, e, g\}, \{b, d, f\}, \{a, c\}, \{b\}, \{a, c\}, \{b\}, \emptyset)$.

without shooting any of the leaves (while any of the leaves may be occupied by the rabbit initially). Indeed, consider the strategy for one hunter where on every odd round he shoots on the centre and on every even round he shoots on an arbitrary neighbour of the centre that was not previously shot. Figure 1 illustrates the above strategy.

Therefore, two actions of the hunters may clear a vertex: either a hunter shoots a vertex v at round i and does not shoot the rabbit (i.e. there is no rabbit trajectory, such that $r_{i-1} = v$, that is winning against a strategy shooting at v at round i), or the hunters shoot at every

contaminated vertex in the neighbourhood of v. In this case, either v was occupied and the rabbit has to leave v, or it was not and cannot be occupied after the move of the rabbit. In both cases, $v \notin Z_i$. This discussion motivates the following definition for the monotonicity of hunter strategies.

3.1 Definition of monotone strategies and first properties

Given a graph G, a winning hunter strategy $\mathcal{S}$ in G with respect to $W \subseteq V$, is *monotone* if for every vertex $v \in V$, once v has been "cleared", then it is shot again every time the rabbit can potentially reach v. Formally, we say that a vertex v is *cleared* at round i if either $v \in S_i$ or $(N(v) \cap Z_{i-1} \neq \emptyset$ and $N(v) \cap Z_{i-1} \subseteq S_i)$. Note that, in the second condition, the fact that we require that $N(v) \cap Z_{i-1} \neq \emptyset$ comes from technicalities when $W \neq V$. A strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ is *monotone* if, for every vertex $v \in V$, if there exists an i such that v is cleared at round i, then for every $j > i$ such that $v \in Z_j$, the strategy ensures that $v \in S_{j+1}$. A vertex v is *recontaminated* at round j if there exists $i \leq j$ such that v is cleared at round i and $v \in Z_j \setminus S_{j+1}$.

The *monotone hunter number* of a graph G with respect to $W \subseteq V(G)$, denoted by $mh_W(G)$, is the minimum number k such that k hunters have a monotone winning hunter strategy in G with respect to W. Let us denote the *monotone hunter number* $mh_V(G)$ of G by $mh(G)$. Note that, by definition:

Proposition 1. *For every graph $G = (V, E)$ and $W \subseteq V$, $h_W(G) \leq mh_W(G) \leq mh(G)$.*

In this subsection, we prove some general properties of (non-)monotone strategies. Let us start with two technical claims that will be used in several proofs below.

Proposition 2. *Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a hunter strategy in a graph $G = (V, E)$. Let $v \in V$ and $1 \leq i \leq \ell$. If there exists a vertex $u \in N(v)$ and a vertex $x \in N(u)$ (possibly $x = v$) such that $u, x \notin \bigcup_{j \leq i} S_j$, then $v \in Z_p$ for each $p \leq i$.*

Proof. This clearly holds if $p = 0$ since $Z_0 = V$. If $p = 1$, there exists a rabbit trajectory $(r_0 = u \in N(v) \setminus S_1, r_1 = v)$ and so $v \in Z_1$. Hence, we assume that $p > 1$.

The rabbit can follow the following strategy depending on whether p is odd or even:

1. p is odd: The rabbit can follow the following trajectory: $(r_0 = u, r_1 = x, \ldots, r_{p-1} = u, r_p = v)$ where, for $q < p$, $r_q = u$ if q is even and $r_q = x$ if q is odd.
2. p is even: The rabbit can follow the following trajectory: $(r_0 = x, r_1 = u, \ldots, r_{p-1} = u, r_p = v)$ where, for $q < p$, $r_q = x$ if q is even and $r_q = u$ if q is odd.

In both cases, for every $0 \leq j < p$, $r_j \notin S_{j+1}$ since $p \leq i$ and $x, u \notin \bigcup_{j<i} S_j$. Therefore, $v \in Z_p$. □

The next lemma shows that, as expected, if the hunters follow a monotone strategy, the set of potential positions for the rabbit cannot increase.

Lemma 6. *Let $G = (V, E)$ be a graph with at least two vertices. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a monotone hunter strategy in G. For any $0 \leq p \leq i \leq \ell$, $Z_i \subseteq Z_p$.*

Proof. This clearly holds if $p = 0$ or $i = 1$ since $Z_0 = V$. Hence, let us assume that $p \geq 1$ and $i > 1$. Let $v \in Z_i$. Since $v \in Z_i$, there exists a rabbit trajectory $R = (r_0, \ldots, r_{i-2} = x, r_{i-1} = u, r_i = v)$ such that, for any $0 \leq j < i$, $r_j \notin S_{j+1}$. By definition of a rabbit trajectory, $u \in N(v)$ and $x \in N(u)$. Moreover, by monotonicity of $\mathcal{S}$, since $u \in Z_{i-1} \setminus S_i$ (resp. $x \in Z_{i-2} \setminus S_{i-1}$), $u \notin \bigcup_{q \leq i} S_q$ (resp. $x \notin \bigcup_{q \leq i-1} S_q$). By Proposition 2, $v \in Z_p$ for each $p \leq i$. □

The next lemma states that, for any non-monotone strategy, there must exist a vertex that has been shot at some round and that is recontaminated later (recall that it is not trivial since a vertex may be recontaminated without being previously shot).

Lemma 7. *Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a non-monotone winning hunter strategy in a graph $G = (V, E)$. Then, there exist a vertex $v \in V$ and $1 \leq i \leq \ell$ such that $v \in Z_{i-1} \setminus S_i$ and $v \in \bigcup_{p<i} S_p$.*

Proof. Towards a contradiction, assume that the statement of the lemma is false, i.e., for every vertex $v \in V$ and every $1 \leq i \leq \ell$, if $v \in Z_{i-1} \setminus S_i$ then $v \notin \bigcup_{p<i} S_p$.

Since $\mathcal{S}$ is non-monotone and winning, there exists a vertex u such that u is cleared at a round $1 \leq q \leq \ell - 2$, and then recontaminated at a round $j > q$ (i.e., $u \in Z_j \setminus S_{j+1}$). Moreover, by our assumption, u is cleared by shooting each contaminated vertex in $N(u)$ at round q, i.e., $Z_{q-1} \cap N(u) \subseteq S_q$.

Let us show that $N(u) \subseteq \bigcup_{p \leq q} S_p$. Let us assume that there exists a vertex $x \in N(u)$ such that $x \notin \bigcup_{p<q} S_p$. Since both $u, x \notin \bigcup_{p<q} S_q$ and $u \in N(x)$, by Proposition 2, we get that $x \in Z_{q-1}$. Therefore, $x \in Z_{q-1} \cap N(u) \subseteq S_q$. Hence, $N(u) \subseteq \bigcup_{p \leq q} S_p$.

Since $v \in Z_j$ then there exists $w \in (N(u) \cap Z_{j-1}) \setminus S_j$ and $w \in \bigcup_{p<j} S_p$, i.e., w satisfies the statement of the lemma, a contradiction. □

Now, let us generalise Lemmas 1 and 2 to monotone strategies.

Lemma 8. *For any non-empty connected subgraph H of a graph $G = (V, E)$, $mh(H) \leq mh(G)$. More precisely, if there exists a monotone winning hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ in G, then there exists a monotone winning hunter strategy $\mathcal{S}'$ in H using at most $\max_{1 \leq i \leq \ell} |S_i \cap V(H)|$ hunters.*

Proof. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a monotone winning hunter strategy for G.

If $|V(H)| = 1$, the result clearly holds since $mh(H) = 0$. Hence, let us assume that $|V(H)| > 1$. Let m be the minimum integer such that $S_m \cap V(H) \neq \emptyset$ and let $u \in S_m \cap V(H)$ (by Lemma 3, such an integer m exists because $|V(H)| > 1$). Let $\mathcal{S}' = (S'_1, \ldots, S'_\ell)$ be the hunter strategy, such that for every $1 \leq i \leq \ell$,

$$S'_i = \begin{cases} S_i \cap V(H), & \text{if } S_i \cap V(H) \neq \emptyset \\ \{u\}, & \text{otherwise} \end{cases}$$

First, we have the following claim.

Claim 1. *For every $0 \leq i \leq \ell$ and for any vertex $v \in V(H)$, if $v \in Z_i(\mathcal{S}')$, then $v \in Z_i(\mathcal{S})$.*

Proof of Claim. Let $\mathcal{R} = (r_0, \ldots, r_i = v)$ be a rabbit trajectory in H such that for any $0 \leq j < i$, $r_j r_{j+1} \in E(H)$, $r_j \notin S'_{j+1}$ and $r_i = v$ (such a trajectory exists since $v \in Z_i(\mathcal{S}')$). By construction of $\mathcal{S}'$, for any $1 \leq j \leq \ell$, $S_j \cap V(H) \subseteq S'_j$. Therefore, $\mathcal{R}$ is also a rabbit trajectory in G with $r_j \notin S_{j+1}$, for all $0 \leq j < i$. Thus, $v \in Z_i(\mathcal{S})$. ◇

Let us show that $\mathcal{S}'$ is a monotone winning hunter strategy in H. First, we show that $\mathcal{S}'$ is indeed a winning hunter strategy in H. Towards a contradiction, assume that $\mathcal{S}'$ is not a winning strategy in H. This implies that $Z_\ell(\mathcal{S}') \neq \emptyset$. Hence, Claim 1 implies that $Z_\ell(\mathcal{S}) \neq \emptyset$, contradicting the fact that $\mathcal{S}$ is a winning hunter strategy in G.

Thus, $\mathcal{S}'$ is a winning strategy in H. Next, we establish that $\mathcal{S}'$ is indeed monotone. Towards a contradiction, let us assume that $\mathcal{S}'$ is non-monotone. Hence, by Lemma 7, there exist $v \in V(H)$ and $1 \leq q < i \leq \ell$ such that $v \in S'_q$ and $v \in Z_i(\mathcal{S}') \setminus S'_{i+1}$. By Claim 1 and because $v \in Z_i(\mathcal{S}')$, $v \in Z_i(\mathcal{S})$. Since $S_p \cap V(H) \subseteq S'_p$ for any $1 \leq p \leq \ell$ and because $v \notin S'_{i+1}$, $v \notin S_{i+1}$.

If $v = u$, $i + 1 > m$ (since $u \in S'_p$ for all $1 \leq p \leq m$) and so $v \in S_m$ and in $Z_i(\mathcal{S}) \setminus S_{i+1}$, contradicting the monotonicity of $\mathcal{S}$.

Otherwise, $v \neq u$. By construction of $\mathcal{S}'$, $S'_p \setminus \{u\} \subseteq S_p$ for all $1 \leq p \leq \ell$. Hence, $v \in S_q$ and $v \in Z_i(\mathcal{S}) \setminus S_{i+1}$, contradicting the monotonicity of $\mathcal{S}$.

Finally, the fact that $h(\mathcal{S}') \leq \max_{1 \leq i \leq \ell} |S_i \cap V(H)| \leq h(\mathcal{S})$ completes the proof. □

Lemma 9. *For a graph $G = (V, E)$ and any $k \geq mh(G)$, there exists a parsimonious monotone winning hunter strategy in G using at most k hunters.*

Proof. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a monotone winning hunter strategy in G using at most $k \geq mh(G)$ hunters such that ℓ is minimum. If $\mathcal{S}$ is parsimonious, we are done. Otherwise, among such strategies, let us consider $\mathcal{S}$ that maximizes the first round $1 \leq j \leq \ell$ that makes $\mathcal{S}$ not parsimonious. There are several cases to be considered.

- Let $\mathcal{Z}(\mathcal{S}) = (Z_0(\mathcal{S}), \ldots, Z_\ell(\mathcal{S}))$ be the set of contaminated vertices for each round of $\mathcal{S}$. If there exists an integer $\ell' < \ell$ such that $Z_{\ell'}(\mathcal{S}) = \emptyset$, then $\mathcal{S} = (S_1, \ldots, S_{\ell'})$ is also a monotone winning hunter strategy in G using at most k hunters, contradicting the minimality of ℓ.

  Hence, we may assume that $Z_i(\mathcal{S}) \neq \emptyset$ for every $0 \leq i < \ell$.

- Let $1 \leq j \leq \ell$ be the smallest integer such that $S_j \setminus Z_{j-1}(\mathcal{S}) \neq \emptyset$ (if no such integer exists, then $\mathcal{S}$ is parsimonious and we are done). If $S_j \cap Z_{j-1}(\mathcal{S}) \neq \emptyset$, replace S_j by $S_j \cap Z_{j-1}(\mathcal{S})$. This leads to a winning monotone hunter strategy $\mathcal{S}'$ (indeed, $Z_h(\mathcal{S}) = Z_h(\mathcal{S}')$ for all $1 \leq h \leq \ell$) contradicting the maximality of j.

  Hence, we may assume that $S_j \cap Z_{j-1}(\mathcal{S}) = \emptyset$. Note that this implies that $j < \ell$ (since otherwise, $\mathcal{S}$ would not be winning).

- If any, let $0 < i$ be the minimum integer such that $S_{j+i} \cap Z_{j+i-1}(\mathcal{S}) \neq \emptyset$. Let $v \in S_{j+i} \cap Z_{j+i-1}(\mathcal{S})$. Since $v \in Z_{j+i-1}(\mathcal{S})$, by Lemma 6, $v \in Z_{j-1+i'}(\mathcal{S})$ for every $0 \leq i' < i$. Then, for every $0 \leq i' < i$, replace $S_{j+i'}$ by $\{v\}$. Let us prove that this leads to a monotone hunter strategy contradicting the maximality of j.

  Let $\mathcal{S}'$ be the strategy obtained by the above modifications. First, note that, for any $0 \leq h < j$, $S_h = S'_h$ and so $Z_h(\mathcal{S}) = Z_h(\mathcal{S}')$. By definition, $Z_j(\mathcal{S}) = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}) \setminus S_j \wedge (xy \in E)\}$ and, since $S_j \cap Z_{j-1} = \emptyset$, we get $Z_j(\mathcal{S}) = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}) \wedge (xy \in E)\}$. On the other hand, $Z_j(\mathcal{S}') = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}') \setminus S'_j \wedge (xy \in E)\} = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}) \setminus \{v\} \wedge (xy \in E)\}$ since $Z_{j-1}(\mathcal{S}') = Z_{j-1}(\mathcal{S})$ and $S'_j = \{v\}$. Hence, $Z_j(\mathcal{S}') \subseteq Z_j(\mathcal{S})$. By induction on $j \leq i' \leq \ell$ and using the same arguments, we get that $Z_{i'}(\mathcal{S}') \subseteq Z_{i'}(\mathcal{S})$ for every $j \leq i' \leq \ell$. Thus, $\mathcal{S}'$ is a winning hunter strategy in G using at most $k \geq mh(G)$ hunters (because $Z_\ell(\mathcal{S}') \subseteq Z_\ell(\mathcal{S}) = \emptyset$). It remains to show that $\mathcal{S}'$ is monotone.

  For purpose of contradiction, let us assume that $\mathcal{S}'$ is non-monotone. By Lemma 7, there exists a vertex x and $1 < m \leq \ell$ such that $x \in Z_{m-1}(\mathcal{S}') \setminus S'_m$ and $x \in \bigcup_{h<m} S'_h$.

  If $x \neq v$, then by definition of $\mathcal{S}'$ (for every $1 \leq r \leq \ell$, either $S'_r = S_r$ or $S'_r = \{v\}$) and because $Z_r(\mathcal{S}') \subseteq Z_r(\mathcal{S})$ for all $1 \leq r \leq \ell$, we get that $x \in \bigcup_{h<m} S_h$ and $x \in Z_{m-1}(\mathcal{S}) \setminus S_m$ which contradicts the monotonicity of $\mathcal{S}$. Hence, let us assume that $x = v$. Recall that we proved that $v \in Z_{j-1}(\mathcal{S}) \setminus S_j$. Therefore, by monotonicity of $\mathcal{S}$, $v \notin \bigcup_{r<j} S_r$ which implies that $v \notin \bigcup_{r<j} S'_r$. Since $v \in S'_{j+i'}$ for all $0 \leq i' \leq i$, we get that $m > j + i$. This means that $v \in Z_{m-1}(\mathcal{S}) \setminus S_m$ (because $Z_r(\mathcal{S}') \subseteq Z_r(\mathcal{S})$ for all $1 \leq r \leq \ell$ and $S'_r = S_r$ for all $r \geq m > j + 2i$) and $v \in S_{j+i}$, which contradicts the monotonicity of $\mathcal{S}$.

  Hence, we may assume that $S_{j+i'} \cap Z_{j+i'-1}(\mathcal{S}) = \emptyset$ for all $0 \leq i'$ such that $j + i' \leq \ell$.

- Let us recall that $j < \ell$ and that $Z_{j-1}(\mathcal{S}) \neq \emptyset$. Thus, let $v \in Z_{j-1}(\mathcal{S})$. So, there exists $w \in N(v) \cap Z_{j-2}(\mathcal{S}) \setminus S_{j-1}$. Moreover, since $w \in Z_{j-2}(\mathcal{S}) \setminus S_{j-1}$ (resp. $v \in Z_{j-1}(\mathcal{S}) \setminus S_j$), $w \notin S_r$ (resp. $v \notin S_r$) for any $r \leq j$ (otherwise, it contradicts the monotonicity of $\mathcal{S}$). Let us recall that for any $0 \leq i'$ such that $j + i' \leq \ell$, $S_{j+i'} \cap Z_{j+i'-1}(\mathcal{S}) = \emptyset$. Therefore, $w \notin S_r$ and $v \notin S_r$ for any $r \leq \ell$. By Proposition 2, $v \in Z_\ell(\mathcal{S})$, which contradicts that $\mathcal{S}$ is winning.

This completes the proof. □

To conclude this subsection, we give an alternative point of view of Lemma 6. In particular, we show that, if the hunters follow a monotone parsimonious strategy, after having shot at one vertex, the hunters must always shoot at this vertex until it cannot be reached by the rabbit anymore.

Lemma 10. *Let $G = (V, E)$ be a graph and $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a parsimonious monotone winning hunter strategy in G.*

- *If there exist $1 \leq i < j \leq \ell$ such that $v \in S_i \cap S_j$, then $v \in S_{i+1}$.*
- *If there exists an integer $1 \leq i < \ell$ such that $v \notin Z_{i-1}$, then $v \notin S_j$ for every $j \geq i$.*

Proof. First assume that $v \in S_i \cap S_j$. Since $\mathcal{S}$ is parsimonious, it implies that $v \in Z_{i-1} \cap Z_{j-1}$. By Lemma 6 and since $v \in Z_j$, $v \in Z_{i'}$ for all $i' < j$. Since $v \in S_i$ and $v \in Z_{p-1}$ for all $i \leq p \leq j$, by monotonicity of $\mathcal{S}$, $v \in S_p$ for all $i \leq p \leq j$.

For the second statement, the fact that $v \notin Z_{i-1}$ and Lemma 6 imply that $v \notin Z_j$ for all $j \geq i$. Since $\mathcal{S}$ is parsimonious, $v \notin S_j$ for every $j \geq i$. □

Surprisingly, the above lemma is not a characterization of monotone strategies. Indeed, consider the path (a, b, c, d) on four vertices. It can be checked that the hunter strategy $(\{a\}, \{b, c\}, \{b, c\})$ is parsimonious and winning (with respect to V) and satisfies the condition of the previous lemma, but this strategy is non-monotone (since $a \in S_1$ and $a \in Z_1 \setminus S_2$).

3.2 Monotone Hunter Number and Pathwidth

In this subsection, we relate the monotone hunter number of a graph to its pathwidth. Our result might be surprising since the pathwidth of a graph G is equivalent to the number of searchers required to (monotonously) capture an arbitrary fast invisible fugitive [4] while, in our case, the invisible rabbit seems much weaker than the fugitive: the rabbit is "slow" (it moves only to neighbours) and constrained to move at every round. In this view, we might guess that the monotone hunter number of a graph could be arbitrary smaller than its pathwidth. On the contrary, we show that both parameters differ by at most one.

A *path-decomposition* of a graph $G = (V, E)$ is a sequence $P = (X_1, \ldots, X_p)$ of subsets of vertices, called *bags*, such that (1) $\bigcup_{i \leq p} X_i = V$; (2) for every $uv \in E$, there exists $i \leq p$ with $\{u, v\} \subseteq X_i$; and (3): for every $1 \leq i \leq j \leq q \leq p$, $X_i \cap X_q \subseteq X_j$. The *width* $w(P)$ of P is the size of the biggest bag of P, i.e., $w(P) = \max_{i \leq p} |X_i| - 1$. The *pathwidth* $pw(G)$ of G is the minimum width of its path-decompositions. A path-decomposition of G of width $pw(G)$ is said to be *optimal*. A path-decomposition is *reduced* if no bag is contained in another one. It is well known that any graph admits an optimal reduced path-decomposition.

Theorem 1. *For any graph $G = (V, E)$, $pw(G) \leq mh(G) \leq pw(G) + 1$.*

Proof. First, let $P = (X_1, \ldots, X_\ell)$ be a reduced path-decomposition of G with width k. Then, P is a monotone hunter strategy in G using $k+1$ hunters. This directly comes from the well known fact that, for every $1 \leq i < \ell$, $X_i \cap X_{i+1}$ separates $\bigcup_{1 \leq j \leq i} X_j \setminus X_i$ from $\bigcup_{i < j \leq \ell} X_j \setminus X_{i+1}$, and so $Z_i \subseteq \bigcup_{i < j \leq \ell} X_j$ for every $1 \leq i \leq \ell$. In particular, $mh(G) \leq pw(G) + 1$.

To show the other inequality, let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a parsimonious winning monotone hunter strategy in G using at most $k \geq mh(G)$ hunters (it exists by Lemma 9).

Claim 2. *For every $v \in V \setminus \bigcup_{1 \leq i \leq \ell} S_i$, there exists $1 \leq j \leq \ell$ such that $N(v) \subseteq S_j$.*

Proof of Claim. For the purpose of contradiction, let $v \notin \bigcup_{1 \leq i \leq \ell} S_i$ such that, for every $1 \leq i \leq \ell$, there exists $u_i \in N(v) \setminus S_i$. Then, $\mathcal{R} = (r_0 = v, u_1, v, u_3, \ldots, u_{2i-1}, v, u_{2i+1}, v \ldots)$ is a winning rabbit trajectory against $\mathcal{S}$, contradicting the fact that $\mathcal{S}$ is a winning hunter strategy. $\diamond$

Let us build a path-decomposition $\mathcal{P}$ of G as follows. Start with $\mathcal{P}_0 = \emptyset$ and let $Y_0 = V \setminus \bigcup_{1 \leq i \leq \ell} S_i$ (Y_0 is the set of vertices that are never shot by $\mathcal{S}$). Assume, by induction, that the sequence $\mathcal{P}_i$ and the set Y_i have been built for some $0 \leq i < \ell$. Let us define $\mathcal{P}_{i+1}$ and Y_{i+1} as follows. Let $H_{i+1} = \{u_1^{i+1}, \ldots, u_{r_{i+1}}^{i+1}\} = \{v \in Y_i \mid N(v) \subseteq S_{i+1}\}$. Let $\odot$ denote the concatenation of two sequences. Let $\mathcal{P}_{i+1} = \mathcal{P}_i \odot (S_{i+1} \cup \{u_1^{i+1}\}, \ldots, S_{i+1} \cup \{u_{r_{i+1}}^{i+1}\})$ and let $Y_{i+1} = Y_i \setminus H_{i+1}$. Finally, let $\mathcal{P} = (X_1, \ldots, X_r) = \mathcal{P}_\ell$.

Note that, by construction, for every $1 \leq i \leq r$, $|X_i| \leq k+1$, and so $w(\mathcal{P}) \leq k$. Let us show that $\mathcal{P}$ satisfies the three properties of a path-decomposition.

By construction, $\bigcup_{1 \leq i \leq \ell} S_i \subseteq \bigcup_{1 \leq i \leq r} X_i$. Moreover, by Claim 2, $Y_0 \subseteq \bigcup_{1 \leq i \leq r} X_i$. Hence, $\bigcup_{1 \leq i \leq r} X_i = V$ and the Property (1) of path-decomposition is satisfied.

By construction, for every $v \in Y_0$, there exists a unique $1 \leq i \leq r$ such that $v \in X_i$. Now, for any $v \in V \setminus Y_0$, let $1 \leq i \leq j \leq r$ such that $v \in X_i \cap X_j$. Let $1 \leq i' \leq \ell$ (resp., $i' \leq j' \leq \ell$) such that X_i has been built from $S_{i'}$ (resp., X_j has been built from $S_{j'}$). By Lemma 10, $v \in S_{p'}$ for all $i' \leq p' \leq j'$. Hence, by construction, $v \in X_p$ for all $i \leq p \leq j$. Therefore, Property (3) of the path-decomposition is satisfied for every $v \in V$.

Let $uv \in E$. First, let us assume that $u \in Y_0$. By Claim 2, $v \notin Y_0$. Let $1 \leq j \leq r$ such that $u \in X_j$, then by construction, $N(u) \subset X_j$ and so $u, v \in X_j$. Second, assume that $u, v \in V \setminus Y_0$. For purpose of contradiction, let us assume that, for every $1 \leq i \leq r$, $|\{u, v\} \cap S_i| \leq 1$. W.l.o.g., $M = \max\{1 \leq j \leq r \mid u \in X_j\} < m = \min\{1 \leq j \leq r \mid v \in X_j\}$ (both integers m and M are well defined since Property (1) is satisfied). Let $1 \leq M' \leq \ell$ (resp., $M' \leq m' \leq \ell$) such that X_M has been built from $S_{M'}$ (resp., X_m has been built from $S_{m'}$). By definition of $\mathcal{P}$, $u \notin \bigcup_{M' < i \leq \ell} S_i$ and $v \in S_{m'} \setminus \bigcup_{1 < i < m'} S_i$. Because $\mathcal{S}$ is parsimonious, $v \in Z_{m'-1}(\mathcal{S})$ and so, there exists a $w \in N(v) \cap Z_{m'-2}(\mathcal{S}) \setminus S_{m'-1}$. By monotonicity of $\mathcal{S}$, $w \notin \bigcup_{1 \leq i < m'} S_i$. By Proposition 2, $u \in Z_{m'-1}(\mathcal{S})$. Since $u \in (Z_{m'-1}(\mathcal{S}) \cap S_{M'}) \setminus S_{m'}$, the vertex u is recontaminated, contradicting the monotonicity of $\mathcal{S}$. Therefore, in all cases, for every $uv \in E$, there exists $1 \leq j \leq r$ such that $u, v \in X_j$. Hence, Property (2) of path-decompositions is satisfied.

Hence, $\mathcal{P}$ is a path-decomposition of width at most k. In particular, $pw(G) \leq mh(G)$. $\square$

Theorem 1 has important consequences.

Corollary 1. *Unless $\mathsf{P} = \mathsf{NP}$, there is no polynomial time algorithm that, given a graph G and $k \in \mathbb{N}$, decides whether $mh(G) \leq k$.*

Proof. This comes from Theorem 1 and the fact that the pathwidth of a graph cannot be approximated up to an additive constant [5] (unless $\mathsf{P} = \mathsf{NP}$). $\square$

Moreover, Theorem 1 implies that recontamination may help in the Hunters and Rabbit game.

Corollary 2. *There exists $\varepsilon > 0$ such that, for any $k \in \mathbb{N}$, there exists a tree T with $h(T) \geq k$ and $mh(T) \geq (1+\varepsilon)h(T)$.*

Proof. For any $n \in \mathbb{N}$, let T_n be the rooted tree defined as follows: T_0 is a single node, and, for any $n > 0$, T_n is obtained from three copies of T_{n-1} and a new node r (the root of T_n) such that r is made adjacent to each of the three roots of the copies of T_{n-1}. We have that $|V(T_n)| = \frac{3^{n+1}-1}{2}$ and, by the Parsons' Lemma [25], $pw(T_n) = n = \Theta(\log_3(V(T)))$. On the other hand, it is shown in [15] that, for any tree T, $h(T) \leq \lceil \frac{\log_2 |V(T)|}{2} \rceil$. The result follows for $(1+\varepsilon) = 2\frac{\ln(2)}{\ln(3)}$. □

4 (Monotone) hunter number of some graph classes

In this section, we characterize the monotone hunter number of several graph classes such as split graphs, interval graphs, cographs and trees. In particular, in all these cases, our results lead to a polynomial time algorithm to compute the monotone hunter number.

4.1 Split and interval graphs

A graph $G = (V, E)$ is a *split graph* if $V = C \cup I$ can be partitioned into a set C inducing an inclusion-maximal clique and a set I inducing an independent set. Note that given a split graph G, a partition (C, I) of $V(G)$ can be computed in linear time [17]. In what follows, we denote a split graph by $G = (C \cup I, E)$ where C induces an inclusion-maximal clique and I induces an independent set. Let us recall the following result on the pathwidth of split graphs:

Lemma 11. *[16] Let $G = (C \cup I, E)$ be a split graph. Then, $|C| - 1 \leq pw(G) \leq |C|$.*

First, we have the following easy observation.

Proposition 3. *Let $G = (C \cup I, E)$ be a split graph. Then, $|C| - 1 \leq h(G) \leq mh(G) \leq |C|$.*

Proof. By Lemma 1, $h(G) \geq h(G[C])$, and by Lemma 4, $h(G[C]) \geq \delta(G[C]) = |C| - 1$. Therefore, $h(G) \geq |C| - 1$. Moreover, the hunter strategy that consists in shooting to all the vertices of C twice is clearly a monotone winning hunter strategy in G. Hence, $h(G) \leq mh(G) \leq |C|$. □

The following theorem fully characterizes the hunter number of split graphs.

Theorem 2. *Let $G = (C \cup I, E)$ be a split graph. Then, $h(G) = |C|$ if and only if for every two distinct vertices $x, y \in C$, there exists a vertex $z \in N_I(x) \cap N_I(y)$. Otherwise, $h(G) = |C| - 1$.*

Proof. First we show that if for every two distinct vertices $x, y \in C$, there exists a vertex $z \in I$ such that $xz \in E$ and $yz \in E$, then $h(G) = |C|$. We prove this by showing that there exists a winning rabbit strategy against $|C| - 1$ hunters. That is, for any (fixed) hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ such that $|S_i| \leq |C| - 1$ for every $i \geq 1$, we design a rabbit trajectory $\mathcal{R} = (r_0, r_1, \ldots, r_{\ell-1})$ such that for every $i \geq 0$, $r_i \notin S_{i+1}$. Since $|S_1| \leq |C| - 1$, there is at least one vertex, say, $v \in C$, such that $v \notin S_1$. Let $r_0 = v$. Hence the rabbit is safe for the first round (since $r_0 \notin S_1$). Now, for $i \geq 0$, let us assume that we have built $(r_0, \ldots, r_i)$ such that $r_j \notin S_{j+1}$ for every $0 \leq j \leq i$ and $r_i \in C$. If there is at least one vertex $u \neq r_i$ in C such that u is not shot in round $i + 2$ (i.e., $u \notin S_{i+2}$), then let $r_{i+1} = u$. Otherwise, $S_{i+2} = C \setminus \{r_i\}$. Moreover, observe that there is at least one vertex $w \in C$ such that $w \notin S_{i+3}$ (since $|S_{i+3}| < |C|$). We note here that w may be the same vertex as r_i. Due to our assumptions, there exists a vertex $z \in I$ such that $wz, r_i z \in E$. Let us set $r_{i+1} = z$ and $r_{i+2} = w$. Observe that $r_{i+1} \notin S_{i+2}$ (since $S_{i+2} = C \setminus \{r_i\}$ and $z \in I$), $r_{i+2} \notin S_{i+3}$, and $r_{i+2} \in C$. Therefore, using the above strategy,

we can design $\mathcal{R}$ such that it is a winning trajectory against $\mathcal{S}$. Therefore, $h(G) \geq |C|$. Since $h(G) \leq |C|$ (due to Proposition 3), we have that $h(G) = |C|$.

To prove the reverse direction, we show that if there exist two distinct vertices $x, y \in C$ such that $N_I(x) \cap N_I(y) = \emptyset$ (i.e., there is no $z \in I$ such that $xz \in E$ and $yz \in E$), then $h(G) \leq |C|-1$ (and so $h(G) = |C| - 1$ by Proposition 3). We prove this by giving a (simple) winning hunter strategy $\mathcal{S}$ using $|C| - 1$ hunters. Let $\mathcal{S} = (S_1, S_2, S_3, S_4, S_5)$ where $S_1 = S_2 = S_5 = C \setminus \{y\}$ and $S_3 = S_4 = C \setminus \{x\}$. Let $\mathcal{R} = (r_0, \ldots, r_4)$ be any rabbit trajectory. If the rabbit is not shot at the first round, i.e., $r_0 \notin S_1$, then either $r_0 = y$ or $r_0 \in I$. Accordingly, we consider both these cases below to show that $\mathcal{S}$ is a winning hunter strategy.

Case 1. $\mathbf{r_0 = y}$: In this case, assume $r_1 \notin S_2$ (otherwise, the rabbit will be shot). Note that $r_1 \in N_I(y)$. Since $N_I(x) \cap N_I(y) = \emptyset$, $r_2 \in C \setminus \{x\}$. As $S_3 = C \setminus \{x\}$, $r_2 \in S_3$, and therefore, $\mathcal{S}$ is a winning hunter strategy.

Case 2. $\mathbf{r_0 \in I}$: In this case, if $r_1 \notin S_2$, then $r_1 = y$. Now, assuming that $r_2 \notin S_3$, the rabbit can either move to x (i.e, $r_2 = x$) or the rabbit can move to $N_I(y)$ (i.e., $r_2 \in N_I(y)$). We have the following two cases accordingly:

2.a $\mathbf{r_2 \in N_I(y)}$: This case is similar to Case 1. Since $N_I(x) \cap N_I(y) = \emptyset$, $r_3 \in C \setminus \{x\}$. As $S_4 = C \setminus \{x\}$, $r_3 \in S_4$, and therefore, $\mathcal{S}$ is a winning hunter strategy.

2.b $\mathbf{r_2 = x}$: In this case, if $r_3 \notin S_4$, then $r_3 \in N_I(x)$. Therefore, similarly to previous arguments, $r_4 \in C \setminus \{y\}$. Since $S_5 = C \setminus \{y\}$, $\mathcal{S}$ is a winning hunter strategy.

This completes the proof. □

The above characterization allows us to show that the hunter number and the pathwidth of split graphs coincide.

Corollary 3. *For any split graph $G = (C \cup I, E)$, $h(G) = pw(G)$.*

Proof. If $|C| = 1$ and $I = \emptyset$, then $pw(G) = h(G) = 0$. If $|C| = 1$ and $I \neq \emptyset$, then $pw(G) = h(G) = 1$. Let us now assume that $|C| > 1$.

Since $pw(G), h(G) \in \{|C| - 1, |C|\}$ by Lemma 11 and Proposition 3, let us assume first that $h(G) = |C|$. By Theorem 2, for any two distinct vertices $x, y \in C$, there exist $z \in N_I(x) \cap N_I(y)$. For purpose of contradiction, let us assume that there exists a reduced optimal path-decomposition $P = (X_1, \ldots, X_\ell)$ of width $|C| - 1$. It is well known that there exists $1 \leq i \leq \ell$ with $C \subseteq X_i$. Moreover, since P has width $|C| - 1$, $X_i = C$. Let us prove now that $1 < i < \ell$. Let us suppose by contradiction that $i = 1$, i.e., $X_1 = C$ (the case $i = \ell$ is symmetric). Since P has width $|C| - 1$ and is reduced, let $v \in X_1 \setminus X_2$ and let $z \in N_I(v)$ (z exists since $|C| > 1$ and any two distinct vertices of C have a common neighbour in I). Since $v \in X_1 = C$ and $v \notin \bigcup_{1<j\leq\ell} X_j$, no bag contains both v an z, contradicting the definition of a path-decomposition. Hence, $1 < i < \ell$. Now, let $x \in C \setminus X_{i-1}$ and $y \in C \setminus X_{i+1}$ (x and y exists since P is reduced and $X_i = C$). Let $z \in N_I(x) \cap N_I(y)$. If $x = y$, no bag contains both x and z (since x appears only in $X_i = C$). If $x \neq y$, there must exist $1 \leq j \leq \ell$ such that $\{y, z\} \subseteq X_j$. Since $y \notin \bigcup_{i<h\leq\ell} X_h$, $j < i$ and since $z \in X_j \setminus X_i$, $z \notin \bigcup_{i<h\leq\ell} X_h$. Finally, since $x \notin \bigcup_{1\leq h<i} X_h$, there is no bag containing both x and z, contradicting the definition of a path-decomposition.

Second, let us assume that $h(G) = |C| - 1$. By Theorem 2, there exist distinct vertices $x, y \in C$ such that $N_I(x) \cap N_I(y) = \emptyset$. Let us prove that, in that case, $pw(G) = |C| - 1$. Let $N_I(x) = \{x_1, \ldots, x_m\}$ and $I \setminus N_I(x) = \{y_1, \ldots, y_t\}$. Then, $(x_1 \cup (C \setminus \{y\}), \ldots, x_m \cup (C \setminus \{y\}), C, y_1 \cup (C \setminus \{x\}), \ldots, y_t \cup (C \setminus \{x\}))$ is a path decomposition of G with width $|C| - 1$ and, since $pw(G) \geq |C| - 1$ by Lemma 11, we get that $pw(G) = |C| - 1$. □

Next, let us characterize the monotone hunter number of split graphs. We start with the following general lemma.

Lemma 12. *Let G be a graph that contains a complete subgraph C such that $N(v) \setminus C \neq \emptyset$ for every $v \in C$. Then, $mh(G) \geq |C|$.*

Proof. By Lemmas 1, 4 and Proposition 1, $mh(G) \geq h(G) \geq h(C) \geq |C|-1$. Let $H = G[N[C]]$. We will show that $mh(H) \geq |C|$ and so, the result will follow from Lemma 8.

Let us assume by contradiction that $mh(H) = |C| - 1$. By Lemma 9, there exists a parsimonious monotone winning hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ in H using $|C| - 1$ hunters.

There must be an index $1 \leq i \leq \ell$ such that $|S_i \cap C| = |C| - 1$. Otherwise, $(r_0, \ldots, r_\ell)$ where $r_0 \in C \setminus S_1$ and $r_j \in C \setminus (S_{j+1} \cup \{r_{j-1}\})$ for every $1 \leq j \leq \ell$ is a winning rabbit trajectory, contradicting the fact that $\mathcal{S}$ is winning. Hence, let i be the smallest integer such that $|S_i \cap C| = |C| - 1$, let $\{v\} = C \setminus S_i$ and let $w \in N(v) \setminus C$ (which exists by hypothesis). Note that $v \notin \bigcup_{1 \leq j \leq i} S_j$ since otherwise, v would have been recontaminated.

If $w \in \bigcup_{1 \leq j < i} S_j$, let $(r_0, \ldots, r_{i-1} = v)$ be a rabbit trajectory such that $r_j \notin S_{j+1}$ for any $1 \leq j < i$ (with $r_{i-1} = v$ and $r_j \in C \setminus (S_{j+1} \cup \{r_{j+1}\})$ for every $1 \leq j < i-1$, which is possible since $|S_j \cap C| < |C| - 1$ for all $j < i$). If $w \notin S_{i+1}$, then let $r_i = w$: this contradicts the monotonicity of $\mathcal{S}$ (since $w \in Z_{i+1} \cap \bigcup_{1 \leq j < i} S_j) \setminus S_{i+1}$). Hence, $w \in S_{i+1}$ and so there exists $z \in C \setminus (S_{i+1} \cup \{v\})$. In this latter case, let $r_i = z$, contradicting the monotonicity of $\mathcal{S}$ (since $z \in Z_{i+1} \cap S_i) \setminus S_{i+1}$).

Otherwise (if $w \notin \bigcup_{1 \leq j < i} S_j$), let $j > i$ be the smallest integer such that $w \in S_j$, or, if w is never shot, let $j > i$ be the smallest integer such that $v \in S_j$ (it must exists otherwise the rabbit may oscillate between v and w without being never shot). In both cases, let $z \in C \setminus (S_j \cup \{v\})$. Thus, by Proposition 2, $z \in Z_{j-1} \setminus S_j$, contradicting the monotonicity of $\mathcal{S}$.

□

Recall that a vertex in a graph G is *simplicial* if its neighbourhood induces a clique. In particular, in a split graph $G = (C \cup I, E)$, a vertex $v \in C$ is simplicial if and only if $N(v) \setminus C = \emptyset$ (recall that C is supposed to be an inclusion-maximal clique).

Theorem 3. *Let $G = (C \cup I, E)$ be a split graph. Then, $mh(G) = |C| - 1$ if and only if there exists a simplicial vertex in C. Otherwise, $mh(G) = |C|$.*

Proof. Note first that if there is no simplicial vertex in C, then by Lemma 12, $mh(G) \geq |C|$ and so, by Proposition 3, $mh(G) = |C|$. Otherwise, if there exists a simplicial vertex $v \in C$, then $S = (C \setminus v, C \setminus v)$ is a monotone winning hunter strategy in G. □

Note that above results imply that there exist split graphs G for which $mh(G) \neq h(G)$, i.e., recontamination helps in split graphs.

To conclude this section, let us show another application of Lemma 12. Recall that an *interval graph* is the intersection graph of a set of intervals in the real line. It is well known that, for any interval graph G, $pw(G) = \omega(G) - 1$ where $\omega(G)$ is the maximum size of a clique in G, and that G admits an optimal path-decomposition where each bag induces a complete graph.

Theorem 4. *Let G be an interval graph. Then, $h(G) = mh(G) = \omega(G) - 1$ if every maximum clique has a simplicial vertex. Otherwise, $mh(G) = \omega(G)$.*

Proof. By Theorem 1, $pw(G) \leq mh(G) \leq pw(G) + 1 = \omega(G)$. Moreover, by Lemma 4, $h(G) \geq \omega(G) - 1$. If there exists a clique of maximum size that does not contain any simplicial vertex, then by Lemma 12, $mh(G) = \omega(G)$. Otherwise, let $(X_1, \ldots, X_\ell)$ be an optimal path-decomposition of G such that all bags induce a complete graph. For every $1 \leq$

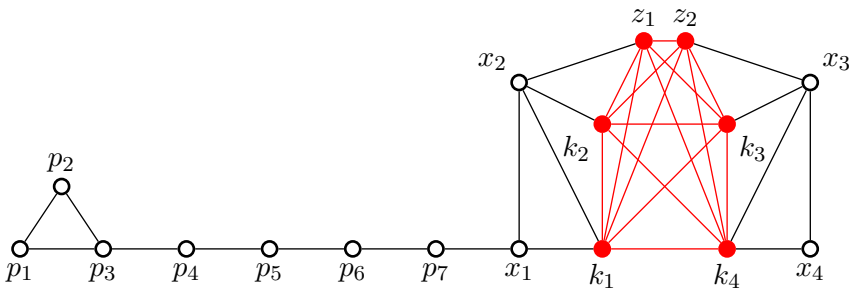


Figure 2: An interval graph G containing the complete graph K_6 (represented by the red vertices and edges). Observe that the maximum clique of G does not contain a single simplicial vertex.

$i \leq \ell$, if X_i contains a simplicial vertex v_i, let $Y_i = \{v_i\}$ and let $Y_i = \emptyset$ otherwise. Then, $(X_1 \setminus Y_1, X_1 \setminus Y_1, X_2 \setminus Y_2, X_2 \setminus Y_2, X_3 \setminus Y_3, \ldots, X_\ell \setminus Y_\ell, X_\ell \setminus Y_\ell)$ is a monotone winning hunter strategy using $\omega(G) - 1$ hunters. □

It follows that $\omega(G) - 1 \leq h(G) \leq \omega(G)$. But, the question of deciding $h(G)$ in interval graph when some maximum clique has no simplicial vertex seems much more challenging.

To illustrate this, consider the interval graph shown in Figure 2. Let $K = \{k_1, k_2, k_3, k_4\}$ and $\mathcal{S} = (K \cup \{x_4\}, K \cup \{x_3\}, K \cup \{z_2\}, K \cup \{z_1\}, K \cup \{x_2\}, K \cup \{x_1\}, \{x_1, p_7\}, \{p_7, p_6\}, \{p_6, p_5\}, \{p_5, p_4\}, \{p_4, p_3\}, \{p_3, p_2, p_1\})$. Note $\mathcal{S}$ is a winning hunter strategy if $Z_0 \subseteq \{p_1, p_2, p_3, p_5, p_7, x_2, z_2, x_4\} \cup K$. It follows that applying $\mathcal{S}$ twice leads is a winning hunter strategy in respect to $Z_0 \subseteq V$ and, thus, $h(G) \leq |K| + 1 = \omega(G) - 1$. However, $K \cup \{z_1, z_2\}$ induces a maximal clique without any single simplicial vertex. Let G' be the graph obtained from G by contracting the path $(p_3, p_4, p_5, p_6, p_7, x_1)$ into a single edge. Unfortunately, we were not able to find a winning hunter strategy in G' using $\omega(G') - 1$ hunters. Thus, we leave as an open question whether the determination of the hunter number of interval graph defined similarly to the graph G'.

4.2 Cographs

The class of *cographs* can be defined recursively as follows [10]. One vertex is a cograph. Given two cographs A and B, their disjoint union $A \cup B$ is a cograph, and their join $A \bowtie B$ (where all edges between A and B are added) is a cograph. Note that a decomposition of a cograph (*i.e.*, a building sequence of unions and joins performed from single vertices) can be computed in linear time [10].

Theorem 5. *$mh(G)$ can be computed in linear time in the class of cographs.*

Proof. Let A and B be two cographs. We prove that:

- $mh(A \cup B) = max(mh(A), mh(B))$, and
- $mh(A \bowtie B) = min(mh(A) + |V(B)|, |V(A)| + mh(B))$.

The result then follows from the linear time algorithm to compute the recursive decomposition of cographs [10].

The first statement is obvious, so let us prove the second one. Let $G = A \bowtie B$ and let $\mathcal{S}^A = (S_1^A, \ldots, S_\ell^A)$ and $\mathcal{S}^B$ be two monotone winning hunter strategies for A and B and using respectively $mh(A)$ and $mh(B)$ hunters. Note that $\mathcal{S}^A \cup V(B) = (S_1^A \cup V(B), \ldots, S_\ell^A \cup V(B))$ and $\mathcal{S}^B \cup V(A)$ are both monotone winning hunter strategies in G. Therefore, $mh(G) \leq min(mh(A) + |V(B)|, |V(A)| + mh(B))$.

Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a parsimonious monotone winning hunter strategy in G using at most $k \geq mh(G)$ hunters (it exists by Lemma 9) and such that ℓ is minimized among all such strategies. If $\ell = 1$, then $k \geq |S_1| \geq |V(G)| \geq min(mh(A) + |V(B)|, |V(A)| + mh(B))$. Hence, let us assume that $\ell > 1$. Let $\mathcal{Z} = (Z_0, \ldots Z_\ell)$ be the set of contaminated vertices with respect to $\mathcal{S}$. Note first that since ℓ is minimum, $(S_2, \ldots, S_\ell)$ is not a winning hunter strategy, and so $Z_1 \neq V$. Let $v \in V \setminus Z_1$.

Let us assume that $v \in V(A)$ (the case $v \in V(B)$ is symmetric). Since $v \notin Z_1$ and $Z_0 = V$, $N(v) \subseteq S_1$. Since $B \subseteq N(v)$, we have that $V(B) \subseteq S_1$. Moreover, $V(B) \subseteq Z_1$. Otherwise, there exists $w \in B$ such that $N(w) \subseteq S_1$ and since $V(A) \subseteq N(w)$, we would have $S_1 = V(A) \cup V(B) = V(G)$, a contradiction to the fact that $\ell > 1$.

Let us prove by induction on i that, for every $1 \leq i < \ell$, $V(B) \subseteq Z_i$ and $V(B) \subseteq S_i$. This statement holds for $i = 1$ by the previous paragraph. By induction, let us assume that $V(B) \subseteq Z_i$ and $V(B) \subseteq S_i$ for some $1 \leq i < \ell - 1$. Since $\mathcal{S}$ is monotone, $V(B) \subseteq S_{i+1}$. Let us assume that there exists $b \in V(B)$ such that $b \notin Z_{i+1}$. It implies that $V(A) \cap Z_i \subseteq S_{i+1}$. Therefore, $Z_i = (V(A) \cap Z_i) \cup (V(B) \cap Z_i) \subseteq S_{i+1}$, which implies that $Z_{i+1} = \emptyset$, contradicting the minimality of ℓ. Thus $V(B) \subseteq Z_{i+1}$ and the induction hypothesis holds for $i + 1$. In particular, $V(B) \subseteq Z_{\ell-1}$ and so, by monotonicity, $V(B) \subseteq S_\ell$. Therefore, $V(B) \subseteq S_i$ for all $1 \leq i \leq \ell$. Since $V(B) \subseteq S_i$ for all $1 \leq i \leq \ell$, the strategy $\mathcal{S} \cap V(A) = (S_1 \cap V(A), \ldots, S_\ell \cap V(A))$ is a monotone winning hunter strategy in $G[A]$ using $k - |V(B)|$ hunters. Hence, $k - |V(B)| \geq mh(A)$ which concludes the proof. □

Once again, the case of the hunter number seems more challenging. In particular, the following lemma shows that recontamination may help in cographs.

Lemma 13. *For every $k \geq 1$, there exists a cograph G such that $h(G) \geq k$ and $mh(G) \geq \frac{3}{2}h(G) - 1$.*

Proof. Let $a \geq 3$. Let A and B be two (isomorphic) cographs that consist of the disjoint union of a complete graph with a vertices (denoted by K_A and K_B respectively) and a independent vertices (so $|V(A)| = |V(B)| = 2a$). Let $G = A \bowtie B$. Clearly, $h(A) = mh(A) = h(B) = mh(B) = a - 1$ and, by the proof of Theorem 5, $mh(G) = 3a - 1$. Note also that $h(G) \geq a - 1$ by Lemma 4. Now, $(A, K_A \cup K_B, K_B, A)$ is a (non-monotone) winning hunter strategy in G using $2a$ hunters and so $h(G) \leq 2a$. □

4.3 Trees

This section is devoted to showing that the monotone hunter number of trees can be computed in polynomial time. Roughly, we show that a Parsons' like lemma [25] holds for the monotone hunter number in trees and then the algorithm follows the one for computing the pathwidth of trees in [12].

Let us start with the easy case of paths.

Proposition 4. *Let P be any path with at least 4 vertices. Then, $1 = h(P) < mh(P) = 2$.*

Proof. The fact that $h(P) = 1$ has been proven in [1], and the fact that $mh(P) \leq 2$ is easy.

Towards a contradiction, let us assume that there exists a winning monotone hunter strategy in P using one hunter and let $\mathcal{S} = (S_1, \ldots S_\ell)$ be a shortest such strategy (i.e., minimizing ℓ). Let $Z = (Z_1, \ldots, Z_\ell)$ be the set of contaminated vertices with respect to $\mathcal{S}$. Let $w \in V(P)$ such that $S_1 = \{w\}$. Note that $w \in Z_1$ and so, $\ell > 1$. Since P has length at least 4, there exist $x, y \in V(P)$ such that $x \in N(w)$ and $y \in N(x) \setminus \{w\}$. We will prove by induction on i that $S_i = \{w\}$ for all $1 \leq i \leq \ell$. The base case ($i = 1$) is already proven. Assume now that

for some $1 \leq q < \ell$, it holds that $S_j = \{w\}$ for all $1 \leq j \leq q$. Thus, $x, y \notin \bigcup_{1 \leq j \leq q} S_j$ and so, by Proposition 2, $w \in Z_q$. Hence, by the monotonicity of $\mathcal{S}$, we have that $w \in S_{q+1}$, showing the step of the induction. Therefore, $x, y \notin \bigcup_{1 \leq j \leq \ell} S_j$ and so, by Proposition 2, $w, x, y \in Z_\ell$, contradicting the fact that $\mathcal{S}$ is a winning strategy in P. Therefore, $mh(P) \geq 2$. □

We then need the following two technical results.

Proposition 5. *Let $G = (V, E)$ be any connected graph and H be a connected subgraph of G. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be any parsimonious monotone winning hunter strategy in G. Moreover, let $1 \leq i \leq \ell$ and $x, y \in V(H)$ such that $x \in \bigcup_{j<i} S_j$, $y \in Z_{i-1}$ and minimising the distance between such x and y in H. If $x, y \notin S_i$, then $xy \in E(H)$.*

Proof. Note first that $x \neq y$. Indeed, assuming otherwise would imply that $\mathcal{S}$ is not monotone since $y = x \in (\bigcup_{j<i} S_j \cap Z_{i-1}) \setminus S_i$. Hence, we may assume that $x \neq y$. Let P be a shortest path from x to y in H. Let a be the neighbour of x in P. If $a = y$, then $\{x, y\} \in E(G)$, and the claim holds. Hence, we may assume that $a \neq y$. By minimality of the distance between x and y, $a \notin Z_{i-1}$ and $a \notin \bigcup_{j<i} S_j$. Let $b \neq x$ be the other neighbour of a in P. We show that $b \notin \bigcup_{j<i} S_j$. If $b \neq y$, then, by minimality of the distance between x and y, $b \notin \bigcup_{j<i} S_j$. If $b = y$, since $y \in Z_{i-1} \setminus S_i$, then $y \notin \bigcup_{j<i} S_j$, because otherwise this would contradict the monotonicity of $\mathcal{S}$. Therefore, by Proposition 2, $a \in Z_q$ for every $q < i$, a contradiction to the minimality of the distance between x and y. □

Lemma 14. *Let $G = (V, E)$ be any graph and $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a parsimonious monotone winning hunter strategy in G that uses at most k hunters. Let H be a connected subgraph of G with $|V(H)| > 1$. If $S_i \cap V(H) \neq \emptyset$ and $S_j \cap V(H) \neq \emptyset$ for some $1 \leq i < j \leq \ell$, then $S_z \cap V(H) \neq \emptyset$ for every $i \leq z \leq j$.*

Proof. Let $2 \leq i + 1 < j \leq \ell$ be such that $V(H) \cap S_i \neq \emptyset$ and $V(H) \cap S_j \neq \emptyset$. Towards a contradiction, let us assume that there exists $i < z < j$ such that $S_z \cap V(H) = \emptyset$. Let $X = V(H) \cap \bigcup_{q<z} S_q$. Since $V(H) \cap S_i \neq \emptyset$ and $i < z$, we get that $X \neq \emptyset$. Let $Y = V(H) \cap Z_{z-1}$. Since $S_j \cap V(H) \neq \emptyset$ and $\mathcal{S}$ is parsimonious, we have that $Z_{j-1} \cap V(H) \neq \emptyset$. Let $u \in Z_{j-1} \cap S_j \cap V(H)$. By Lemma 6, $u \in Z_q$ for every $q < j$. In particular, $u \in Z_{z-1}$ and so $Y \neq \emptyset$. Let $x \in X$ and $y \in Y$ such that the distance bewteen x and y in H is minimum. By Proposition 5, $xy \in E$. Thus, since $y \in Z_{z-1} \setminus S_z$, we get that $x \in Z_z$. Therefore, since $\mathcal{S}$ is monotone and $x \in \bigcup_{q<z} S_q$, we must have $x \in S_{z+1}$. By Lemma 10, since $x \in \bigcup_{q<z} S_q$ and $x \in S_{z+1}$, then $x \in S_z$. Hence, $V(H) \cap S_z \neq \emptyset$, a contradiction. □

Let T be a tree and $v \in V(T)$. A *branch* at v is any connected component of $T - v$. A *star* is any tree with at least two vertices and at most one vertex with degree at least three. Roughly speaking, Parsons' Lemma [25] states that, for any tree T and $k \in \mathbb{N}$, $pw(T) \geq k + 1$ if and only if there exists a vertex v such that at least three branches at v have pathwidth at least k. Here, we adapt this lemma in the case of the monotone hunter number of trees.

Lemma 15. *[Parsons' like lemma] Let $T = (V, E)$ be any tree.*

- *$mh(T) = 0$ if and only if $|V| = 1$;*
- *$mh(T) = 1$ if and only if T is a star;*
- *$mh(T) = 2$ if and only if T is not a star and contains a path P such that $T \setminus P$ is a forest of stars and isolated vertices;*
- *For every $k \geq 3$, $mh(T) \geq k$ if and only if there exists a vertex $v \in V$ such that at least three branches at v have monotone hunter number at least $k - 1$.*

Proof. The first item is trivial. Then, if T is a star, then shooting twice a vertex of maximum degree is a monotone winning hunter strategy using one hunter. If T is not a star, it contains a path with at least 4 vertices as a subgraph. By Proposition 4 and Lemma 8, it follows that $mh(T) \geq 2$, which concludes the proof of the second item. If T is not reduced to a star and contains a path P such that $T \setminus P$ is a forest of stars and isolated vertices, it is easy to show that $mh(T) \leq 2$. Otherwise, T contains a vertex v such that at least three components of $T - v$ contain a path with 4 vertices. The "if" statement of the fourth item then shows that $mh(T) > 2$ and concludes the proof of the third item.

Let us prove the fourth item. Let $k \geq 3$.

Proof of $\Leftarrow$: Let us first assume that there exists some vertex v and three branches B_1, B_2 and B_3 at v, such that $mh(B_1), mh(B_2), mh(B_3) \geq k-1$. We will show that $mh(T) \geq k$. Towards a contradiction, let us also assume that $mh(T) < k$. By Lemma 9, there exists a parsimonious monotone strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ in T that uses at most $k-1$ hunters. Let $Z = (Z_1, \ldots, Z_\ell)$ be the set of contaminated vertices with respect to $\mathcal{S}$.

For $j \in \{1, 2, 3\}$, let $1 \leq i_j \leq \ell$ be the minimum integer such that $V(B_j) \cap Z_{i_j} = \emptyset$. Note that, by Lemma 6, $V(B_j) \cap Z_q = \emptyset$ for all $i_j \leq q \leq \ell$, and since $\mathcal{S}$ is parsimonious, $V(B_j) \cap S_q = \emptyset$ for all $i_j < q \leq \ell$. Note also that, since $mh(B_j) \geq k-1 \geq 2$, B_j has at least two vertices. Since $Z_{i_j-1} \cap V(B_j) \neq \emptyset$ and $Z_{i_j} \cap V(B_j) = \emptyset$, it implies that $S_{i_j} \cap V(B_j) \neq \emptyset$ (otherwise, let $w \in (Z_{i_j-1} \cap V(B_j)) \setminus S_{i_j}$ and $u \in N(w) \cap V(B_j)$, then $u \in Z_{i_j} \cap V(B_j)$, a contradiction with the definition of i_j). W.l.o.g., let us assume that $i_1 \leq i_2 \leq i_3$.

We will show that there exists a round j_2 during which all the $k-1$ hunters will have to shoot on vertices of B_2, and that $v \in Z_{j_2-1}$, which will lead to a contradiction.

For any $1 \leq i \leq 3$, let j_i be an index such that $|S_{j_i} \cap V(B_i)| = k-1$. These indices exist as, otherwise, by Lemma 8, $mh(B_i) < k-1$. We will first show that $j_2 < i_3$, which will be used to prove that $v \in Z_{j_2-1}$. Observe that $j_2 \leq i_2 \leq i_3$. Moreover, $j_2 \neq i_3$ since $S_{i_3} \cap V(B_3) \neq \emptyset$ and $\mathcal{S}$ uses at most $k-1$ hunters. Thus, $j_2 < i_3$. Therefore, by Proposition 2 and Lemma 14, $V(B_3) \cup \{v\} \subseteq Z_q$ for all $q \leq j_2$. In particular, $v \in Z_{j_2-1}$. Moreover, since $S_{j_2} \subseteq V(B_2)$, $v \notin S_{j_2}$. Hence, $x \in Z_{j_2}$ where x is the neighbour of v in B_1, i.e., $Z_{j_2} \cap V(B_1) \neq \emptyset$.

Since $Z_{i_1} \cap V(B_1) = \emptyset$, if $i_1 < j_2$, then there is a contradiction to Lemma 6. Otherwise, $j_2 < i_1 \leq i_2$ (because $j_2 \neq i_1$) and so either $j_1 < j_2 < i_1$ or $j_2 < j_1 < i_2$ (because $j_1 \neq j_2$, $j_1 \leq i_1 \leq i_2$ and $j_1 \neq i_2$), both contradicting Lemma 14.

Proof of $\Rightarrow$: Now let us assume that, for every $v \in V$, at most two branches at v have monotone hunter number at least $k-1$. Let us prove that there exists a parsimonious monotone winning hunter strategy $\mathcal{S}$ in T using at most $k-1$ hunters.

First, let us assume that there exists a path $P = (v_1, \ldots, v_p)$ such that for any connected component C of $T \setminus P$, $mh(C) < k-1$. The following hunter strategy is parsimonious monotone winning and uses at most $k-1$ hunters. The strategy consists of p phases executed sequentially from $i = 1$ to p. Phase i consists in shooting v_i at each round, and in using the $k-2$ remaining shots to clear sequentially each connected component of $T \setminus P$ which is adjacent to v_i (this is possible since each of these branches at v have monotone hunter number at most $k-2$). Finally, the last round of Phase i (except for $i = p$) consists in shooting to both v_i and v_{i+1} (recall that $k - 1 \geq 2$).

Let us now show that a path P, defined as in the previous paragraph, exists. Let X be the set of vertices v such that exactly two branches at v have monotone hunter number at least $k-1$. First, let us assume that $X \neq \emptyset$ and let us show that it induces a path. Let $x, y \in X$ and let z be any internal vertex of the path between x and y. Let B (resp., B') be the branch at z that contains x (resp., that contains y). One branch B_x at x with $mh(B_x) \geq k-1$ is a subgraph of B and so, by Lemma 8, $mh(B) \geq k-1$. Similarly, $mh(B') \geq k-1$ and so, $z \in X$ and therefore X induces a subtree of T. If there exists a node w of degree at least three in $T[X]$,

by similar arguments, there are at least three branches at w with monotone hunter number at least $k-1$, a contradiction with the initial hypothesis. Hence, X induces a path $(v_2, \ldots, v_{p-1})$. Let B_1 be the branch at v_2 not containing v_3 such that $mh(B_1) = k-1$ (if v_3 does not exist, B_1 is any branch at v_2) and let v_1 be the neighbour of v_2 in B_1. Symmetrically, let B_p bet the branch at v_{p-1} not containing v_{p-2} such that $mh(B_p) = k-1$ (if v_{p-1} does not exist, B_p has to be distinct from B_1) and let v_p be the neighbour of v_{p-1} in B_p. Then, $P = (v_1, v_2, \ldots, v_{p-1}, v_p)$ satisfies the desired conditions.

Finally, if $X = \emptyset$, let v_1 be any vertex of T. We build the path P starting from v_1 as follows. Let us assume that a path $(v_1, \ldots, v_i)$ has already been built for some $i \geq 1$. If there exists a branch B at v_i, not containing v_{i-1} (if $i > 1$), and with monotone hunter number at least $k-1$ (if any, such a branch must be unique since $X = \emptyset$), then, let v_{i+1} be the neighbour of v_i in B. The process ends when no such branch B exists and the obtained path satisfies the desired conditions.

This completes the proof. □

We design a dynamic programming algorithm to compute the monotone hunter number of a tree T. Let us first root T at any vertex $r \in V(T)$. Let $T[u]$ denote the subtree of T induced by $u \in V(T)$ and all the descendent of u. Let $T[u, x, \ldots, y]$ denote the subtree obtained from $T[u]$ after the vertices from $V(T[x]) \cup \ldots \cup V(T[y])$ have been removed. Finally, for $k \geq 2$, a vertex $x \in V(T)$ is said *k-critical* if and only if $mh(T[x]) = k$ and there exist two children v_1 and v_2 of x in T such that $mh(T[v_1]) = mh(T[v_2]) = k$. In the case $k = 1$, we will say that a vertex x is 1-critical if and only if $mh(T[x]) = 1$ and there exists a child v such that $mh(T[v]) = 1$. Let us recall that $mh(T[w]) \geq 1$ if $T[w]$ contains at least one edge, i.e. w has at least one child. Therefore, if a vertex w is 1-critical in $T[w]$, then $T[w]$ is a star rooted in the child w' of w such that $mh(T[w']) = 1$, i.e. w' is the only vertex of $T[w]$ that has degree at least 2. Also if a vertex w is not 1-critical and $mh(T[w]) = 1$, then $T[w]$ is star rooted in w, i.e. w is the only vertex of degree at least 2 but can also have degree 1 if $|V(T[w])| = 2$.

The upcoming corollary, obtained from Lemma 15, describes how to compute $mh(T)$ of a rooted tree T, bottom-up, from its leaves to the root, such that, for any $u \in V(T)$, $mh(T[u])$ is computed from the values $(mh(T[u_i]))_{u_i \text{ child of } u}$ and from the critical vertices the subtrees $T[u_i]$ contain.

The next lemma is used throught the proof of Corollary 4 and Lemma 17.

Lemma 16. *Let T be any tree rooted in a vertex $v \in V(T)$. Let $k = \max_{1 \leq i \leq d}\{mh(T[v_i]\}$, where $v_1, \ldots, v_d$ are the children of v in T. For any vertex w in T, there is at most 1 branch at w in T that has monotone hunter number $k+1$.*

Proof. Note first that, by definition of k, there is no branch at v that has monotone hunter number $k+1$. Hence, let w be any vertex $V(T) \setminus \{v\}$ and let us denote by x the child of v such that $w \in V(T[x])$. By definition of k, $mh(T[x]) \leq k$. Let us denote by y the parent of w in $T[x] \cup \{v\}$ Note that for any vertex in $z \in N(w) \setminus \{y\}$, z is a child of w and $T[z]$ is a connected subgraph of $T[x]$. Thus, by Lemma 8, $mh(T[z]) \leq k$ and so there is at most 1 branch at w that has monotone hunter number $k+1$. □

Corollary 4. *Let T be a rooted tree, $u \in V(T)$ and let $u_1, \ldots, u_d$ be the d children of u in T. Let us order the children of u such that $k = mh(T[u_1]) \geq mh(T[u_2]) \geq \ldots mh(T[u_d])$.*

1. *If $d = 0$, then $mh(T[u]) = 0$;*
2. *If $k = 0$, then $mh(T[u]) = 1$;*
3. *If $k = 1$, $d = 1$ and the only child of u is not 1-critical, then $mh(T[u]) = 1$;*

4. *If $k = 1$ and $d = 1$ and the only child of u is 1-critical, then $mh(T[u]) = 2$;*

5. *If $k = 1$ and $d \geq 2$, then $mh(T[u]) = 2$;*

6. *If $k > 1$ and $mh(T[u_3]) = k$, then $mh(T[u]) = k + 1$;*

7. *If $k > 1$, $mh(T[u_2]) = k$, and $T[u_1]$ or $T[u_2]$ contains a k-critical vertex, then $mh(T[u]) = k + 1$.*

8. *If $k > 1$, $mh(T[u_2]) = k$ ($mh(T[u_3]) < k$ or $d = 2$), and neither $T[u_1]$ nor $T[u_2]$ contains a k-critical vertex, then $mh(T[u]) = k$.*

9. *If $k > 1$, $mh(T[u_1]) = k$, $T[u_1]$ contains a k-critical vertex x and $mh(T[u, x]) = k$, then $mh(T[u]) = k + 1$;*

10. *If $k > 1$, $mh(T[u_1]) = k$ ($mh(T[u_2]) < k$ or $d = 1$) and $T[u_1]$ contains a k-critical vertex x and $mh(T[u, x]) < k$, then $mh(T[u]) = k$;*

11. *If $k > 1$, $mh(T[u_1]) = k$ ($mh(T[u_2]) < k$ or $d = 1$) and $T[u_1]$ does not contain any k-critical vertex, then $mh(T[u]) = k$.*

Proof. Each statement can be shown using Lemma 15.

1. If $d = 0$, then $T[u]$ is a single vertex and by Lemma 15, $mh(T[u]) = 0$.

2. If $k = 0$, then there exists at least one child w of u such that $mh(T[w]) = 0$ and there is no child w' of u such that $mh(T[w']) > 0$. Thus, $T[u]$ contains at least one edge and is a star graph rooted in u. By Lemma 15, we get that $mh(T[u]) = 1$.

3. If $k = 1$, $d = 1$ and the only child of u is not 1-critical, i.e. u_1 is the only child of u and $mh(T[u_1]) = 1$. Thus, $T[u_1]$ is a star rooted at u_1. Therefore, $T[u]$ is also star rooted at u_1, and so, by Lemma 15, $mh(T[u]) = 1$.

4. If $k = 1$, $d = 1$ and the only child of u is 1-critical, i.e. u_1 is the only child of u, $mh(T[u_1]) = 1$ and there exists a child w of u_1 such that $mh(T[w']) = 1$. By Lemma 15, $T[w]$ contains at least 1 edge. Let v be the child of w in $T[w]$. Thus, $T[u]$ contains the path (v, w, u_1, u) and so is not a star. Moreover, $T[u] \setminus u$ is a forest of spider, and so, by Lemma 15, $mh(T[u]) = 2$.

5. If $k = 1$ and $d \geq 2$, then u has at least two child. Thus, By Lemma 15, $T[u_1]$ contains at least one edge. Let w be a child of u_1 in $T[u_1]$. Note that $T[u]$ contains the path $P = (u_2, u, u_1, w)$ as subgraph and that $T[u] \setminus u$ is a forest of spider. Therefore, by Lemma 15, $mh(T[u]) = 2$.

6. If $k > 1$ and $mh(T[v_3]) = k$, then $mh(T[u]) \geq k+1$ by Lemma 15. Moreover, by Lemma 16 and Lemma 15, we get that $mh(T[u]) = k + 1$.

7. If $k > 1$, $mh(T[u_1]) = mh(T[u_2]) = k$ and $T[u_1]$ or $T[u_2]$ contains a k-critical vertex. Let x be a k-critical vertex in $T[u] \setminus \{u\}$. W.l.o.g., let us assume that $x \in T[u_1]$. Let us denote by y the parent of x in $T[u]$. Let us denote by B_1 and B_2 the two branch at x in $T[x]$ such that $mh(B_1) = mh(B_2) = k$. Let us also denote by B_y the branch at x in $T[u]$ that contains y. Note that B_y contains $T[u_2]$ as subgraph. Thus, by Lemma 8, $mh(B_y) \geq mh(T[u_2]) = k$. Therefore, by Lemma 15, we get that $mh(T[u]) \geq k + 1$. Finally, by Lemma 16 and Lemma 15, $mh(T[u]) \leq k + 1$, and so $mh(T[u]) = k + 1$.

8. If $k > 1$, $mh(T[u_1]) = mh(T[u_2]) = k$ ($mh(T[v_3]) < k$ or $d = 2$) and neither $T[u_1]$ nor $T[u_2]$ contains a k-critical vertex, then u does not have three branches that require monotone hunter number at least k. Note also that for any $w \in T[u_1]$ (resp. $w \in T[u_2]$), w is not k-critical and so there is at most 1 branch at w in $T[w]$ that has monotone hunter number at least k. Therefore, since every branch at w in $T[u]$ is also a branch at w in $T[u_1]$, except the one containing the parent of w, we get that there are at most 2 branches at w that has monotone hunter number at least k. Finally, if $d > 2$, for any $2 < j \leq d$ and any vertex $w \in T[u_j]$, let us denote by y the parent of w in $T[u_j] \cup \{u\}$. Note that for any vertex $z \in N(w) \setminus \{y\}$, $T[z]$ is a subgraph of $T[u_j]$ and so by Lemma 8 $mh(T[z]) \leq mh(T[u_j]) < k$. Therefore, there is no vertex $w \in T[u]$ such that u has 3 branches that have monotone hunter number at least k. Hence, by Lemma 15, $mh(T[u]) \leq k$. By Lemma 8 and because $mh(T[u_1]) = k$, $mh(T[u]) = k$.

9. If $k > 1$ and $mh(T[u_1]) = k$, $T[u_1]$ contains a k-critical vertex x and $mh(T[u, x]) = k$, then clearly, x has 3 branches with monotone hunter number at least k in $T[u]$. Therefore, by Lemma 15, $mh(T[u]) \geq k+1$. Finally, by Lemma 16 and Lemma 15, $mh(T[u]) \leq k+1$ and so $mh(T[u]) = k + 1$.

10. If $k > 1$ and $mh(T[u_1]) = k$, ($mh(T[u_2]) < k$ or $d = 1$) and $T[u_1]$ contains a k-critical vertex x and $mh(T[u, x]) < k$, then, obviously, u does not have 3 branch that require monotone hunter number at least k. Let w be any vertex of $T[u] \setminus \{u\}$. Let us assume first that w is not k-critical. Thus, there is at most one branch at w in $T[w]$ that has monotone hunter number k. Obviously, since there is only one other branch left for w in $T[u]$ (the one containing its parent), w does not have 3 branches that require monotone hunter number at least k.

    Let us assume now that w is k-critical. Thus, $w \in T[u_1]$ since $mh(T[u_j]) < k$ for any $2 \leq j \leq d$. Note that $w = x$, otherwise, either the branch at w containing its parent in $T[u_1]$ contains $T[x]$ as subgraph or the branch at x containing its parent in $T[u_1]$ contains $T[w]$, and so by Lemma 8, there exists a vertex in $T[u_1]$ that has 3 branches that has monotone hunter number k. Therefore, by Lemma 15, $mh(T[u_1]) = k + 1$, a contradiction. Let us recall that by hypothesis, $mh(T[u, x]) < k$. Hence, there is only two branches at x that have monotone hunter number at least k (otherwise, by Lemma 15 and Lemma 8, $k \leq mh(T[x]) \leq mh(T[u_1])$, a contradiction). Therefore, by Lemma 15, $mh(T[u]) \leq k$ and by Lemma 8, $mh(T[u]) \geq mh(T[u_1]) \geq k$.

11. If $k > 1$, $mh(T[u_1]) = k$ ($mh(T[v_2]) < k$ or $d = 1$), and $T[u_1]$ does not contain any k-critical vertex, then there is no vertex in $T[u]$ that has 3 branches that require monotone hunter number at least k (as in previous case). Therefore, by Lemma 15, $mh(T[u]) \leq k$ and by Lemma 8, $mh(T[u]) \geq mh(T[u_1]) \geq k$ and so $mh(T[u]) = k$.

□

Definition 1. *For any tree $T[u]$, the label of u is a list of integers $(a_1, \ldots, a_p)$, where $a_1 > a_2 > \cdots > a_p \geq 0$, and such that there exists a set of vertices $\{\ell_1, \ldots, \ell_p\}$ such that:*

- $mh(T[u]) = a_1$;
- *For $1 \leq i < p$, $mh(T[u, \ell_1, \ldots, \ell_i]) = a_{i+1}$ and ℓ_i is an a_i-critical vertex in $T[u, \ell_1, \ldots, \ell_{i-1}]$. We will say that ℓ_i is associated to a_i;*
- *$\ell_p = u$. If a_p is not marked with a star ($*$), then there is no a_p-critical vertex in $T[u, \ell_1, \ldots, \ell_{p-1}]$. If a_p is marked (with a star), then ℓ_p is an a_p-critical vertex. In both cases, $T[u, \ell_p] = T[u, u]$ is the empty tree.*

Note that, for the algorithm, we will need to add some value a to a label $\lambda = (a_1, \dots a_p)$. This operation will be done with the operator &, i.e. $\lambda' = a\&\lambda = (a'_1 = a, a'_2 = a_1, \dots, a'_{p'-1} = a_{p-1}, a'_{p'} = a_p)$. Let us stress that, by the definition of a label, for any $1 \leq i < p$, a_i is marked by a star and for any $1 \leq i < j \leq p$, $a_i > a_j$. Thus, the operation & will only be used when a is marked by a star and $a > a_1$.

To better understand the above definition, allow us to provide two examples. Consider first, a tree T_1 rooted in a vertex u_1 with the label $(3^*, 2^*, 1)$. First since $a_1 = 3^*$, we get that $mh(T[u]) = 3$. Moreover, there exists a vertex $\ell_1 \in T[u]$ that is 3-critical, meaning that it has two children both containing 3 in their labellings. Moreover, $mh(T[u, \ell_1]) = 2$. But, there exists a vertex $\ell_2 \in T[u, \ell_1]$ that is 2-critical. Finally, $mh(T[u, \ell_1, \ell_2]) = 1$ and since $a_3 = 1$, there is no 1-critical vertex. Let us recall that the definition of 1-critical is different. A vertex v is 1-critical if and only if $T[v] = P_3$ and such that v is an endpoint of P_3. Thus, since $mh(T[u, \ell_1, \ell_2]) = 1$, $T[u, \ell_1, \ell_2]$ is an edge.

For the second example, consider T_2, a tree rooted at u_2 with the label $(3^*, 2^*, 1^*)$. This time, after defining ℓ_1 and ℓ_2 as the 3-critical and the 2-critical vertices, respectively, there is ℓ_3 that is also a 1-critical vertex in $T[u, \ell_1, \ell_2]$. But since $1^* = a_p$ is the last label, $\ell_3 = u$ and so $T[u, \ell_1, \ell_2, \ell_3]$ is the empty graph.

Let us consider now T'_1 and T'_2 both obtained by adding a vertex u' adjacent respectively to u_1 and u_2. Moreover, u' is the new root of T'_1 and T'_2. Therefore, the label of u' in T'_1 is $(3^*, 2^*, 1^*)$, but the label of u' in T'_2 is (4).

Lemma 17. *Let T be a tree, rooted in a vertex u in $V(T)$. Algorithm 1 with input T and any $v \in V(T)$, returns the label of v in $T[v]$.*

Proof. Our goal is to prove that the algorithm computes the label of v in $T[v]$, denoted as $\lambda(v, T[v]) = (a^*_1, \dots, a^*_{p^*})$. Let $d_{T[v]}(v) = d$.

Observe first that if $d = 0$, by Corollary 4, $mh(T[v]) = 0$, and so, by definition of a label, $\lambda(v, T[v]) = (0)$. Note that Algorithm 1 computes (0), thus let us assume that $d > 0$.

Let $v_1, \dots v_d$ be the d children of v in $T[v]$. For any $1 \leq i \leq d$, let $\lambda_i = (a^i_1, \dots, a^i_{p_i})$ be the label of v_i in $T[v_i]$. That is, $\lambda_i = \lambda(v_i, T[v_i]) = (a^i_1, \dots, a^i_{p_i})$. W.l.o.g., let us assume that $a^1_1 \geq a^2_1 \geq \dots \geq a^d_1$. Let $k = a^1_1 = \max_{i \in \{1,\dots,d\}} \max_{q \in \{1,\dots,|\lambda_i|\}} a^i_q$.

For each $1 \leq i \leq d$, let $T^i_k = T[v_i]$. Also, for $0 \leq m \leq k$, if $m = a^i_j$ for any $1 \leq j \leq p_i$, let T^i_{m-1} be obtained from T^i_m by removing $T[\ell^i_j]$ where ℓ^i_j is the vertex associated to a^i_j. Finally, for any $0 \leq m \leq k$, let T_m be the subtree of $T[u]$ induced by $\bigcup_{1 \leq i \leq d} V(T^i_m)$.

Let us prove by induction on $0 \leq m \leq k$ that after the $m+1$-th step of the loop of the algorithm, the current value of λ, the variable of Algorithm 1, denoted by λ^m, corresponds to the label $\lambda(v, T_m)$. If this induction holds, then, when $m = k$, the algorithm returns $\lambda^k = \lambda(v, T_k) = \lambda(v, T[v])$, which concludes the proof.

Let n be the number of children w of v, such that m is in the label of w.

The base case is for $m = 0$.

- Let us assume first that $n \geq 1$. By Case 2 of Corollary 4, $mh(T_0) = 1$. Recall that, by the definition of T_0, for any child w of v, $mh(T[w]) \leq 0$, and so v is not 1-critical in T_0. Thus, by the definition of a label, $\lambda(v, T_0) = (1)$, which corresponds to λ^0.
- Let us assume now that $n = 0$. By Case 1 of Corollary 4, $mh(T_0) = 0$. Thus, $\lambda(v, T_0) = (0)$ (because 0-critical vertices do not exist). Moreover, since, $n = 0$, $\lambda^0 = (0)$.

Let us assume now that $m = 1$. It follows from the case $m = 0$ that $\lambda^0 = \lambda(v, T_0)$. Before analysing the several subcases for $m = 1$, let us first prove that the label $(1^*, 0)$ does not exist.

Algorithm 1 *Compute_label*(T: tree, u: vertex) return the label of u. Let $v_1, \ldots, v_d$ be the d children of u in $T[u]$ and let $\lambda_1, \ldots, \lambda_d$ be their corresponding labels

if $d = 0$ **then**
  return (0);
end if
Let $\lambda = ()$ and let $k = \max_{i \in \{1,\ldots,d\}} \max_{q \in \{1,\ldots,|\lambda_i|\}} \lambda_i$;
for m from 0 to k **do**
  Let n be the number of children with m in their label;
  if $m = 0$ **then**
    if $n \geq 1$ **then**
      $\lambda \leftarrow (1)$;
    else
      $\lambda \leftarrow (0)$;
    end if
  else if $m = 1$ **then**
    if $n = 1$ and element m is not critical and $\lambda = (0)$ **then**
      $\lambda \leftarrow (1^*)$; *//Case 3*
    else if $n \geq 1$ **then**
      $\lambda \leftarrow (2)$; *//Cases 4 and 5*
    end if
  else if $m > 1$ **then**
    if $n \geq 3$ **then**
      $\lambda \leftarrow (m+1)$; *//Case 6*
    else if $n = 2$ and at least one element m is critical **then**
      $\lambda \leftarrow (m+1)$; *//Case 7*
    else if $n = 2$ and no element m is critical **then**
      $\lambda \leftarrow (m^*)$; *//Case 8*
    else if $n = 1$, the element m is critical and $m \in \lambda$ **then**
      $\lambda \leftarrow (m+1)$; *//Case 9*
    else if $n = 1$, the element m is critical and $m \notin \lambda$ **then**
      $\lambda \leftarrow (m^*)\&\lambda$; *//Case 10*
    else if $n = 1$ and the element m is not critical **then**
      $\lambda \leftarrow (m)$; *//Case 11*
    end if
  end if
end for
return λ;

Towards a contradiction, assume that this label does exist. Then, by the definition of a label, there exists a vertex $w \neq v$ in $T[v]$ such that w is 1-critical. By the definition of a vertex being 1-critical, we get that w has a child y such that $mh(T[y]) = 1$. Since $mh(T[y]) = 1$, we get that $T[y]$ contains at least one edge and so there exists $x \in N(y)$. Since $T[v]$ is connected, there exists a (w, v)-path $P = (p_1 = w, \dots p_q = v)$ containing at least two vertices w and v. Thus, $P' = (x, y, p_1 = w, \dots, p_q = v)$ is a path with at least 4 vertices. Hence, $mh(T[v]) > 1$ and which contradicts that the label $(1^*, 0)$ exists by the definition of a label.

- Assume first that $n = 0$. This implies that $T_1 = T_0$, and so, by the case $m = 0$, $\lambda^0 = \lambda(v, T_0)$ and we get that $\lambda^1 = \lambda(v, T_1)$.
- Assume next that $n = 1$, and let v_i be the child of v such that (1) or (1^*) is in the label of v_i.
  - Let us also assume that "element m is not critical", i.e. (1) is in the label of v_i (not (1^*)), and $\lambda^0 = (0)$. Recall that since $\lambda^0 = (0)$, and by the case $m = 0$, T_0 is a single vertex. Thus, v has only one child in T_1 , and so, by Case 3 of Corollary 4, $mh(T_1) = 1$. Moreover, since $n = 1$, we get that v is 1-critical in T_1. Thus, $\lambda(v, T_1) = (1^*)$ (since $(1^*, 0)$ is not a label), which, by the hypothesis, corresponds to λ^1.
  - Let us also assume that "element m is not critical", i.e. (1) is in the label of v_i (not (1^*)), and $\lambda^0 = (1)$. Observe that if $\lambda^0 = (1)$, and by the case $m = 0$, then T_0 contains at least one edge and so there exist at least one child w of v such that $mh(T[w]) = 0$. Thus, v has at least 2 child, and so, by Cases 5 of Corollary 4, $mh(T_1) = 2$. By Lemma 16, for any vertex $w \in T_1$, there is at most 1 branch at w that has monotone hunter number at least 2. Therefore, $\lambda(v, T_1) = (2)$, which corresponds to λ^1 by hypothesis.
  - Let us also assume that "element m is critical", i.e. (1^*) is in the label of v_i. By Case 4 of Corollary 4, $mh(T_1) = 2$. By Lemma 16, for any vertex $w \in T_1$, there is at most 1 branch at w that has monotone hunter number at least 2. Thus, $\lambda(v, T_1) = (2)$, which corresponds to λ^1 by hypothesis.
- Finally, assume that $n \geq 2$. By Case 5 of Corollary 4, $mh(T_1) = 2$. Moreover, by Lemma 16, for any vertex $w \in T_1$, there is at most 1 branch at w that has monotone hunter number at least 2. Therefore, $\lambda(v, T_1) = (2)$, which corresponds to λ^1 by hypothesis.

We are now ready to prove the induction step. Let us assume that, for any $2 \leq i < k$, we have $\lambda(v, T_i) = \lambda^i$. We will prove that $\lambda(v, T_{i+1}) = \lambda^{i+1}$. Observe that it suffices to deal with the cases $6 - 11$ of Corollary 4.

- **Case 6** Let us assume that $n \geq 3$. Then by Case 6 of Corollary 4, $mh(T_{i+1}) = i + 2$. By Lemma 16, for any vertex $w \in T_{i+1}$, there is at most 1 branch at w that has monotone hunter number at least $i + 2$. Thus, $\lambda(v, T_{i+1}) = (i + 2)$, which corresponds to λ^{i+1} by hypothesis.
- **Case 7** Let us assume that $n = 2$ and that at least one of the element $i + 1$ is critical, i.e. there exists a child of v containing a $i + 1$-critical vertex. Then, by Case 7 of Corollary 4, $mh(T_{i+1}) = i + 2$. By Lemma 16, for any vertex $w \in T_{i+1}$, there is at most 1 branch at w that has monotone hunter number at least $i + 2$. Thus, $\lambda(v, T_{i+1}) = (i + 2)$, which corresponds to λ^{i+1} by hypothesis.

- **Case 8** Let us assume that $n = 2$ and that no element $i+1$ is critical, i.e. that there is no $i+1$-critical vertex in $T[v] \setminus \{v\}$. Then, by Case 8 of Corollary 4, $mh(T_{i+1}) = i+1$. Note also that v is clearly $i+1$-critical. Thus, by the definition of a label and since v is $i+1$-critical, $\lambda(v, T_{i+1}) = (i+1^*)$ which corresponds to λ^{i+1} by hypothesis.

- **Case 9** Let us assume that $n = 1$. Let w be the child of v such that $i+1$ or $i+1^*$ is in $\lambda(v, T[v])$. Let us also assume that $i+1^*$ is in $\lambda(v, T[v])$, i.e. "the element m is critical". Let us denote by w' the vertex in $T[w]$ that is $i+1$-critical. Let us finally assume that $i+1 \in \lambda^i$, i.e. $mh(T_1[v,w]) \geq i+1$. Thus, by Case 9 of Corollary 4, $mh(T_{i+1}) = i+2$. Moreover, by Lemma 16, for any vertex $x \in T_{i+1}$, there is at most 1 branch at x that has monotone hunter number at least $i+2$. Thus, $\lambda(v, T_{i+1}) = (i+2)$, which corresponds to λ^{i+1} by hypothesis.

- **Case 10** Let us assume that $n = 1$. Let w be the child of v such that $i+1$ or $i+1^*$ is in $\lambda(v, T[v])$. Let us also assume that $i+1^*$ is in $\lambda(v, T[v])$, i.e. "the element m is critical". Let us denote by w' the vertex in $T[w]$ that is $i+1$-critical. Let us finally assume that $i+1 \notin \lambda^i$, i.e. $mh(T_1[v,w]) = i$. Then, by Case 10 of Corollary 4, $mh(T_{i+1}) = i+1$. Let $\lambda^i = (a_1, \ldots, a_p)$. Recall that by the definition of a label, $a_1, \ldots a_{p-1}$ are all marked by a star. Moreover, there exist vertices $\ell_1, \ell_2, \ldots, \ell_{p-1}$ that are $a_1, \ldots, a_{p-1}$-critical in $T_i[v], T_i[v, \ell_1], \ldots T_i[v, \ell_1, \ldots, \ell_{p-2}]$, respectively. Note also that by the definition of T_i, $w \notin T_i$. Hence, $T_{i+1}[v,w]$ contains T_i as a subgraph. Therefore, $\ell_1, \ell_2, \ldots, \ell_{p-1}$ are also $a_1, \ldots, a_{p-1}$-critical vertices of $T_{i+1}[w], \ldots, T_{i+1}[v, w, \ell_1], \ldots T_{i+1}[v, w, \ell_1, \ldots, \ell_{p-2}]$, respectively. We can conclude that $\lambda(v, T_{i+1}) = (i+1^*)\&\lambda^i$, which corresponds to λ^{i+1} by hypothesis.

- **Case 11** Let us assume that $n = 1$. Let w be the child of v such that $i+1$ or $i+1^*$ in $\lambda(v, T[v])$. Let us also assume that $i+1$ in $\lambda(v, T[v])$, i.e. "the element m is not critical". By Case 11 of Corollary 4, $mh(T_{i+1}) = i+1$. Let us recall that there is no $i+1$-critical vertex in $T[w]$ or in T_i (by definition of T_i). Note also that v is not $i+1$-critical since $n = 1$. Therefore, $\lambda(v, T_{i+1}) = (i+1)$, which corresponds to λ^{i+1} by hypothesis.

□

Thus we have the following:

Theorem 6. *The monotone hunter number of any tree can be computed in polynomial time.*

5 Monotone hunter number in the red variant in trees

So far, we have investigated the Hunters and Rabbit game with the additional monotonicity property since monotone strategies are often easier to deal with. Previous works on the Hunters and Rabbit game in bipartite graphs $G = (V_r \cup V_w, E)$ have shown that studying the *red variant* of the Hunters and Rabbit game, i.e., when the rabbit is constrained to start in a vertex in V_r, could be very fruitful. For instance, recall Lemma 5 which states that $h(G) = h_{V_r}(G)$ for every bipartite graph $G = (V_r \cup V_w, E)$ and which helped to get many results on the Hunters and Rabbit game [1, 6, 15]. Therefore, it is interesting to consider the monotonicity constraint when restricted to the red variant of the Hunters and Rabbit game. This section is dedicated to this study in the case of trees. Recall that $mh_{V_r}(G)$ denotes the minimum number of hunters required to win against a rabbit starting at V_r in a bipartite graph $G = (V_r \cup V_w, E)$ and in a monotone way. It can be checked that, in [15], it is actually shown that, for any tree T, $mh_{V_r}(T) \leq \lceil \frac{\log_2 |V(T)|}{2} \rceil$ (they used a monotone strategy with respect to V_r). Therefore, the proof of Corollary 2 actually shows that:

Corollary 5. *There exists $\varepsilon > 0$ such that, for any $k \in \mathbb{N}$, there exists a tree T with $mh_{V_r}(T) \geq k$ and $mh(T) \geq (1+\varepsilon)mh_{V_r}(T)$.*

Therefore, Proposition 4 and Corollary 5 already show that there exist graphs G for which $mh_{V_r}(G) < mh(G)$. The main result of this section is that there exists an infinite family of trees T such that the difference between $mh_{V_r}(T)$ and $h_{V_r}(T)$ is arbitrarily large. In particular, this improves the result of Corollary 2 since $mh_{V_r}(G) \leq mh(G)$ and $h_{V_r}(G) = h(G)$ for any graph G.

More precisely, this section is devoted to proving:

Theorem 7. *For every $i \geq 3$, there exists a tree T such that $mh_{V_r}(T) \geq i$ and $h_{V_r}(T) = 2$.*

Proof. In Section 5.1, we define a family $(T_{i,2i})_{i\geq 3}$ of trees such that $h_{V_r}(T_{i,2i}) = 2$ for every $i \geq 3$ (Lemma 24). Then, in Section 5.2, Lemma 27 proves that $mh_{V_r}(T_{i,2i}) \geq i$ for every $i \geq 3$. □

In order to prove the Lemmas 24 and 27 below, we first need to adapt several technical lemmas and propositions above in the case of the red variant in bipartite graphs. Since the proofs of Proposition 6 and Lemma 18 (in the red variant) share many similarities with the previous, already proven, versions, we decided to postpone their proofs in the appendix.

Next proposition is an adaptation of Proposition 2 in the red variant of the game.

Proposition 6. *Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a hunter strategy in a bipartite graph $G = (V_r \cup V_w, E)$ with respect to V_r. Let $v \in V_r$ (resp. $v \in V_w$) and $1 \leq i \leq \ell$. If there exists a vertex $u \in N(v)$ and a vertex $x \in N(u)$ (possibly $x = v$) such that $u, x \notin \bigcup_{j\leq i} S_j$, then $v \in Z_{2p}$ for every $2p \leq i$ (resp. $v \in Z_{2p+1}$ for every $2p+1 \leq i$).*

Next lemma adapts Lemma 6 in the red variant of the game.

Lemma 18. *Let $G = (V_r \cup V_w, E)$ be a bipartite graph with at least two vertices. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a monotone hunter strategy in G with respect to V_r. For any $0 \leq p < i \leq \lceil \ell/2 \rceil$, $Z_{2i} \subseteq Z_{2p}$ and $Z_{2i+1} \subseteq Z_{2p+1}$.*

Lemma 19 is a direct adaptation, in the red variant of the game, of Lemma 7. The only difference in their proofs is that, in the case of Lemma 19, Proposition 6 must be used instead of Proposition 2. Therefore, we omit the proof of Lemma 19.

Lemma 19. *Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a non-monotone winning hunter strategy in a bipartite graph $G = (V_r \cup V_w, E)$ with respect to V_r. Then, there exist a vertex $v \in V$ and $1 \leq i \leq \ell$ such that $v \in Z_{i-1} \setminus S_i$ and $v \in \bigcup_{p<i} S_p$.*

Lemma 20 is a direct adaptation, in the red variant of the game, of Lemma 8. The only difference in their proofs is that, in the case of Lemma 20, Proposition 6 and Lemma 19 must be used instead of Proposition 2 and of Lemma 7. Therefore, we omit the proof of Lemma 20.

Lemma 20. *For any non-empty connected subgraph H of a bipartite graph $G = (V_r \cup V_w, E)$, $mh_{V_r \cap V(H)}(H) \leq mh_{V_r}(G)$. Moreover, if $|V(H)| > 1$, we get that, if there exists a monotone winning hunter strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ in G with respect to V_r, then there exists a monotone winning hunter strategy $\mathcal{S}'$ in H with respect to $V_r \cap V(H)$ using at most $\max_{1\leq i\leq \ell} |S_i \cap V(H)|$ hunters.*

Since the proofs of upcoming Lemmas 21 and 22(in the red variant) share many similarities with the proofs of Lemmas 9 and 10, respectively, we decided to postpone their proofs in the appendix.

Lemma 21. *For any bipartite graph $G = (V_r \cup V_w, E)$ and any $k \geq mh_{V_r}(G)$, there exists a parsimonious monotone winning hunter strategy in G with respect to V_r and that uses k hunters.*

Lemma 22. *Let $G = (V_r \cup V_w, E)$ be a bipartite graph and $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a parsimonious monotone winning hunter strategy with respect to V_r.*

- *If there exist $1 \leq i < j \leq \ell$ such that $v \in S_i \cap S_j$, then $v \in S_{i+2}$.*
- *If $v \in V_r$ (resp., $v \in V_w$) and there exists an odd (resp., even) integer $1 \leq i < \ell$ such that $v \notin Z_{i-1}$, then $v \notin S_j$ for every $j \geq i$.*

5.1 The family of trees $(T_{i,q})_{i\geq 3, q\geq 6}$: definition and hunter number

Thomas: In this section, we will prove that the gap between the hunter number and the monotone hunter number in the red variant of the game may be arbitrary large. More precisely, we will design an infinite family $(T_i)_{i\geq 3}$ of trees which exhibits this behaviour.

Let $S_{k,q}$ be the rooted tree obtained from $q \geq 6$ paths of length $k \geq 3$ (with k edges) by identifying an endpoint of each path into a common vertex called the *root* of $S_{k,q}$ and denoted by c. Equivalently, $S_{k,q}$ can be obtained from a star with root c of degree q by subdividing each edge $k-1$ times. From now on, let (V_r, V_w) be the bipartition of $V(S_{k,q})$ and let us assume that $c \in V_r$.

Lemma 23. *For any $k, q \in \mathbb{N}$ such that $k \geq 3$ and $q \geq 6$, it holds that $h(S_{k,q}) = mh_{V_r}(S_{k,q}) = 2$.*

Proof. The fact that $h(S_{k,q}) > 1$ comes from the characterisation of trees with hunter number one in [8]. W.l.o.g., let us suppose that the centre c of $S_{k,q}$ is in V_r. We now prove that $mh_{V_r}(S_{k,q}) \leq 2$ and the result then follows from Lemma 5 and Proposition 1.

The strategy $\mathcal{S}$ with respect to V_r and using two hunters proceeds as follows. At every odd round, the first hunter shoots at c. The second hunter considers sequentially each path $P = (v_1, \ldots, v_k)$ of $S_{k,q} \setminus c$ by iteratively shooting at $v_1, v_2, \ldots, v_k$ (starting by shooting v_1 at an even round).

Formally, let $P^1, \ldots, P^q$ be the q branches of c in $S_{k,q}$, and let $P^i = (v_1^i, \ldots, v_k^i)$ for every $1 \leq i \leq q$ where v_1^i is the neighbour of c in P^i. Then, the strategy $\mathcal{S}$ equals $(\{c\}, \{v_1^1\}, \{c, v_2^1\}, \{v_3^1\}, \{c, v_4^1\}, \ldots, \{v_{k-1}^1\}, \{c, v_k^1\}, \{v_1^2\}, \{c, v_2^2\}, \ldots, \{v_{j-1}^i\}, \{c, v_j^i\}, \ldots, \{c, v_k^q\})$ if k is even, and $\mathcal{S}$ equals $(\{c\}, \{v_1^1\}, \{c, v_2^1\}, \{v_3^1\}, \{c, v_4^1\}, \ldots, \{c, v_{k-1}^1\}, \{v_k^1\}, \{c\}, \{v_1^2\}, \{c, v_2^2\}, \ldots, \{v_k^{i-1}\}, \{c\}, \{v_1^i\}, \{c, v_2^i\}, \ldots, \{v_{j-1}^i\}, \{c, v_j^i\}, \ldots, \{v_k^q\})$ if k is odd.

Clearly, this is a monotone winning hunter strategy in $S_{k,q}$ with respect to V_r. □

Let us denote the strategy described in the previous proof by $\mathcal{S}_1$ and let ℓ_1 be the smallest even integer greater or equal to the length of $\mathcal{S}_1$ (this length equals to $1 + qk$ if k is even and to $q(k+1)$ otherwise).

The construction of the tree $T_{i,q}$. For every $i \geq 2$ and $q \geq 6$, let $T_{i,q}$ be the tree recursively built as follows. First, $T_{1,q} = S_{3,q}$. Then, for $i > 1$, let us assume that $T_{i-1,q}$ has been defined recursively and that there exists a winning hunter strategy, of length ℓ_{i-1}, using 2 hunters in the red variant in $T_{i-1,q}$ (this holds for $i - 1 = 1$ from the previous lemma and it will be proven to hold for every $i \geq 2$ in the next lemma). Let $T_{i,q}$ be obtained from q vertex disjoint copies $T_i^1, \ldots, T_i^q$ of $T_{i-1,q}$ and from a vertex c_i, the root of $T_{i,q}$. Then, for every $1 \leq j \leq q$, add a path P_i^j of length p_i^j (defined below) between the root c_i^j of T_i^j and c_i (that is, c_i and c_i^j are at distance p_j^i in $T_{i,q}$).

The lengths p_i^j are defined recursively as follows. Let $p_i^1 = 2$ and, for every $1 < j \leq q$, let p_i^j be the minimum even integer greater or equal to $\ell_{i-1} + \sum_{1 \leq k < j} p_i^k$ (it will be shown in the next lemma that ℓ_i equals the smallest even integer greater or equal to $q\ell_{i-1} + \sum_{1 \leq j \leq q} jp_i^j$).

Finally, let us assume that $c_i \in V_r$ and note that, since p_i^j is even, this implies that $c_i, c_i^1, \ldots, c_i^q$ all belong to V_r.

Lemma 24. *For any $i \in \mathbb{N}^*$ and $q \geq 6$, $h_{V_r}(T_{i,q}) = 2$.*

Proof. The fact that $h_{V_r}(T_{i,q}) \geq 2$ follows from Lemmas 23 and 1 and since $T_{i,q}$ contains $S_{3,q}$ as a subgraph.

We prove that $h_{V_r}(T_{i,q}) \leq 2$ by induction on i. More precisely, we prove that there exists a winning hunter strategy $\mathcal{S}_i = (S_1, \ldots, S_{\ell_i})$ for $T_{i,q}$, with respect to V_r, using 2 hunters and such that, for any $j \geq 1$, if the root c_i of $T_{i,q}$ is in Z_j, then $c_i \in S_{j+1}$. This holds for $i = 1$ by Lemma 23. Let $i > 1$ and let us assume by induction that such a strategy $\mathcal{S}_{i-1}$ has already been defined for $T_{i-1,q}$.

Recall that, for all $1 \leq j \leq q$, c_i^j denotes the root of the copy T_i^j of $T_{i-1,q}$ linked to the root c_i of T_i by a path P_i^j of length at least $\ell_{i-1} + \sum_{1 \leq k < j} p_i^k$. Moreover, recall that $c_i \in V_r$.

Let us define the strategy $\mathcal{S}_i$ as follows. It proceeds in q phases and ensures that, at every round h, if $c_i \in Z_h$, then $c_i \in S_{h+1}$ and that, for every round h arising after the j^{th} phase, $(Z_h \setminus c_i) \cap \bigcup_{k \leq j} (V(T_i^k) \cup V(P_i^k)) = \emptyset$. This implies that, at the round ℓ_i, if the rabbit is still alive, it has to be on c_i. But during the last phase, we ensured that the rabbit cannot reach c_i. Thus, $\mathcal{S}_i$ is a winning strategy.

Let $1 \leq j \leq q$, let i_0^j be the last round of phase j ($i_0^0 = 0$) and assume by induction on j that i_0^j is even. Let us moreover assume by induction on j that $(Z_{i_0^{j-1}} \setminus c_i) \cap \bigcup_{k \leq j-1} (V(T_i^k) \cup V(P_i^k)) = \emptyset$. The j^{th} phase proceeds into two sub-phases as follows.

Very informally, in the first sub-phase, the hunters "push" the rabbit toward the subtrees T_i^q, then T_i^{q-1} until the subtree T_i^j. Then, in the second sub-phase, the two hunters clear the subtree T_i^j (without the rabbit being able to leave T_i^j if it was there). The lengths of the paths linking the roots of the subtrees to c_i (illustrated in Figure 3) guarantee that the rabbit cannot reach c_i before T_i^j has been cleared.

Formally, during the first sub-phase, the first hunter shoots at c_i at every odd round. Hence, the rabbit cannot leave the component of $T_{i,q} \setminus c_i$ that it was occupying at the end of the $(j-1)^{th}$ phase. During the same first sub-phase, the second hunter sequentially shoots the vertices of $P_i^q, P_i^{q-1}, \ldots, P_i^j$ in this order and from the neighbours of c_i to the vertices $c_i^q, c_i^{q-1}, \ldots, c_i^j$. More precisely, for every $j \leq k \leq q$, let $P_i^k = (v_0^k = c_i, v_1^k, \ldots, v_{p_i^k}^k = c_i^k)$. The second hunter starts at round $i_0^j + 2$ by shooting v_1^q and then sequentially shoots $v_2^q, v_3^q, \ldots, v_{p_i^q}^q = c_i^q, v_1^{q-1}, v_2^{q-1}, \ldots, v_{p_i^{q-1}}^{q-1} = c_i^{q-1}, v_1^{q-2}, \ldots, c_i^j$. Note that, after the round when the second hunter shoots at c_i^q, if the rabbit was occupying a vertex in $T_i^q \cup P_i^q$ at the beginning of the j^{th} phase, then it must occupy a vertex at distance at least p_i^q from c_i. Similarly, after the round when the second hunter shoots at c_i^{q-1}, if the rabbit was occupying a vertex in $T_i^{q-1} \cup P_i^{q-1}$ at the beginning of the j^{th} phase, then it must occupy a vertex at distance at least p_i^{q-1} from c_i. Moreover, if the rabbit was occupying a vertex in $T_i^q \cup P_i^q$ at the beginning of the j^{th} phase, then it must occupy a vertex at distance at least $p_i^q - p_i^{q-1}$ from c_i (since there have been p_i^{q-1} rounds between the shots of c_i^q and of c_i^{q-1}). With similar arguments, and by the definition of p_i^k for $j < k \leq q$, after the round when the second hunter shoots at c_i^j, the rabbit must be at distance at least ℓ_{i-1} from c_i if it was occupying a vertex of $\bigcup_{j < k \leq q} T_i^k \cup P_i^k$ at the end of the $(j-1)^{th}$ phase. Moreover, the rabbit cannot occupy any vertex in $\bigcup_{k \leq j-1} (V(T_i^k) \cup V(P_i^k))$ since

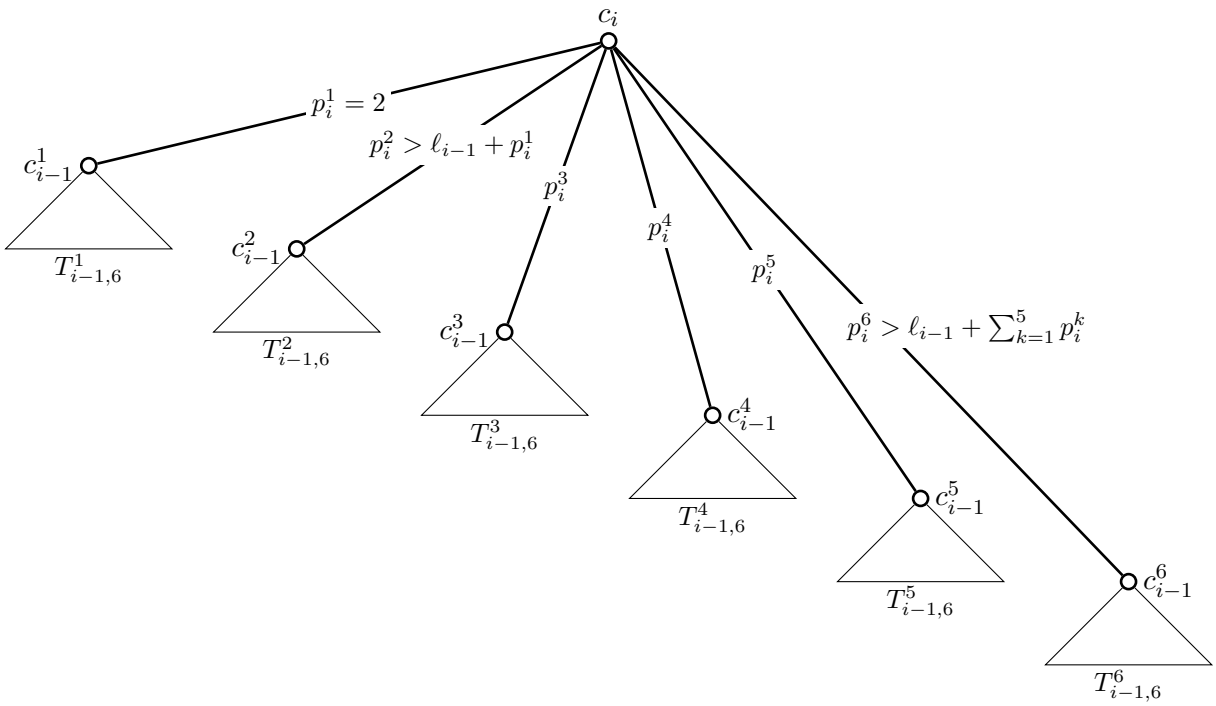


Figure 3: The graph $T_{i,6}$. The labels on the edges are used to represent their respective lengths. In particular, for every $2 \leq j \leq 6$, we have that $p_i^j > \sum_{1 \leq k \leq j-1} p_i^k + \ell_{i-1}$, where $p_i^1 = 2$ and ℓ_{i-1} is equal to the number of turns needed to clear any copy of the $T_{i-1,6}$ graph.

the first hunter is always shooting c_i during the odd rounds. Finally, the rabbit cannot occupy a vertex in P_i^j since the second hunter has just shot sequentially the vertices $v_1^j, v_2^j, \ldots, v_{p_i^j}^j = c_i^j$.

Then, the second sub-phase of phase j starts, during which both hunters execute the strategy $\mathcal{S}_{i-1}$ in the subtree T_i^j (the shot of c_i^j by the second hunter during the first sub-phase, i.e., the last round of the first sub-phase, may be used as the first round of $\mathcal{S}_{i-1}$). By the induction hypothesis, the strategy $\mathcal{S}_{i-1}$ ensures that the rabbit cannot reach the root c_i^j of T_i^j without being shot immediately (if c_i^j is in Z_j for some round j, then $c_i^j \in S_{j+1}$). Thus, if the rabbit was occupying a vertex of T_i^j at the beginning of the second sub-phase, then the rabbit cannot leave this subtree and it is eventually shot. Otherwise, because $\mathcal{S}_{i-1}$ has length at most ℓ_{i-1}, the rabbit cannot reach c_i before the last shots of the hunters in T_i^j. Let i_0^j be the last round of the phase. Note that i_0^j is even since p_i^h and ℓ_h are even for all $1 \leq h \leq q$, and i_0^{j-1} is even by induction. Then, the j^{th} phase ends after the (even) round i_0^j and with the desired property: the rabbit can only occupy a vertex in $c_i \cup \bigcup_{j<k\leq q}(V(T_i^k) \cup V(P_i^k))$, i.e., $(Z_{i_0^j} \setminus c_i) \cap \bigcup_{k \leq j}(V(T_i^k) \cup V(P_i^k)) = \emptyset$.

To conclude, note that $\mathcal{S}_i$ is winning in at most $q\ell_{i-1} + \sum_{1 \leq j \leq q} j p_i^j$ rounds since each phase j proceeds in $\ell_{i-1} + \sum_{j \leq k \leq q} p_i^k$ rounds. □

Theorem 8. *For any tree T, there exists a subdivision T' of T such that $h(T') \leq 2$.*

Proof. Let q be the maximum degree of T. Let r be any vertex of T, and let i be the eccentricity of r (i.e., the largest distance between r and some vertex of T). Then, there exists a subdivision T' of T, that is a subgraph of $T_{i,\max\{6,q\}}$ (each vertex of T being "mapped" to a vertex of degree at least 3 of $T_{i,\max\{6,q\}}$ and r being "mapped" to the root of $T_{i,\max\{6,q\}}$). By Lemmas 1, 5 and 24, $h(T') \leq 2$. □

Corollary 6. *For every $\ell \geq 0$, there exists a tree T and a subdivision T' of T such that $h(T) - h(T') \geq \ell$.*

5.2 Non monotonicity of the red variant in trees

Before proving Lemma 27, we need some additional results. Note that the next lemma is the adaptation of Proposition 5 for the red variant of the game.

Lemma 25. *Let $G = (V_r \cup V_w, E)$ be any bipartite graph and H be a connected subgraph of G. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be any parsimonious monotone winning hunter strategy in G with respect to V_r. Let $1 \leq i \leq \ell$ and $x, y \in V(H)$ such that $x \in \bigcup_{j<i} S_j$ and $y \in Z_{i-1}$ and minimising the distance between such x and y in H. If $x, y \notin S_i$, then $xy \in E(H)$.*

Proof. Note first that if $x = y$, then $\mathcal{S}$ is non-monotone since $y = x \in (\bigcup_{j<i} S_j \cap Z_{i-1}) \setminus S_i$. Hence, we may assume that $x \neq y$. Let P be a shortest path from x to y in H (it exists since H is connected). Let us assume that $S_i \subseteq V_r$ and so i is odd (the case when $S_i \subseteq V_w$ and i is even is similar). Since $y \in Z_{i-1}$ and $S_i \subseteq Z_{i-1}$ (since $\mathcal{S}$ is parsimonious), $y \in V_r$. Let a be the neighbour of x in P.

Towards a contradiction, let us assume that $a \neq y$. By the minimality of the distance between x and y, $a \notin Z_{i-1}$ and $a \notin \bigcup_{j<i} S_j$. Let $b \neq x$ be the other neighbour of a in P. If $b \neq y$, then by the minimality of the distance between x and y, $b \notin Z_{i-1}$ and $b \notin \bigcup_{j<i} S_j$. Therefore, by Proposition 6, if $a \in V_r$, then $a \in Z_i$ and if $b \in V_r$ then $b \in Z_i$. In both cases, there is a contradiction with the fact that P minimises the distance between x and y.

Therefore, we may assume that $b = y$. This implies that $x \in V_r$. Note also that $y \notin S_j$ for all $j \leq i$. Indeed, assuming otherwise would contradict the fact that $\mathcal{S}$ is monotone, since $y \notin S_i$ and $y \in Z_{i-1}$. Thus, by Proposition 6, and since $a, y \notin \bigcup_{j \leq i-1} S_j$, we have that $x \in Z_{i-1}$. This contradicts the monotonicity of $\mathcal{S}$ since $x \notin S_i$ and $x \in \bigcup_{j<i} S_j$. □

Before we prove the next lemma, we introduce an extra definition. Let $G = (V, E)$ be any graph and $\mathcal{S} = (S_1, \ldots, S_\ell)$ be any winning hunter strategy in G with respect to $X \subseteq V$. We say that $W \subseteq V$ is *definitively cleaned at the round i* if $W \cap Z_j(\mathcal{S}) = \emptyset$ and $W \cap S_{j+1} = \emptyset$ for every $i \leq j \leq \ell$.

Informally, the following lemma says that if the degree of the root r of a tree T is large enough, compared to the number of hunters, then, when a first branch of r is definitively cleaned according to any monotone hunter strategy, there must be some other branches of r whose vertices have never been shot.

Lemma 26. *Let $T = (V_r \cup V_w, E)$ be a tree rooted in some vertex $c \in V_r$ with neighbours $N(c) = \{v_1, \ldots, v_d\}$, $d \geq 2k$. For every $1 \leq i \leq d$, let B_i be the branch at c containing v_i and assume that $|V(B_i)| \geq 2$. Let $\mathcal{S} = (S_1 \ldots, S_\ell)$ be any parsimonious monotone winning hunter strategy in T with respect to V_r using at most $k-1$ hunters. Let $1 \leq j \leq \ell$ be the smallest index such that there exists an $1 \leq \alpha \leq d$ such that $V(B_\alpha)$ is definitively cleaned at the round j. Then, there exist $1 \leq \beta < \gamma \leq d$, $\alpha \notin \{\beta, \gamma\}$, such that $(\bigcup_{1 \leq i \leq j} S_i) \cap V(B_\beta) = \emptyset$ and $(\bigcup_{1 \leq i \leq j} S_i) \cap V(B_\gamma) = \emptyset$.*

Proof. Let j and α be defined as in the statement and, w.l.o.g., let us assume that $\alpha = 1$. That is, the branch B_1 is definitively cleaned at round j, and no other branch has been definitively cleaned before round j.

Claim 3. *For any vertex $c \in V(T)$ such that $d(c) \geq 3$ and any branch B at c, let q be the minimum index such that B is definitively cleaned at round q. Then, $S_q \cap V(B) \neq \emptyset$.*

Proof of Claim. Note that for any vertex $c \in V(T)$ with $d(c) \geq 3$ and any branch B at c, $V(B) > 1$.

By the minimality of q, either $Z_{q-1} \cap V(B) \neq \emptyset$ or $S_q \cap V(B) \neq \emptyset$. If $S_q \cap V(B) \neq \emptyset$, then the statement holds. Thus, let us assume that $S_q \cap V(B) = \emptyset$. Then $Z_{q-1} \cap V(B) \neq \emptyset$, as otherwise B would have been definitively cleaned at round prior to q. Let $x \in Z_{q-1} \cap V(B)$. Since $x \notin S_q$, $N_B(x) \subseteq Z_q$. Since B is connected and contains at least two vertices, we have that $N_B(x) \neq \emptyset$. Thus $Z_q \cap V(B) \neq \emptyset$, which contradicts that B is definitively cleaned at round q. $\diamond$

It follows by Claim 3 that $V(B_1) \cap S_j \neq \emptyset$. Thus, since $|S_j| \leq k-1$, there are at most $k-2$ branches at c, other than B_1, which can be shot during the round j. W.l.o.g., let us assume that $B_2, \ldots, B_{k-1}$ are the branches at c that are also shot during round j. That is, $S_j \subseteq \{c\} \cup \bigcup_{1 \leq h < k} V(B_h)$.

For purpose of contradiction, let us assume that there exists at most one branch, w.l.o.g., say B_k, such that $\bigcup_{1 \leq i \leq j} S_i \cap V(B_k) = \emptyset$. Hence, we assume that for every $k < h \leq d$, there exists $j_h < j$ and $x_h \in V(B_h) \cap S_{j_h}$.

For any $k < h \leq d$, let us denote by j_h^* the minimum index such that B_h is definitively cleaned at round j_h^*. By Claim 3, for any $k < h \leq d$, $S_{j_h^*} \cap V(B_h) \neq \emptyset$. Thus, since $V(B_h) \cap S_j = \emptyset$ for every $k < h \leq d$, we have that $j_h^* \neq j$ for every $k < h \leq d$. In particular, it follows by the minimality of j that $j_h^* > j$.

Let us prove that, for some $k < h \leq d$, there exists a vertex $y_h \in Z_{j-1} \cap V(B_h)$. Towards a contradiction, let us assume that for every $k < h \leq d$, we have $Z_{j-1} \cap V(B_h) = \emptyset$. Recall that $S_{j_h^*} \cap V(B_h) \neq \emptyset$, and let $z \in S_{j_h^*} \cap V(B_h)$ (for some $k < h \leq d$). Since $\mathcal{S}$ is parsimonious, $z \in Z_{j_h^*-1}$. Thus, there exists a rabbit trajectory $(r_0, \ldots, r_{j_h^*-1} = z)$ such that $r_q \notin S_{q+1}$ for every $0 \leq q < j_h^* - 1$. Moreover, $r_{j-1} \notin V(B_h)$, since $Z_{j-1} \cap V(B_h) = \emptyset$. Since a rabbit trajectory is a walk, and any walk between a vertex from $V \setminus V(B_h)$ to a vertex of $V(B_h)$ contains c, there exists $j - 1 \leq m < j_h^* - 1$ such that $r_m = c$. Since $r_m \notin S_{m+1}$, it follows that $v_1 \in Z_{m+1}$, where v_1 is the neighbour of c in B_1. This contradicts that B_1 is definitively cleaned at round j. Hence, there exists some vertex $y_h \in Z_{j-1} \cap V(B_h)$.

Finally, for every $k < h \leq d$, let us choose the vertices x_h and y_h, such that $x_h \in V(B_h) \cap \bigcup_{1 \leq i < j} S_i$ and $y_h \in Z_{j-1} \cap V(B_h)$ and the distance between x_h and y_h is minimised. Note that $y_h \notin \bigcup_{1 \leq i < j} S_i$, since, otherwise, $\mathcal{S}$ would not be monotone as $y_h \in Z_{j-1} \setminus S_j$. Since $S_j \cap V(B_h) = \emptyset$ and by Lemma 25, we obtain that $x_h y_h \in E(B_h)$. Thus, $x_h \in Z_j$ for every $k < h \leq d$. Since $\mathcal{S}$ is a monotone strategy, $x_h \in S_{j_h} \cap Z_j$ (with $j_h < j$), that $x_h \in S_{j+1}$ for every $k < h \leq d$. However, $d \geq 2k$ and so $|S_{j+1}| \geq k$, a contradiction. $\square$

Finally, we will need an extra definition: For any tree $T = (V_r \cup V_w, E)$, and any vertex $v \in V(T)$, let B be any branch at v such that $|V(B)| > 1$. For any strategy $\mathcal{S} = (S_1, \ldots, S_\ell)$ in T with respect to V_r, let m be the minimum integer such that $S_m \cap V(B) \neq \emptyset$ and let $u \in S_m \cap V(B)$ (by Lemma 3, such an integer m exists because $|V(B)| > 1$ and B is connected, which implies that $V_r \cap V(B) \neq \emptyset$). Let the *restriction* $\mathcal{S}_B$ of $\mathcal{S}$ be the hunter strategy, such that for every $1 \leq i \leq \ell$,

$$S'_i = \begin{cases} S_i \cap V(B), & \text{if } S_i \cap V(B) \neq \emptyset \\ \{u\}, & \text{otherwise} \end{cases}$$

Recall that, $h(\mathcal{S}_B) \leq \max_{1 \leq i \leq \ell} S_i \cap V(B)$ by Lemma 20.

Lemma 27. *For any $i \in \mathbb{N}^*$, $i \geq 3$ and $d \geq 2i$, we have that $mh_{V_r}(T_{2i,d}) \geq i$.*

Proof. Let γ_{2i} denote the root of $T_{2i,d}$. For purpose of contradiction, let us assume that $mh_{V_r}(T_{2i,d}) < i$. By Lemma 21, there exists a parsimonious monotone winning hunter strategy with respect to V_r using at most $i-1$ hunters; let that strategy be $\mathcal{S}_{2i} = (S_1^{2i}, \ldots, S_\ell^{2i})$.

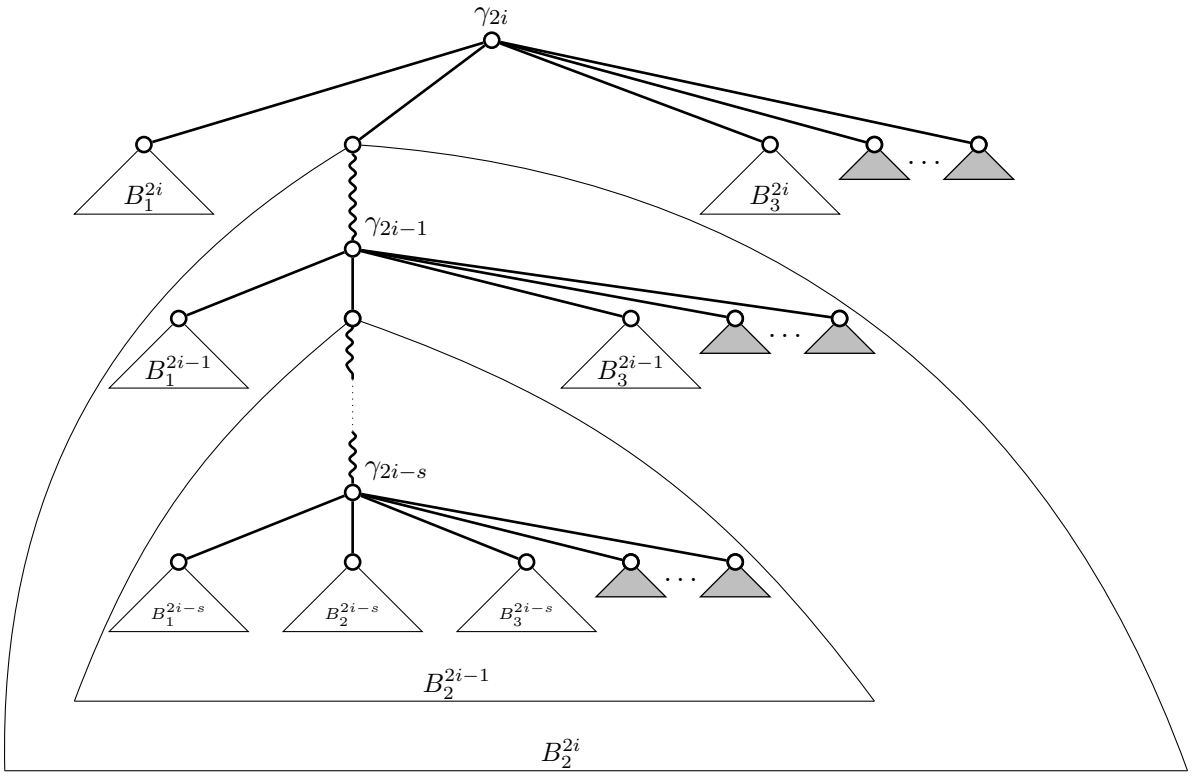


Figure 4: A representation of the tree $T_{2i,d}$ illustrating the notation used throughout the proof of Lemma 27. Wiggly edges are used to represent paths whose internal vertices have degree 2. The branch B_1^{2i} is the first branch of $T_{2i,d}$ at γ_{2i} that is definitively cleaned, and this happens during the round j_{2i}. No vertex of the branches B_2^{2i} and B_3^{2i} has been shot until the round j_{2i}. Among the branches B_2^{2i} and B_3^{2i}, the first branch that is definitively cleaned is B_2^{2i}, and this happens at the round $j'_{2i} > j_{2i}$. The colour grey is used on the small triangles to denote that we do not know the state of the corresponding branches at the same level as B_1^{2i} that are different from B_2^{2i} and B_3^{2i}. Observe that B_2^{2i} contains a copy of $T_{2i-1,d}$, rooted at γ_{2i-1}. Since B_2^{2i} is definitively cleaned at round j'_{2i}, we can iterate the same arguments, and define B_1^{2i-1} to be the first branch of T_{2i-1}, d at γ_{2i-1} that is definitively cleaned, and this happens during the round $j_{2i} < j_{2i-1} < j'_{2i}$, and so on, until we have reached the leaves of B_2^{2i}.

Let $1 \leq j_{2i} \leq \ell$ be the smallest index such that some branch at γ_{2i}, w.l.o.g., B_1^{2i}, is definitively cleaned at round j_{2i}. By Lemma 26, there exist two branches at γ_{2i}, w.l.o.g., B_2^{2i} and B_3^{2i}, such that $(\bigcup_{1\leq q\leq j_{2i}} S_q) \cap V(B_2^{2i}) = \emptyset$ and $(\bigcup_{1\leq q\leq j_{2i}} S_q) \cap V(B_3^{2i}) = \emptyset$.

Let $1 \leq j'_{2i} \leq \ell$ be the smallest index such that at least one branch B_2^{2i} or B_3^{2i} is definitively cleaned.

Note that B_2^{2i} (resp., B_3^{2i}) is connected and has at least two vertices with at least one in V_r. Therefore, by Lemma 3, at least one vertex of B_2^{2i} (resp., B_3^{2i}) must be shot before the branch is definitively cleaned . Hence, $j_{2i} < j'_{2i}$. W.l.o.g., assume that B_2^{2i} is definitively cleaned at round j'_{2i} (possibly, B_3^{2i} may also be definitively cleaned at round j'_{2i}).

We now prove by induction on $0 \leq h < 2i$, that there exist $1 \leq j_{2i} < j_{2i-1} < \cdots < j_{2i-h} < j'_{2i-h} \leq j'_{2i-h+1} \leq \cdots \leq j'_{2i-1} \leq j'_{2i} \leq \ell$ and $\{B_1^{2i-s}, B_2^{2i-s}, B_3^{2i-s}\}_{0\leq s\leq h}$ such that, for every $0 \leq s \leq h$:

1. $(\bigcup_{1\leq q\leq j_{2i-s}} S_q) \cap V(B_2^{2i-s}) = \emptyset$ and $(\bigcup_{1\leq q\leq j_{2i-s}} S_q) \cap V(B_3^{2i-s}) = \emptyset$;

2. B_1^{2i-s}, B_2^{2i-s} and B_3^{2i-s} are vertex disjoint branches at the root γ_{2i-s} of the copy of $T_{2i-s,d}$ contained in $B_2^{2i-(s-1)}$ (with $B_2^{2i+1} = T_{2i,d}$ for $s = 0$), each containing a copy of $T_{2i-(s+1),d}$ (with $T_{0,d} = \emptyset$ for $s = h = 2i-1$);

3. B_1^{2i-s} is definitively cleaned at round j_{2i-s} (not before, i.e., for every $x < j_{2i-s}$, B_1^{2i-s} is not definitively cleaned at round x), B_2^{2i-s} is definitively cleaned at round j'_{2i-s} (not before), B_3^{2i-s} is definitively cleaned at some round $x \geq j'_{2i-s}$ (not before).

See Figure 4 for an illustration of the above notation.

We have already proven that the induction hypothesis holds for $h = 0$. Let us assume that it holds for some $0 \leq h < 2i-1$ and let us show it holds for $h+1$. Let F be the copy of $T_{2i-(h+1),d}$ (rooted at $\gamma_{2i-(h+1)}$) contained in B_2^{2i-h} and let $1 \leq j_{2i-(h+1)} \leq j'_{2i-h}$ be the smallest integer such that some branch B of F at $\gamma_{2i-(h+1)}$ is definitively cleaned. Note that each branch of F at $\gamma_{2i-(h+1)}$ is connected and has at least two vertices with at least one in V_r. Therefore, by Lemma 3, at least one vertex of B must be shot before it is definitively cleaned. Hence, $j_{2i-h} < j_{2i-(h+1)}$. Let $B = B_1^{2i-(h+1)}$. Let $\mathcal{S}_{2i-(h+1)}$ denote the restriction of $\mathcal{S}_{2i-h}$ on F. By Lemma 26 considering the strategy $\mathcal{S}_{2i-(h+1)}$ on F, there exist at least two branches at $\gamma_{2i-(h+1)}$, let us denote them by $B_2^{2i-(h+1)}$ and $B_3^{2i-(h+1)}$, such that $(\bigcup_{1\leq q\leq j_{2i-(h+1)}} S_q) \cap V(B_2^{2i-(h+1)}) = \emptyset$ and $(\bigcup_{1\leq q\leq j_{2i-(h+1)}} S_q) \cap V(B_3^{2i-(h+1)}) = \emptyset$. Also, it follows by Lemma 3 that $j_{2i-(h+1)} < j'_{2i-(h+1)}$. Finally, $B_1^{2i-(h+1)}$, $B_2^{2i-(h+1)}$ and $B_3^{2i-(h+1)}$ are all contained in B_{2i-h} and, thus, $max(j_{2i-(h+1)}, j'_{2i-(h+1)}) = j'_{2i-(h+1)} \leq j'_{2i-h}$. This finishes the proof of the induction step.

For every $1 \leq s \leq 2i$, let H_s be the subgraph induced by B_1^s and B_3^s and γ_s (so H_s is connected and the subgraphs H_s and $H_{s'}$ are vertex disjoint for any $s \neq s'$). Since B_1^s has been definitively cleaned at round j_s, has at least two vertices and by Lemma 3, there exists a vertex $x'_s \in V(B_1^s) \cap \bigcup_{1\leq q\leq j_{2i}} S_q$.

Note that B_3^s is connected and has at least two vertices with at least one in V_r. Therefore, by Lemma 3, at least one vertex of B_3^s must be shot before the branch is definitively cleaned. Thus, since B_3^s is definitely cleaned at round $z_s \geq j'_s$, but not definitely cleaned at a previous round, $S_{z_s} \cap V(B_3^s) \neq \emptyset$. Moreover, since $\mathcal{S}_{2i}$ is parsimonious, any vertex $v \in S_{z_s} \cap V(B_3^s)$ is such that $v \in Z_{z_s-1}$. It follows that $(N(v) \cap Z_{z_s-2}) \setminus S_{z_s-1} \neq \emptyset$. Let $w_s \in (N(v) \cap Z_{z_s-2}) \setminus S_{z_s-1}$ and note that $w_s \in N[V(B_3^s)] = V(B_3^s) \cup \{\gamma_s\}$. Since $\mathcal{S}_{2i}$ is monotone, we get that $w_s \notin S_j$ for every $j < z_s$ and that $N(w_s) \not\subseteq S_j$ for every $j < z_s$, i.e. w_s has not been shot before round z_s. Note also that if $w_s = \gamma_s$, then the neighbour of γ_s in B_1^s is in Z_{z_s-1}, a contradiction since $z_s > j_1$ and B_1^s is definitively cleaned at round $j_s \leq j_1$. Hence, $w_s \neq \gamma_s$ and so $w_s \in V(B_3^s)$.

Similarly, there exists a vertex $w'_s \in N(w_s) \cap Z_{z_s-3} \setminus S_{z_s-2}$ such that w'_s has not been cleaned before round $z_s - 1$. Hence, we have two adjacent vertices w_s and w'_s in $N[B_3^s]$ that have never been shot before the round $z_s - 1$. Thus, since $\{w_s, w'_s\} \cap V_r \neq \emptyset$, there exists a rabbit trajectory $(\dots, w_s, w'_s, w_s, w'_s, \dots)$ consisting in oscillating between w_s and w'_s which implies that $\{w_s, w'_s\} \cap Z_j \neq \emptyset$ for all $j < z_s$. In particular, since $j_1 - 1 < z_s$, there exists a vertex $y'_s \in N[B_3^s] \cap Z_{j_1-1}$.

For every $1 \leq s \leq 2i$, let x_s and y_s be two vertices in $V(H_s)$ such that $x_s \in V(H_s) \cap \bigcup_{1\leq q\leq j_1} S_q$, $y_s \in V(H_s) \cap Z_{j_1-1}$ and the distance between x_s and y_s is minimised. Let $\mathcal{P} = \{s \mid 1 \leq s \leq 2i, y_s \in S_{j_1} \cup S_{j_1+1} \text{ or } x_s \in S_{j_1} \cup S_{j_1+1}\}$, i.e. $\mathcal{P}$ is the set of indices s such that at least one of y_s or x_s is shot during the round j_1 or $j_1 + 1$. Since $\mathcal{S}_{2i}$ uses at most $i-1$ hunters, we have that $|\mathcal{P}| \leq 2(i-1)$. Thus, let $1 \leq s^* \leq 2i$ such that $s^* \notin \mathcal{P}$, i.e., $x_{s^*}, y_{s^*} \notin S_{j_1} \cup S_{j_1+1}$. It follows from Lemma 25 that $x_{s^*}y_{s^*} \in E(T_{2i,d})$. Since $y_{s^*} \in Z_{j_{2i}-1} \setminus S_{j_1}$, we have that that $x_{s^*} \in Z_{j_1}$. However, $x_{s^*} \in \bigcup_{1\leq q\leq j_1} S_q \setminus S_{j_1+1}$ and so, x_{s^*} is recontaminated, contradicting the monotonicity of $\mathcal{S}_{2i}$. □

6 Kernelization by vertex cover

Let us first remind some of the basic definitions regarding parameterized complexity. An instance of a parameterized version Π_p of a decision problem Π is a pair (I, t), where I is an instance of Π and t is a non-negative integer, called a *parameter*, associated with I. We say that Π_p is *fixed-parameter tractable* (FPT) if there exists an algorithm (called as *FPT algorithm*) that, given an instance (I, t) of Π_p, solves it in time $f(t) \cdot |I|^{\mathcal{O}(1)}$, where f is any computable function of t.

Definition 2 (Equivalent Instances)**.** *Let Π_1 and Π_2 be two parameterized problems. Two instances, $(I, t) \in \Pi_1$ and $(I', t') \in \Pi_2$, are equivalent when (I, t) is a Yes-instance if and only if (I', t') is a Yes-instance.*

A parameterized (decision) problem Π_p admits a *kernel* of size $f(t)$, for some function f that depends only on t, if the following is true: there exists an algorithm (called a *kernelization algorithm*) that, given as input an instance (I, t) of Π_p, runs in $(|I| + t)^{\mathcal{O}(1)}$ time and outputs an equivalent instance (I', t') of Π_p such that $|I'| \leq f(t)$ and $t' \leq t$. If the function f is polynomial, then the problem is said to admit a *polynomial kernel*. It is well-known that a decidable parameterized problem is FPT if and only if it admits a kernel [11].

Recall that a *vertex cover* of a graph G, is a set $U \subseteq V(G)$ such that for every edge $uv \in E(G)$, $U \cap \{u, v\} \neq \emptyset$. The order of a minimum vertex cover of G is usually referred to as the *vertex cover number* of G and denoted by $vc(G)$. In what follows, we consider the Hunters and Rabbit Problem parameterized by the vertex cover number. That is, an instance $((G, k), t)$ is defined by an input (G, k) where the problem aims at deciding whether $h(G) \leq k$ and the parameter t is any upper bound on $vc(G)$.

First, we have the following observation.

Proposition 7. *For any connected graph G, $h(G) \leq mh(G) \leq vc(G)$.*

Proof. Let U be a vertex cover of size t in G and I be the independent set $V(G) \setminus U$. The hunter player can win simply by shooting all the vertices of U twice. If the rabbit starts at a vertex $u \in U$, then it gets shot in the first round. Otherwise, the rabbit was on a vertex $v \in I$, and then it has to move to a vertex in U (since I is an independent set) that is, $Z_1 = U$ and then, the rabbit is shot by a hunter in the next round. Finally, note that this strategy is also monotone. □

Let U be a vertex cover of size $t \geq vc(G)$ of G and I be the independent set $V(G) \setminus U$. For each subset $S \subseteq U$, we define the following equivalence class: $\mathcal{C}_S = \{v \mid v \in I \text{ and } N(v) = S\}$.

Lemma 28. *Let $G = (V, E)$ be a connected graph, $U \subseteq V$ be a vertex cover of G, $k \geq 1$ and let $S \subseteq U$ be such that $|\mathcal{C}_S| > k + 1$. Let $\mathcal{C}_S = \{v_1, \ldots, v_q\}$. Then, $h(G) \leq k$ (resp., $mh(G) \leq k$) if and only if $h(G[V \setminus \{v_{k+2}, \ldots, v_q\}]) \leq k$ (resp., $mh(G[V \setminus \{v_{k+2}, \ldots, v_q\}]) \leq k$).*

Proof. By Lemma 1, $h(G[V \setminus \{v_{k+2}, \ldots, v_q\}]) \leq h(G)$. Similarly, due to Lemma 8, $mh(G[V \setminus \{v_{k+2}, \ldots, v_q\}]) \leq mh(G)$. So, it only remains to prove that, if $h(G) > k$ (resp., $mh(G) > k$), then $h(G[V \setminus \{v_{k+2}, \ldots, v_q\}]) > k$ (resp., $mh(G[V \setminus \{v_{k+2}, \ldots, v_q\}]) > k$). Let $H = G[V \setminus \{v_{k+2}, \ldots, v_q\}]$ and let $X = \{v_1, \ldots, v_{k+1}\}$ (i.e., $X = V(H) \cap \mathcal{C}_S$).

In the following we show that if $h(G) > k$ (resp., $mh(G) > k$), then $h(H) > k$ (resp., $mh(H) > k$). To this end, we establish that if the rabbit has a winning strategy in G against k hunters, then the rabbit has a winning strategy in H against k hunters.

(1) $h(G) > k \implies h(H) > k$: Let $\mathcal{S} = (S_1, S_2, \ldots, S_\ell)$ be any hunter strategy in H using at most k hunters. Then, $\mathcal{S}$ is a hunter strategy in G using at most k hunters. Since $h(G) > k$, there

exists a rabbit-trajectory $\mathcal{R}' = (r'_0, r'_1, \ldots, r'_{\ell-1})$ in G such that $r'_i \notin S_{i+1}$ for every $0 \leq i < \ell$. Let $\mathcal{R} = (r_0, \ldots, r_{\ell-1})$ be such that, for every $0 \leq i < \ell$, let $r_i = r'_i$ if $r'_i \in V(H)$ and, otherwise, let r_i be any vertex of $X \setminus S_{i+1}$ (such a vertex exists since $|S_{i+1}| \leq k$ and $|X| > k$). Note that $r'_i \neq r_i$ only if $r'_i \notin V(H)$ and therefore $r'_i \in \mathcal{C}_S$. This implies that, if $r'_i \notin V(H)$, then $r'_{i-1}, r'_{i+1} \in S \subset V(H)$ (since $N(r'_i) = S$). Therefore, $r_{i-1} = r'_{i-1}$ and $r_{i+1} = r'_{i+1}$ and $r_{i-1}, r_{i+1} \in N_H(r_i)$ (since $r_i \in X$ and so $N_G(r_i) = N_H(r_i) = S$). Therefore, $\mathcal{R}$ is a rabbit trajectory in H and, by construction, $r_i \notin S_{i+1}$ for every $0 \leq i < \ell$. Hence, $\mathcal{S}$ is not a winning hunter strategy. Therefore, $h(H) > k$.

(2) $mh(G) > k \implies mh(H) > k$: Since $mh(G) > k$, for every hunter strategy strategy $\mathcal{S}$ in G using at most k hunters, there is a rabbit trajectory $\mathcal{R}'$ that either is a winning rabbit trajectory (the rabbit never gets shot) or recontaminates a vertex (rabbit may be shot at a later round). Now, similarly to previous case, consider $\mathcal{R}$. If the rabbit never gets shot in $\mathcal{R}'$, then due to the arguments presented in (1), the rabbit evades getting shot in $\mathcal{R}$ as well. Hence, we assume that the rabbit gets shot in $\mathcal{R}'$, during, say, a round p, but recontaminates a vertex, say x, during a round $p' < p$. Since only vertices of H can be shot in the hunter strategy $\mathcal{S}$, only the vertices of H can be recontaminated by $\mathcal{R}'$ (recall Lemma 7). Hence $x \in V(H)$. Therefore, x gets recontaminated by $\mathcal{R}$ in H as well. Thus, $mh(H) > k$. □

Finally, we present our kernelization result.

Theorem 9. *The problem that takes an n-node connected graph G and an integer $k \geq 1$ as inputs and decides whether $h(G) \leq k$, parameterized by any upper bound t on $vc(G)$, admits a kernel of size at most $4^t(t+1) + 2t$. Moreover, this problem can be solved in* FPT *time $(4^t(t+1) + 2t)^{t+1} \cdot n^{\mathcal{O}(1)}$.*

Proof. The kernelization proceeds as follows. First, if $k > t$, then answer that $h(G) \leq mh(G) \leq k$ (this is correct by Proposition 7). Otherwise, let U be a vertex cover of size at most $2t$ of G (which can be computed in time $O(tn)$ by classical 2-approximation for vertex cover using maximal matching [27]). Let H be the graph obtained from G as follows. For every $S \subseteq U$, if $|\mathcal{C}_S| > k+1$, then remove $|\mathcal{C}_S| - (k+1)$ vertices from $\mathcal{C}_S$. By Lemma 28 (applied iteratively for each $S \subseteq U$), $h(G) \leq k$ (resp., $mh(G) \leq k$) if and only if $h(H) \leq k$ (resp., $mh(H) \leq k$). Moreover, $|V(H)| = |U| + \sum_{S \subseteq U} |\mathcal{C}_S \cap V(H)| \leq 2t + 2^{2t}(k+1) \leq 4^t(t+1) + 2t$ (the last inequality holds by Proposition 7). Hence, the above algorithm is the desired kernelization algorithm.

Finally, applying the XP-algorithm [1], it can be decided in time $|V(H)|^{k+1}$ whether $h(H) \leq k$. Since, by Proposition 7, $k \leq t$, this gives the FPT algorithm that decides whether $h(G) \leq k$ (resp., $mh(G) \leq k$) in time $(4^t(t+1) + 2t)^{t+1} + n^{\mathcal{O}(1)}$. □

7 Some Future Directions

In this paper, we studied the Hunters and Rabbit game by defining the notion of monotonicity for this game. Using this notion of monotonicity, we characterized the monotone hunter number for various classes of graphs. Moreover, we established that, unlike several graph searching games, the monotonicity helps in this game, i.e., the $h(G)$ can be arbitrary larger than $mh(G)$. Furthermore, our results also establish that the monotonic hunter number does not behave "nicely" with taking minors and subdivisions, unlike its other counterparts such as pathwidth and treewidth.

There are still several challenging open questions in this area. The most important among them is the computational complexity of deciding/computing the hunter number of a graph. Although our results establish that computing $mh(G)$ is unlikely to be polynomial time solvable

(unless $\mathsf{P} = \mathsf{NP}$), the computational complexity of computing/deciding $h(G)$ remains open, even if G is restricted to be a tree graph.

We also establish that Hunters and Rabbit as well as its monotone variant are FPT parameterized by the vertex cover number, by designing exponential kernels prameterized by $vc(G)$ for both variants. It is trivial to see that both of these variants admit AND Decomposition. Since Monotone Hunters and Rabbit is most likely to be NP-hard (not solvable in polynomial time unless $\mathsf{P} = \mathsf{NP}$), it is unlikely for Monotone Hunters and Rabbit to admit a polynomial kernel parameterized by the solution size, under some reasonable complexity-theoretic assumptions. But, since Monotone Hunters and Rabbit game is closely related to the pathwidth of a graph and pathwidth admits a polynomial kernel with respect to $vc(G)$, it is hopeful that Monotone Hunters and Rabbit also admits a polynomial kernel parameterized by $vc(G)$. Moreover, the NP-Hardness result of Hunters and Rabbit would also imply that it is unlikely for Hunters and Rabbit to admit a polynomial kernel parameterized by the solution size.

Finally, we propose some open questions concerning the computation of $h(G)$ for various graph classes including trees, cographs, and interval graphs. Specifically, it will be interesting to design a polynomial time algorithm, similar to Algorithm 1, to compute $h(T)$ for a tree T, a question that was already proposed in [1]. The natural way that one could tackle this question is through the notion of monotonicity, which we defined and studied in this paper. Unfortunately, Theorem 7 implies that such an approach will not work. This means that a positive answer to this question (if any) would require the introduction of new tools and techniques. Moreover, it seems that for an $m \times n$ grid for $min(m, n) > 2$, $mh(G) = pw(G) + 1$. It might be of interest to find elegant arguments to prove that $mh(G) = pw(G) + 1$ for a grid $G_{m\times n}$ with $min(m, n) > 2$.

References

[1] Tatjana V. Abramovskaya, Fedor V. Fomin, Petr A. Golovach, and Michal Pilipczuk. How to hunt an invisible rabbit on a graph. European Journal of Combinatorics, 52:12–26, 2016. doi:10.1016/j.ejc.2015.08.002.

[2] I. Adler. Marshals, monotone marshals, and hypertree-width. Journal of Graph Theory, 47(4):275–296, 2004.

[3] Isolde Adler. Directed tree-width examples. J. Comb. Theory, Ser. B, 97(5):718–725, 2007. doi:10.1016/j.jctb.2006.12.006.

[4] Daniel Bienstock and Paul D. Seymour. Monotonicity in graph searching. Journal of Algorithms, 12(2):239–245, 1991. doi:10.1016/0196-6774(91)90003-H.

[5] Hans L. Bodlaender, John R. Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. Journal of Algorithms, 18(2):238–255, 1995.

[6] Jessalyn Bolkema and Corbin Groothuis. Hunting rabbits on the hypercube. Discrete Mathematics, 342(2):360–372, 2019. URL: https://www.sciencedirect.com/science/article/pii/S0012365X18303455, doi:https://doi.org/10.1016/j.disc.2018.10.011.

[7] Richard Breisch. An intuitive approach to speleotopology. Southwestern cavers, 6(5):72–78, 1967.

[8] John R. Britnell and Mark Wildon. Finding a princess in a palace: a pursuit-evasion problem. The Electronic Journal of Combinatorics, 20(1):25, 2013. doi:10.37236/2296.

[9] T. H. Chung, G. A. Hollinger, and V. Isler. Search and pursuit-evasion in mobile robotics: A survey. Autonomous Robots, 31(299):299–316, 2011.

[10] D. G. Corneil, Y. Perl, and L. K. Stewart. A linear recognition algorithm for cographs. SIAM Journal on Computing, 14(4):926–934, 1985.

[11] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Parameterized Algorithms. Springer Publishing Company, Incorporated, 1st edition, 2015.

[12] J. A. Ellis, I. H. Sudborough, and J. S. Turner. The vertex separation and search number of a graph. Information and Computation, 113(1):50–79, 1994.

[13] A. C. Giannopoulou, P. Hunter, and D. M. Thilikos. Lifo-search: A min-max theorem and a searching game for cycle-rank and treedepth. Discrete Applied Mathematics, 160(15):2089–2097, 2012.

[14] G. Gottlob, N. Leone, and F. Scarcello. Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width. Journal of Computer and System Sciences, 66(4):775–808, 2003.

[15] Vytautas Gruslys and Ares Méroueh. Catching a mouse on a tree. arXiv preprint arXiv:1502.06591, 2015.

[16] Jens Gustedt. On the pathwidth of chordal graphs. Discret. Appl. Math., 45(3):233–248, 1993. doi:10.1016/0166-218X(93)90012-D.

[17] P. L. Hammer and B. Simeone. The splittance of a graph. Combinatorica, 1:275–284, 1981.

[18] John Haslegrave. An evasion game on a graph. Discrete Mathematics, 314:1–5, 2014.

[19] David Ilcinkas, Nicolas Nisse, and David Soguet. The cost of monotonicity in distributed graph searching. Distributed Computing, 22(2):117–127, 2009. doi:10.1007/s00446-009-0089-1.

[20] A. Isaza, J. Lu, V. Bulitko, and R. Greiner. A cover-based approach to multi-agent moving target pursuit. In proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, pages 54–59. AAAI Press, 2008.

[21] T. Johnson, N. Robertson, P. D. Seymour, and R. Thomas. Directed tree-width. Journal of Combinatorial Theory Series B, 82(1):138–154, 2001.

[22] Andrea S. LaPaugh. Recontamination does not help to search a graph. Journal of the ACM, 40(2):224–245, 1993. doi:10.1145/151261.151263.

[23] Frédéric Mazoit and Nicolas Nisse. Monotonicity of non-deterministic graph searching. Theoretical Computer Science, 399(3):169–178, 2008. doi:10.1016/j.tcs.2008.02.036.

[24] Nicolas Nisse. Network decontamination. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, Distributed Computing by Mobile Entities, Current Research in Moving and Computing, volume 11340 of Lecture Notes in Computer Science, pages 516–548. Springer, 2019. doi:10.1007/978-3-030-11072-7_19.

[25] T. D. Parsons. Pursuit-evasion in a graph. In Theory and Applications of Graphs, LNCS, volume 642, pages 426–441. Springer, 1978.

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

[27] D. P. Williamson and D. B. Shmoys. The Design of Approximation Algorithms. Cambridge University Press, 2011.

[28] Boting Yang, Danny Dyer, and Brian Alspach. Sweeping graphs with large clique number. Discrete Mathematics, 309(18):5770–5780, 2009. doi:10.1016/j.disc.2008.05.033.

A Appendix

Proposition 6. *Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a hunter strategy in a bipartite graph $G = (V_r \cup V_w, E)$ with respect to V_r. Let $v \in V_r$ (resp. $v \in V_w$) and $1 \leq i \leq \ell$. If there exists a vertex $u \in N(v)$ and a vertex $x \in N(u)$ (possibly $x = v$) such that $u, x \notin \bigcup_{j \leq i} S_j$, then $v \in Z_{2p}$ for every $2p \leq i$ (resp. $v \in Z_{2p+1}$ for every $2p + 1 \leq i$).*

Proof. This clearly holds if $p = 0$ (i.e., when $v \in V_r$) since $Z_0 = V_r$. If $p = 1$ (i.e., when $v \in V_w$), there exists a rabbit trajectory $(r_0 = u \in N(v) \cap V_r \setminus S_1, r_1 = v)$ and so $v \in Z_1$. Hence, we assume that $p > 1$.

The rabbit can follow the following strategy depending on whether p is odd or even:

1. p is odd (and so $v \in V_w$): The rabbit can follow the following trajectory: $(r_0 = u, r_1 = x, \ldots, r_{p-1} = u, r_p = v)$ where, for $q < p$, $r_q = u$ if q is even and $r_q = x$ if q is odd.
2. p is even (and so, $v \in V_r$): The rabbit can follow the following trajectory: $(r_0 = x, r_1 = u, \ldots, r_{p-1} = u, r_p = v)$ where, for $q < p$, $r_q = x$ if q is even and $r_q = u$ if q is odd.

In both cases, for every $0 \leq j < p$, $r_j \notin S_{j+1}$ since $p \leq i$ and $x, u \notin \bigcup_{j<i} S_j$. Therefore, $v \in Z_p$. □

Lemma 18. *Let $G = (V_r \cup V_w, E)$ be a bipartite graph with at least two vertices. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a monotone hunter strategy in G with respect to V_r. For any $0 \leq p < i \leq \lceil \ell/2 \rceil$, $Z_{2i} \subseteq Z_{2p}$ and $Z_{2i+1} \subseteq Z_{2p+1}$.*

Proof. Let us recall that for any hunter strategy in G with respect to V_r, if q is even, then $Z_q \subseteq V_r$ and $Z_q \subseteq V_w$ otherwise. Thus , if $p = 0$ or if $i = 2$, $Z_i \subseteq Z_p$. Also, if $i = 1$, then $Z_i = Z_p$. Hence, let us assume that $p \geq 1$ and $i > 2$. Let $v \in Z_i$. Since $v \in Z_i$, there exists a rabbit trajectory $R = (r_0, \ldots, r_{i-2} = x, r_{i-1} = u, r_i = v)$ such that, for any $0 \leq j < i$, $r_j \notin S_{j+1}$. By definition of a rabbit trajectory, $u \in N(v)$ and $x \in N(u)$. Moreover, by monotonicity of $\mathcal{S}$, since $u \in Z_{i-1} \setminus S_i$ (resp. $x \in Z_{i-2} \setminus S_{i-1}$), $u \notin \bigcup_{q \leq i} S_q$ (resp. $x \notin \bigcup_{q \leq i-1} S_q$). By Proposition 6, $v \in Z_p$ for each $p \leq i$ such that p and i has the same parity. □

Lemma 21. *For any bipartite graph $G = (V_r \cup V_w, E)$ and any $k \geq mh_{V_r}(G)$, there exists a parsimonious monotone winning hunter strategy in G with respect to V_r and that uses k hunters.*

Proof. Let $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a monotone winning hunter strategy with respect to V_r using at most $k \geq mh_{V_r}(G)$ hunters such that ℓ is minimum. If $\mathcal{S}$ is parsimonious, we are done. Otherwise, among such strategies, Let us consider $\mathcal{S}$ that maximizes the first round $1 \leq j < \ell$ that makes $\mathcal{S}$ not parsimonious. There are several cases to be considered.

- Let $\mathcal{Z}(\mathcal{S}) = (Z_0(\mathcal{S}), \ldots, Z_\ell(\mathcal{S}))$ be the set of contaminated vertices for each round of $\mathcal{S}$. If there exists an integer $\ell' < \ell$ such that $Z_{\ell'}(\mathcal{S}) = \emptyset$, then $\mathcal{S} = (S_1, \ldots, S_{\ell'})$ is also a winning hunter strategy with respect to V_r using at most k hunters, contradicting the minimality of ℓ.

  Hence, we may assume that $Z_i(\mathcal{S}) \neq \emptyset$ for every $0 \leq i < \ell$.

- Let $1 \leq j \leq \ell$ be the smallest integer such that $S_j \setminus Z_{j-1}(\mathcal{S}) \neq \emptyset$ (if no such integer exists, then $\mathcal{S}$ is parsimonious and we are done). If $S_j \cap Z_{j-1}(\mathcal{S}) \neq \emptyset$, replace S_j by $S_j \cap Z_{j-1}(\mathcal{S})$. This leads to a winning monotone hunter strategy $\mathcal{S}'$ (indeed, $Z_h(\mathcal{S}) = Z_h(\mathcal{S}')$ for all $1 \leq h \leq \ell$) contradicting the maximality of j.

  Hence, we may assume that $S_j \cap Z_{j-1}(\mathcal{S}) = \emptyset$. Note that this implies that $j < \ell$ (since otherwise, $\mathcal{S}$ would not be winning).

- If any, let $0 < i$ be the minimum integer such that $S_{j+2i} \cap Z_{j+2i-1}(\mathcal{S}) \neq \emptyset$. Let $v \in S_{j+2i} \cap Z_{j+2i-1}(\mathcal{S})$. Since $b \in Z_{j+2i-1}(\mathcal{S})$, by Lemma 18, $v \in Z_{j-1+2i'}$ for every $0 \leq i' \leq i$.

  Then, for every $0 \leq i' < i$, replace $S_{j+2i'}$ by $\{v\}$. Let us prove that this leads to a monotone hunter strategy contradicting the maximality of j.

  Let $\mathcal{S}'$ be the strategy obtained by the above modifications. First, note that, for any $0 \leq h < j$, $S_h = S'_h$ and so $Z_h(\mathcal{S}) = Z_h(\mathcal{S}')$. By definition, $Z_j(\mathcal{S}) = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}) \setminus S_j \wedge (xy \in E)\}$ and, since $S_j \cap Z_{j-1} = \emptyset$, we get $Z_j(\mathcal{S}) = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}) \wedge (xy \in E)\}$. On the other hand, $Z_j(\mathcal{S}') = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}') \setminus S'_j \wedge (xy \in E)\} = \{x \in V \mid \exists y \in Z_{j-1}(\mathcal{S}) \setminus \{v\} \wedge (xy \in E)\}$ since $Z_{j-1}(\mathcal{S}') = Z_{j-1}(\mathcal{S})$ and $S'_j = \{v\}$. Hence, $Z_j(\mathcal{S}') \subseteq Z_j(\mathcal{S})$. By induction on $j \leq i' \leq \ell$ and using the same arguments, we get that $Z_{i'}(\mathcal{S}') \subseteq Z_{i'}(\mathcal{S})$ for every $j \leq i' \leq \ell$. Thus, $\mathcal{S}'$ is a winning hunter strategyin G with respect to $W \subseteq V_r$ using at most $k \geq mh_{V_r}(G)$ hunters (because $Z_\ell(\mathcal{S}') \subseteq Z_\ell(\mathcal{S}) = \emptyset$). It remains to show that $\mathcal{S}'$ is monotone.

  For purpose of contradiction, let us assume that $\mathcal{S}'$ is non-monotone. By Lemma 19, there exists a vertex x and $1 < m \leq \ell$ such that $x \in Z_{m-1}(\mathcal{S}') \setminus S'_m$ and $x \in \bigcup_{h<m} S'_h$.

  If $x \neq v$, then by definition of $\mathcal{S}'$ (for every $1 \leq r \leq \ell$, either $S'_r = S_r$ or $S'_r = \{v\}$) and because $Z_r(\mathcal{S}') \subseteq Z_r(\mathcal{S})$ for all $1 \leq r \leq \ell$, we get that $x \in \bigcup_{h<m} S_h$ and $x \in Z_{m-1}(\mathcal{S}) \setminus S_m$ which contradicts the monotonicity of $\mathcal{S}$. Hence, let us assume that $x = v$.

  Recall that we proven that $v \in Z_{j-1}(\mathcal{S}) \setminus S_j$. Therefore, by monotonicity of $\mathcal{S}$, $v \notin \bigcup_{r<j} S'_r$. Since $v \in S'_{j+2i'}$ for all $0 \leq i' \leq i$ and, by parity, $v \notin Z_{j+2i'}(\mathcal{S}')$ for all $0 \leq i' \leq i$, we get that $m > j + 2i$. This means that $v \in Z_{m-1}(\mathcal{S}) \setminus S_m$ (because $Z_r(\mathcal{S}') \subseteq Z_r(\mathcal{S})$ for all $1 \leq r \leq \ell$ and $S'_r = S_r$ for all $r \geq m > j + 2i$) and $v \in S_{j+2i}$, which contradicts the monotonicity of $\mathcal{S}$.

  Hence, we may assume that $S_{j+2i'} \cap Z_{j+2i'-1}(\mathcal{S}) = \emptyset$ for all $0 \leq i'$ such that $j + 2i' \leq \ell$.

- If $Z_{j+1}(\mathcal{S}) = Z_{j-1}(\mathcal{S})$, then remove S_j and S_{j+1} from $\mathcal{S}$. Let $\mathcal{S}'$ be the obtained strategy. We have that $Z_r(\mathcal{S}) = Z_r(\mathcal{S}')$ and $S_r = S'_r$ for all $r < j$ and $Z_{r+2}(\mathcal{S}) = Z_r(\mathcal{S}')$ and $S_{r+2} = S'_r$ for all $j \leq r \leq \ell - 2$. Hence, $\mathcal{S}'$ is winning since $Z_{\ell-2}(\mathcal{S}')$. Moreover, if $\mathcal{S}'$ is non-monotone, then, by Lemma 7, there exists x and $1 < m \leq \ell$ such that $x \in Z_{m-1}(\mathcal{S}') \setminus S'_m$ and $x \in \bigcup_{h<m} S'_h$. If $m < j$, this implies that $x \in Z_{m-1}(\mathcal{S}) \setminus S_m$ and $x \in \bigcup_{h<m} S_h$ contradicting the monotonicity of $\mathcal{S}$. Otherwise ($m \geq j$), $x \in Z_{m+1}(\mathcal{S}) \setminus S_{m+2}$ and $x \in \bigcup_{h<m+2} S_h$, also contradicting the monotonicity of $\mathcal{S}$.

  Hence, we may assume that $Z_{j+1}(\mathcal{S}) \neq Z_{j-1}(\mathcal{S})$.

- Note first that by Lemma 18, $Z_{j+1}(\mathcal{S}) \subseteq Z_{j-1}(\mathcal{S})$. Thus, $Z_{j+1}(\mathcal{S}) \setminus Z_{j-1}(\mathcal{S}) = \emptyset$. Hence, since $Z_{j+1}(\mathcal{S}) \neq Z_{j-1}(\mathcal{S})$, we get that $Z_{j-1}(\mathcal{S}) \setminus Z_{j+1}(\mathcal{S}) \neq \emptyset$.

  Hence, there exists $v \in Z_{j-1}(\mathcal{S}) \setminus Z_{j+1}(\mathcal{S})$. Let $\mathcal{S}'$ be obtained by replacing S_j by $\{v\}$. By arguments similar to the ones of the third item of this proof, we can prove that $\mathcal{S}'$ is a monotone hunter strategy contradicting the maximality of j.

This completes the proof. □

Lemma 22. *Let $G = (V_r \cup V_w, E)$ be a bipartite graph and $\mathcal{S} = (S_1, \ldots, S_\ell)$ be a parsimonious monotone winning hunter strategy with respect to V_r.*

- *If there exist $1 \leq i < j \leq \ell$ such that $v \in S_i \cap S_j$, then $v \in S_{i+2}$.*
- *If $v \in V_r$ (resp., $v \in V_w$) and there exists an odd (resp., even) integer $1 \leq i < \ell$ such that $v \notin Z_{i-1}$, then $v \notin S_j$ for every $j \geq i$.*

Proof. First, for purpose of contradiction, let us assume that $v \in S_i \cap S_j$ and $v \notin S_{i+2}$ with $i < j$. Note that, since G is bipartite, that $\mathcal{S}$ is a strategy with respect to V_r and $S_i \cap S_j \neq \emptyset$, i and j must have the same parity and so, $j \geq i + 4$. Since $\mathcal{S}$ is parsimonious, $v \in Z_{i-1} \cap Z_{j-1}$. Thus, by Lemma 18, $v \in Z_{i+1} \setminus S_{i+2}$ and $v \in S_i$, contradicting the monotonicity of $\mathcal{S}$. Hence, the first statement holds.

Let us now prove the second statement. Let us assume now that $v \in V_r$ (the case $v \in V_w$ can be handled similarly). If $v \notin Z_{i-1}$ for any odd $1 \leq i < \ell$, then $v \notin Z_{j-1}$ for any odd $i \leq j \leq \ell$ by Lemma 18. Therefore, since $\mathcal{S}$ is parsimonious, $v \notin S_j$ for any odd $i \leq j \leq \ell$. Moreover, since G is bipartite and that $\mathcal{S}$ is a parsimonious strategy with respect to V_r, for any even $1 \leq j \leq \ell$, $v \notin S_j$.

□