SEARCHING FOR A VISIBLE, LAZY FUGITIVE*

DAVID RICHERBY† AND DIMITRIOS M. THILIKOS‡

Abstract. Graph searching problems are described as games played on graphs, between a set of cops and a fugitive. Variants of the game restrict the abilities of the cops and the fugitive and the corresponding search numbers (the least number of cops that have a winning strategy) are related to several well-known parameters in graph theory. We study the case where the fugitive is visible (the cops' strategy can take into account his current position) and lazy (he moves only when the cops move to his position). Our results are stated and proven in a general setting where the fugitive's speed (i.e., the lengths of paths he can move along) can be unbounded or bounded by some constant. We give a min-max characterization of the corresponding parameters, which we show to be computable in polynomial time for fugitives with unbounded speed and speed at most 3 and to be NP-complete for all other finite speeds. This is in contrast to the other standard versions of the game, where the parameters corresponding to fugitives with unbounded speed are NP-complete. Several consequences of our results are also discussed.

Key words. Graph Searching, Pursuit evasion games, Obstructions

AMS subject classifications. 05C57, 05C85, 05C83, 68R10

1. Introduction. Graph searching games are played between a group of cops and a fugitive, on the vertices and edges of a graph. The cops aim at capturing the fugitive, while the fugitive tries to escape capture. The rules by which the players move lead to several variants of the game. While the definition and study of such games dates back to the late 1970s [22, 23], they have recently been studied widely, mainly due to numerous applications in security problems in networks [1, 4, 12, 13].

There are several basic variants of the game and we consider only those where the cops and the fugitive reside on the vertices of the graph. At any one time, the fugitive occupies some vertex of the graph but each cop, independently, may be either on a vertex of the graph or out of play. In *node search* games, the cops are moved either by placing them on or removing them from vertices; in the more general setting of *mixed search*, a cop may, in addition, slide along an edge from the endpoint he occupies to the other, vacant, endpoint. In both variants, the fugitive moves along cop-free paths in the graph. The fugitive is captured if a cop moves to the vertex he occupies and he has no path along which to escape. If the fugitive is captured, the cops win; if he remains on the run forever, he wins. (We do not consider edge search, where the fugitive resides on edges of the graph, as this can be reduced to mixed search by standard techniques [5, 29].)

Further variants of the game come from altering the properties of the fugitive. He may be either *visible* to the cops, in which case the cops may use the fugitive's current position to choose their moves, or *invisible*, in which case the cops do not know where he is and their moves may be specified in advance. He may also be *lazy*,

---

*A previous version of this paper appeared as: David Richerby and Dimitrios M. Thilikos, Searching for a visible, lazy fugitive, 34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008), Lecture Notes in Computer Science, 2008, Volume 5344/2008, 348-359.

†School of Computing, University of Leeds, Leeds LS2 9JT, United Kingdom. E-mail: `richerby@comp.leeds.ac.uk`. Research carried out while visiting the Graduate Program in Logic, Algorithms and Computation ($\mu\pi\lambda\forall$), at the Department of Mathematics, National and Kapodistrian University of Athens, supported by the European Social Fund and Greek National Resources — (ΕΠΕΑΕΚ II) PYTHAGORAS II.

‡Department of Mathematics, National and Kapodistrian University of Athens, Panepistimioupolis, GR157-84 Athens, Greece. E-mail: `sedthilk@math.uoa.gr`.

node search	Visible	Invisible
Lazy	$\delta^s + 1$, in P for $s \in \{1, 2, 3, \infty\}$, NP-complete for $4 \leqslant s < \infty$ [this paper]	$\mathbf{tw} + 1$ [9], NP-c [2]
Active	$\mathbf{tw} + 1$ [26], NP-c [2]	$\mathbf{pw} + 1$ [5,17], NP-c [2]

TABLE 1.1

Graph node-search variants (for fugitives with unbounded speed), their corresponding graph parameters and their complexities.

in which case he moves only when a cop moves to his vertex, or *active*, in which case he may move at every round of the game.

Each variant of the game generates a graph parameter that is the minimum number of cops that have a winning strategy in a given graph. For the visible–active and invisible–lazy cases, the node search number is known to be one greater than the treewidth of the graph; for the invisible–active case, it is one greater than the graph's pathwidth. Similar parameters can be defined for mixed search [5, 24, 27, 28]. The decision problems associated with these graph parameters are known to be NP-complete.

In this paper, we study the remaining case, where the fugitive is visible and lazy, which does not seem to have been considered before. Generalizing, we parameterize the game by the speed s of the fugitive, i.e., the maximum length of paths along which he may move. We write, respectively, $\mathbf{vlns}^s$ and $\mathbf{vlms}^s$ for the node- and mixed-search numbers for a fugitive with speed s, with $s = \infty$ denoting a fugitive with unbounded speed. Our main result is a min-max theorem for the two parameters, for any speed $s \in \mathbb{N} \cup \{\infty\}$. In particular, we give a characterization in terms of the existence of specific obstructing structures, which we call *hide-outs*, that guarantee an escape strategy for the fugitive.

We also introduce two hierarchies of graph parameters, defined in terms of layouts, which we write δ^s and δ^s_{m} (defined in Section 4). These parameters are equivalent, respectively, to $\mathbf{vlns}^s$ and $\mathbf{vlms}^s$. The min-max theorem implies that our search parameters, in the case of a fugitive with unbounded speed, can be computed in polynomial time, which is quite unexpected, since all other variants of the game discussed above lead to NP-complete parameters. The parameters can also be computed in polynomial time for fugitives with speed at most 3; for other finite speeds, they are NP-complete. The known and new results for fugitives of unbounded speed are summarized in Table 1.1.

δ^s is a natural generalization of the classical graph parameter of *degeneracy*, defined as

$$\delta^*(G) = \max\{\delta(H) \mid H \subseteq G\},$$

where $\delta(H)$ is the minimum degree of H's vertices. It is known from folklore that $\delta^*(G) = \delta^1(G)$ [7,18,21] and $\delta^*(G)+1$ is also known as the graph's *colouring number*, since there is an easy greedy algorithm that colours a graph G with that many colours [10].

We prove that each of $\mathbf{vlns}^s$ and $\mathbf{vlms}^s$ defines a nontrivial hierarchy of parameters: for any r and s with $3 \leqslant r < s < \infty$, there are graphs with

$$\mathbf{vlns}^r(G) < \mathbf{vlns}^s(G) < \mathbf{vlns}^\infty(G)$$

and similarly for mixed search.

To give a lower bound for treewidth, Bodlaender, Koster and Wolle define the *contraction degeneracy* of a graph G to be $\delta\mathrm{C}(G)$, the maximum $\delta(H)$ over non-trivial minors H of G [7,19,30]. We extend contraction degeneracy by replacing the term $\delta(H)$ with $\delta^s(H)$ and show that the extension $\delta^\infty\mathrm{C}(G)$, where $\delta(H)$ is replaced with $\delta^\infty(H)$, approximates treewidth, in the sense that there is a function f such that, for all graphs, $\delta^\infty\mathrm{C}(G) \leqslant \mathbf{tw}(G) \leqslant f(\delta^\infty\mathrm{C}(G))$. This improves on contraction degeneracy, which is known to provide only a lower bound for treewidth.

The remainder of the paper is organized as follows. Section 2 gives basic definitions. The searching model for a visible, lazy fugitive is formally described in Section 3. In Section 4, we define hide-outs and our generalization of graph degeneracy. Our main results appears in 5 where we prove our min-max characterisation. In Section 6 we give the algorithmic and complexity results on our parameters. The nontriviality of the hierarchies defined is shown in Section 7. In Section 8, we generalize contraction degeneracy. Finally, we make concluding remarks and present some open problems in Section 9.

2. Preliminaries. We write $\mathbb{N}$ for the set $\{1,2,\dots\}$ and $\mathbb{N}^+$ for $\mathbb{N}\cup\{\infty\}$. Given a set S and an object x, we write $S+x$ and $S-x$ for $S\cup\{x\}$ and $S\setminus\{x\}$, respectively.

All graphs considered in this paper are finite, simple and undirected. To avoid trivial exceptions, we assume that all graphs contain at least one edge.

We write $V(G)$ and $E(G)$, respectively, for the vertex set and edge set of a graph G and xy for the undirected edge $\{x,y\}$. For $X \subseteq V(G)$, $G[X]$ is the subgraph of G induced by the vertices in X and, for $Y \subseteq E(G)$, $G - Y = (V(G), E(G) \setminus Y)$. Given a vertex $x \in V(G)$, we let $E_G(x)$ be the set of all edges of G incident with x. Also we denote by $N_G[x]$ the *closed neighborhood* of x in G that contains x together with all vertices of G that are adjacent to x. If $x \in V(G)$ and $e \in E(G)$, we write $G - x$ for the graph $G[V(G)-x]$ and $G-e$ for the graph $(V(G), E(G)-e)$ respectively. We use the notation $\delta(G)$ and $\Delta(G)$ for the minimum and maximum degree of G's vertices, respectively. We also use the notation $H \subseteq G$ to denote the fact that a graph H is a subgraph of a graph G. Finally, we denote by K_d the complete graph with d vertices.

The operation of *dissolving* a vertex $x \in V(G)$ of degree two is the removal of x from G and the addition of an edge connecting its two former neighbours. A graph H is a *topological minor* of G if it can be made from some subgraph of G by dissolving vertices of degree two. A graph H is a *minor* of G if it can be made from a subgraph of G by contracting edges (i.e., identifying the two endpoints of the edge and deleting the resulting loop).

A *tree decomposition* of a graph G is a pair (T, X) where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a collection of subsets of $V(G)$ such that:

- $\bigcup_{i\in V(T)} X_i = V(G)$;
- for each edge $xy \in E(G)$, $\{x,y\} \subseteq X_i$ for some $i \in V(T)$; and
- for each $x \in V(G)$, $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is defined to be

$$\max\{|X_i| - 1 \mid i \in V(T)\}$$

and the *treewidth* of a graph G is the minimum width over all tree decompositions of G. If we restrict the tree T to be a path, then we define the notions of *path decomposition* and *pathwidth.* We write $\mathbf{tw}(G)$ and $\mathbf{pw}(G)$, respectively, for the treewidth and pathwidth of a graph G.

3. The searching model. In this section, we define a model for the graph search game against a visible, lazy fugitive. The players have complete information about each other's position and may use this to decide their next move. The cops' goal is to capture the fugitive who tries, of course, to evade capture. Initially, there are no cops in the graph but, at any moment before his capture, the fugitive is on some vertex. The fugitive is *lazy*, in that he may move only when a cop is moved to his current vertex. When he moves, he does so with speed $s \in \mathbb{N}^+$: that is, he moves along a cop-free path of length at most s.

A play of the game consists of a sequence of rounds, with each round being composed of three parts, as follows.

Announcement. The cops announce their intended move to the fugitive. This can be: the *placement* of a cop on a vertex x, not currently occupied by a cop; the *removal* of a cop from an occupied vertex x; or *sliding* a cop from one end x of an edge xy to the other, which is initially not occupied by a cop.

Avoidance. If a cop is to be placed on or slid to the fugitive's current vertex, the latter may move along any cop-free path of length at most s. In the case of placement to x, that vertex is not considered blocked at this round; for sliding from x to y, the edge xy is considered blocked but the vertices x and y are not.

Realization. The cops carry out the announced action.

The fugitive is captured if a cop moves to his vertex and he has no move to escape. We may assume that the fugitive has full knowledge of the cops' strategy and will take the optimal decision towards avoiding capture. The fugitive is visible, so the cops' moves take his position into account and the game is interactive.

We denote the position of the fugitive in the graph at the ith round by a vertex $v_i \in V(G)$. Since, at any time, there is at most one cop on each vertex, we may represent the position of the cops after the ith move as a set $S_i \subseteq V(G)$.

We say that a finite or infinite sequence $S_0, S_1, \ldots$ of subsets of $V(G)$ is *consistent* if, for all $i \geqslant 0$, $G[S_i \vartriangle S_{i+1}]$ is either a single vertex or a two-clique (we use symbol $\vartriangle$ for the symmetric difference of two sets). Thus, the sequence corresponds to a sequence of cop positions in a play of the game and either $S_{i+1} = S_i + x$ for some $x \notin S_i$ (placement to x), $S_{i+1} = S_i - x$ for some $x \in S_i$ (removal from x) or $S_i \vartriangle S_{i+1}$ is an edge of G (sliding from the unique vertex in $S_i \setminus S_{i+1}$ to the unique vertex in $S_{i+1} \setminus S_i$).

Given two consecutive sets S and S' of a consistent sequence, we say that a path P in G is (S, S')-*avoiding* if its internal vertices avoid $S \cap S'$ and its edges avoid the edge $e = S \vartriangle S'$, in the case that $|e| = 2$.

Let $k \in \mathbb{N}$ and $s \in \mathbb{N}^+$. A (k, s)-*play* of the game on a graph G is a finite or countably infinite sequence of alternating vertex sets and vertices

$$\langle S_i, v_i \mid 0 \leqslant i < \kappa \rangle$$

for some $\kappa \in \mathbb{N}^+$, such that:

- $S_0 = \emptyset$;
- the sequence S_0, $S_1, \ldots$ is consistent;
- $|S_i| \leqslant k$ for all i; and
- for each i with $0 < i < \kappa$, either
  - $v_{i-1} \notin S_i$ and $v_i = v_{i-1}$ (the cops did not move to the fugitive's vertex so he did not move);

– $v_{i-1} \in S_i$, there is an (S_{i-1}, S_i)-avoiding path of length at most s from v_{i-1} to v_i and $v_i \notin S_i$ (the cops moved to the fugitive's vertex and he ran along a cop-free path of length at most s); or
– $i = \kappa - 1$, $v_i = v_{i-1} \in S_i$ and there are no (S_{i-1}, S_i)-avoiding paths from v_i (we are at the last move of a finite play, a cop has moved to the fugitive's vertex and he has no cop-free path on which to escape: the fugitive has been captured).

Each move made by the cops may depend both on their current position and that of the fugitive. A (k, s)-*strategy* is a function

$$\mu\colon V(G)^{[\leqslant k]} \times V(G) \to V(G)^{[\leqslant k]},$$

whose inputs are the position S of the cops and the position v of the fugitive and whose output is S', the new position of the cops, which is reached by a single placement, removal or sliding move and with $|S'| \leqslant k$. The strategy is to be used against a fugitive who has speed $s \in \mathbb{N}^+$.

Note that, when we define strategies, we will not define the action of the cops in positions that can never occur when the strategy is executed. Thus, we give only a partial function. Formally, the strategy is any total extension of this partial function, assigning arbitrary moves to the cops in situations that do not occur in any play.

Given a (k, s)-strategy μ, a μ-*play* is any (k, s)-play $\langle S_i, v_i \mid 0 \leqslant i < \kappa\rangle$ where $S_{i+1} = \mu(S_i, v_i)$ for all i. A (k, s)-strategy μ is said to be *winning* for the cops against a fugitive with speed s if every μ-play is finite (i.e., results in the capture of the fugitive).

We define the *visible, lazy mixed-search number* against a fugitive with speed s for a graph G to be

$$\mathbf{vlms}^s(G) = \min\{k \mid \text{there is a winning } (k, s)\text{-strategy for } G\}.$$

Recall that we demanded the symmetric difference of consecutive cop positions S_i and S_{i+1} to be either a singleton or a set of two adjacent vertices $x \in S_i$ and $y \in S_{i+1}$. The latter case reflects the sliding move of a searcher along the edge $e = \{x, y\}$, from x to y and assumes that the vertex y is *not occupied* by a searcher before sliding. Actually, it is possible to drop this assumption by allowing more than one searcher to occupy a vertex. Certainly, this requires the definition of a more general model. However, it would not make any real difference: sliding a searcher along e from x to y, while y is occupied is equivalent to the removal of this searcher from x. Moreover, the case where x is occupied by two searchers and one of them slides along e from x to y is the same as having this searcher out of the graph and placing him on y.

By applying the same definitions but now demanding that the symmetric difference of consecutive cop positions is always a singleton (i.e., only placement and removal moves are allowed), we define the analogous *visible, lazy node search number* against a fugitive with speed s for a graph G, which we denote $\mathbf{vlns}^s(G)$.

The following lemma gives key properties of the defined parameters.

LEMMA 3.1. *For any graphs G and H and any $s \in \mathbb{N}^+$,*
1. $\delta(G) + 1 \leqslant \mathbf{vlns}^s(G) \leqslant \Delta(G) + 1$ *and* $\delta(G) \leqslant \mathbf{vlms}^s(G) \leqslant \Delta(G)$;
2. *if* $H \subseteq G$, *then* $\mathbf{vlns}^s(H) \leqslant \mathbf{vlns}^s(G)$ *and* $\mathbf{vlms}^s(H) \leqslant \mathbf{vlms}^s(G)$;

Proof. To capture the fugitive on any vertex x, a cop must be placed on every one of the at least $\delta(G)$ vertices adjacent to x and either a further cop placed on x (node search) or one of those cops slid onto x (mixed search). The fugitive can always be caught on any vertex x by placing cops on the at most $\Delta(G)$ neighbours of x and

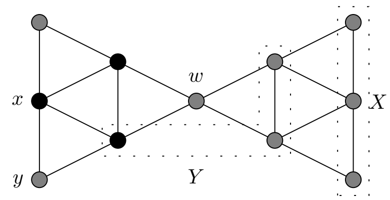

FIGURE 4.1. *An example graph with $\delta^s(G) = 2$ for any $s \in \mathbb{N}^+$.*

proceeding as before. This completes the proof of (1). For (2), notice that if there are fewer than $\mathbf{vlns}^s(H)$ cops (respectively, $\mathbf{vlms}^s(H)$ cops) then the fugitive can avoid capture in the node-search game (respectively, mixed-search game) on G forever by staying within H. ◻

Variants of the above model have already appeared in the literature, for fugitives of unbounded speed. The version where the fugitive is visible and active is due to Seymour and Thomas [26], who show that the corresponding node-search number is $\mathbf{tw}(G) + 1$; the node-search number is the same for an invisible, lazy fugitive [9]. Finally, the version with an invisible, active fugitive was introduced by Kirousis and Papadimitriou [17] and studied further by Bienstock and Seymour [5]. In this case, the node-search number is $\mathbf{pw}(G) + 1$ [16, 17]. It follows immediately that determining any of the above search numbers is NP-complete and the same can also be shown for the mixed search variants of all of these games. However, we prove that the parameters $\mathbf{vlns}^\infty(G)$ and $\mathbf{vlms}^\infty(G)$ are polynomial-time computable. These results are summarized in Table 1.1.

4. Degeneracy and hide-outs. Let G be a graph, $x \in V(G)$ and $X \subseteq V(G) - x$. For any $s \in \mathbb{N}^+$, we say that a set $A \subseteq V(G) - x$ is an *(s, x, X)-separator* if $G - A$ contains no path from x to X of length at most s. Define $\mathbf{sep}^s_G(x, X)$ to be the minimum size of any (s, x, X)-separator in G. For example, for the graph G in Figure 4.1, $\mathbf{sep}^\infty_G(x, X) = 1$, as w is a cut vertex. However, $\mathbf{sep}^s_G(x, X) = 0$ for any $s < 4$. Moreover, $\mathbf{sep}^3_G(x, Y) = 2$ and $\mathbf{sep}^2_G(x, Y) = \mathbf{sep}^1_G(x, Y) = 1$.

For $s \leqslant 3$ and $s = \infty$, $\mathbf{sep}^s_G(x, X)$ is the maximum cardinality of any set of x–X paths of length at most s that are vertex-disjoint apart from the common endpoint x. This is immediate from Menger's theorem in the case $s = \infty$ and can be shown by a simple modification to the proof of Lovász et al. [20, Theorem 3] for $s \leqslant 3$. On the other hand, for finite $s \geqslant 4$, there are graphs where $\mathbf{sep}^s_G(x, X)$ is greater than the maximum number of disjoint s-paths from x to X [20].

LEMMA 4.1. *Let $s \in \mathbb{N}^+$. Given a graph G, $x \in V(G)$, $X \subseteq V(G) - x$ and $k \in \mathbb{N}$, the problem of determining whether $\mathbf{sep}^s_G(x, X) \leqslant k$ can be solved in polynomial time for $s \in \{1, 2, 3, \infty\}$ and is NP-complete for all other s.*

Proof. The case $s = \infty$ is immediate from the max-flow min-cut theorem and the existence of polynomial-time algorithms for computing maximal flows. For finite values of s, the result follows from [15] and [20]. ◻ In fact, for any finite $s \geqslant 4$, it is NP-hard even to approximate $\mathbf{sep}^s_G(x, X)$ within a constant factor of 1.1377 [3].

Let G be a graph, $k \in \mathbb{N}$ and $s \in \mathbb{N}^+$. A *(k, s)-layout* of G is an ordering $v_1, \dots, v_n$ of $V(G)$ such that, for every $i \in \{1, \dots, n\}$, $\mathbf{sep}^s_G(v_i, \{v_1, \dots, v_{i-1}\}) \leqslant k$. We define the *$s$-degeneracy* of G to be $\delta^s(G)$, the least k for which G has a (k, s)-layout. Notice that the 1-degeneracy is identical to the classical graph-theoretic parameter of

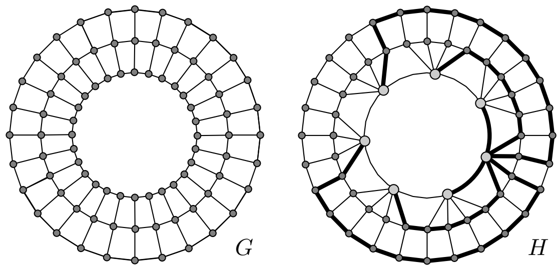

FIGURE 4.2. *The graph H is a minor of G, however $\mathbf{vlns}^\infty(G) = 5$, while $\mathbf{vlns}^\infty(H) = 7$. Notice that the vertices of degree 4 in G form a $(4, \infty)$-hide-out. The vertices of degree six in H form a $(6, \infty)$-hide-out, as shown by the bold paths.*

degeneracy.

A (k, s)-*hide-out* in a graph G is any set $R \subseteq V(G)$ such that, for every $x \in R$, $\mathbf{sep}^s_G(x, R - x) \geqslant k$. In Figure 4.2, the set of vertices in H of degree six is a $(6, \infty)$-hide-out; one of the sets of paths is shown in bold.

The s-degeneracy of a graph and the presence or absence of (k, s)-hide-outs within it are closely linked with the node search number for a visible, lazy fugitive. We now adapt these two concepts for mixed-search. First, set

$$\mathbf{msep}^s_G(x, X) = \min \{\mathbf{sep}^s_{G-Y}(x, X) \mid Y \subseteq E_G(x) \text{ and } |Y| \leqslant 1\}.$$

That is, Y can be either empty, or a singleton containing one edge incident with x. The following Lemma is immediate from the definitions.

LEMMA 4.2. *For any graph G and any $x \in V(G)$, $X \subseteq V(G) - x$ and $s \geqslant 2$,*

$$\mathbf{msep}^1_G(x, X) + 1 = \mathbf{sep}^1_G(x, X) = d_{G[X+x]}(x)$$
$$\mathbf{msep}^s_G(x, X) \leqslant \mathbf{sep}^s_G(x, X) \leqslant \mathbf{msep}^s_G(x, X) + 1.$$

For example, in graph G of Figure 4.1, we have $\mathbf{sep}^3_G(x, Y) = \mathbf{msep}^3_G(x, Y) = 2$, while $\mathbf{sep}^3_G(y, Y) = 2 = \mathbf{msep}^3_G(y, Y) + 1$.

LEMMA 4.3. *Let $s \in \mathbb{N}^+$. Given a graph G, $x \in V(G)$, $X \subseteq V(G) - x$ and $k \in \mathbb{N}$, the problem of determining whether $\mathbf{msep}^s_G(x, X) \leqslant k$ is in polynomial time for $s \in \{1, 2, 3, \infty\}$ and is* NP*-complete for all other s.*

Proof. The polynomial part follows directly from Lemma 4.1 and the definition of $\mathbf{msep}^s_G$. For the NP-completeness part, construct a graph $G' = G \cup P$, where P is a path of length s from x to some vertex in X, disjoint from G apart from its endpoints. We have $\mathbf{msep}^s_{G'}(x, X) = \mathbf{sep}^s_G(x, X)$ and the result is then immediate from Lemma 4.1. ◻

Let G be a graph, $k \in \mathbb{N}$ and $s \in \mathbb{N}^+$. A *mixed (k, s)-layout* is an ordering $v_1, \ldots, v_n$ of $V(G)$ such that $\mathbf{msep}^s_G(v_i, \{v_1, \ldots, v_{i-1}\}) \leqslant k$ for every $i \in \{1, \ldots, n\}$. Let the *mixed s-degeneracy* of G be $\delta^s_{\mathrm{m}}(G)$, the least k for which G has a mixed (k, s)-layout. In Figure 4.1, $\delta^1_{\mathrm{m}}(G) = 1$ and $\delta^s_{\mathrm{m}}(G) = 2$ for $s \geqslant 2$. Notice that $\delta^1(G) = \delta^1_m(G) + 1$, by Lemma 4.2.

We define a *mixed (k, s)-hide-out* in a graph G to be any set $R \subseteq V(G)$ such that, for every $x \in R$, $\mathbf{msep}^s_G(x, R - x) \geqslant k$. In the graph G of Figure 4.1, the black vertices form a mixed $(2, s)$-hide-out for any $s \geqslant 2$.

5. Min-max theorems. We are now ready to give our min-max characterizations for both $\mathbf{vlns}^s(G)$ and $\mathbf{vlms}^s(G)$ for all $s \in \mathbb{N}^+$.

THEOREM 5.1. *For any graph G and any $s \in \mathbb{N}^+$,*

$$\mathbf{vlns}^s(G) - 1 = \delta^s(G) = \max\{k \mid G \textit{ contains a } (k,s)\textit{-hide-out}\}$$
$$\mathbf{vlms}^s(G) - 1 = \delta^s_{\mathrm{m}}(G) = \max\{k \mid G \textit{ contains a mixed } (k,s)\textit{-hide-out}\}\,.$$

Proof. We give a full proof of the more complicated mixed-search case and indicate what needs to be changed to give a proof of the simpler node-search case. It suffices to show that, for any graph $G = (V, E)$, any $k \in \mathbb{N}$ and any $s \in \mathbb{N}^+$, the following are equivalent:

1. $\delta^s_{\mathrm{m}}(G) \leqslant k$ (resp., $\delta^s(G) \leqslant k$),
2. $\mathbf{vlms}^s(G) \leqslant k+1$ (resp., $\mathbf{vlns}^s(G) \leqslant k+1$) and
3. G contains no mixed $(k+1, s)$-hide-out (resp., no $(k+1, s)$-hide-out).

(1) $\Rightarrow$ (2). Let $v_1, \dots, v_n$ be a mixed (k, s)-layout of G and, for $i \in \{1, \dots, n\}$, let $X_i = \{v_1, \dots, v_{i-1}\}$. By the definition of **msep**, for each $i \in \{1, \dots, n\}$, there is a $Y_i \subseteq E_G(v_i)$, of size at most one, such that $\mathbf{sep}^s_{G-Y_i}(v_i, X_i) \leqslant k$.

By construction, $G - Y_i$ contains an (s, v_i, X_i)-separator S_i of size at most k. By definition, any v_i–X_i path in G of length at most s must meet S_i or use the edge Y_i if it exists. Let $Z_i = S_i$ if $Y_i = \emptyset$ or $S_i + x$ if $Y_i = \{v_i x\}$.

We can now define a winning $(k+1, s)$-strategy μ for G. Suppose that, at some point in the game, the cops are at position Z and the fugitive is on vertex v_i. The strategy proceeds by first removing the cops one at a time from $Z \setminus Z_i$ and then placing cops one at a time on the vacant vertices of Z_i. Finally, a cop is either placed on the fugitive's vertex v_i, if $Y_i = \emptyset$, or slid there, otherwise. Formally, let $z_1, \dots, z_\alpha$ enumerate $Z \setminus Z_i$ and let $z'_1, \dots, z'_\beta$ enumerate $Z_i \setminus Z$, with both being sub-sequences of $v_1, \dots, v_n$. For $j \in \{1, \dots, \alpha\}$, we set

$$\mu(Z \setminus \{z_1, \dots, z_{j-1}\}, v_i) = Z \setminus \{z_1, \dots, z_j\}$$

and, for $j \in \{1, \dots, \beta\}$, we set

$$\mu(Z_i \setminus \{z'_1, \dots, z'_j\}, v_i) = Z_i \setminus \{z'_1, \dots, z'_{j-1}\}\,.$$

Finally, if $Y_i = \emptyset$ then $|Z_i| \leqslant k$ so we can set $\mu(Z_i, v_i) = Z_i + v_i$. Otherwise, it is possible that $|Z_i| = k+1$ but, in that case, we can set $\mu(Z_i, v_i) = Z_i \vartriangle Y_i$.

When the cops move to the fugitive's vertex v_i, there is a cop on every vertex of S_i and, if $Y_i = \{e\}$, there is a cop sliding along edge e. Since every path of length at most s from v_i to X_i passes through S_i or e, the fugitive, if he can move at all, can only move to some vertex v_j with $j > i$. Since there are only n vertices in G and the cops force the fugitive to move at least once every $2k+3$ rounds (at most $k+1$ removals, $k+1$ placements and one slide), the fugitive will be caught in at most $n(2k+3)$ rounds.

In the case of node search, $Y_i = \emptyset$ for all $i \in \{1, \dots, n\}$ so there are no sliding moves to consider.

(2) $\Rightarrow$ (3). Let R be a mixed $(k+1, s)$-hide-out in G. The fugitive can avoid capture forever, against any $(k+1, s)$-strategy of the cops by using the following escape strategy.

- Initially, the fugitive occupies any vertex $v \in R$.

- Suppose a cop is placed on the fugitive's current vertex v and let $S \subseteq V(G)$ be the position of the cops after this placement. We have $|S - v| \leqslant k$. $\mathbf{sep}^s_G(v, R-v) \geqslant \mathbf{msep}^s_G(v, R-v) \geqslant k+1$, so $G-(S-v)$ contains an s-path from v to $R-v$ and the fugitive can run to some other vertex in the hide-out.
- In the mixed-search case, suppose a cop slides from some vertex u to the fugitive's current vertex v. Since $\mathbf{msep}_G(v, R-v) \geqslant k+1$, we have $\mathbf{sep}^s_{G-uv}(v, R-v) \geqslant k+1$ so the fugitive escapes as before.

(3) $\Rightarrow$ (1). Assuming that G does not contain a mixed $(k+1, s)$-hide-out, we inductively construct a mixed (k,s)-layout $v_1, \dots, v_n$ of G. Since $V(G)$ is not a mixed $(k+1, s)$-hide-out, there is a vertex v with $\mathbf{msep}^s_G(v, V(G)-v) \leqslant k$. Let $v_n = v$. Suppose we have chosen $v_\ell, \dots, v_n$ for some $\ell \in \{2, \dots, n\}$. $V(G) \setminus \{v_\ell, \dots, v_n\}$ is not a mixed $(k+1, s)$-hide-out so, as before, we may choose $v_{\ell-1}$ to be some vertex v with $\mathbf{msep}^s_G(v, V(G) \setminus \{v, v_\ell, \dots v_n\}) \leqslant k$. For the case of node search, replace "$\mathbf{msep}^s_G$" with "$\mathbf{sep}^s_G$" and mixed hide-outs with ordinary hide-outs. ∎

We remark that the strategies we have considered only allow the cops to choose their next move based on the current position in the game and not, for example, on the history of the play so far.

The following lemma gives some consequences of Theorem 5.1.

LEMMA 5.2. *Let G and H be graphs and let $s \in \mathbb{N}^+$.*

1. $\mathbf{vlms}^s(G) \leqslant \mathbf{vlns}^s(G) \leqslant \mathbf{vlms}^s(G) + 1$.
2. $\mathbf{vlns}^1(G) = \mathbf{vlms}^1(G) + 1 = \delta^*(G) + 1$.
3. *If H is a topological minor of G, then* $\mathbf{vlns}^\infty(H) \leqslant \mathbf{vlns}^\infty(G)$ *and* $\mathbf{vlms}^\infty(H) \leqslant \mathbf{vlms}^\infty(G)$.

Proof. By Lemma 4.2 and the definitions of δ and δ_m, we obtain that $\delta^1(G) = \delta^1_m(G) + 1$ and $\delta^s_m(G) \leqslant \delta^s(G) \leqslant \delta^s_m(G) + 1$. Then parts 1 and 2 follow directly from Theorem 5.1.

For part 3, let H be obtained from $H' \subseteq G$ by dissolving vertices of degree 2. Observe that every (k, ∞)-hide-out (mixed (k, ∞)-hide-out) in H is also a (k, ∞)-hide-out (mixed (k, ∞)-hide-out) in H'. Then the result follows from Theorem 5.1. ∎

Thus, from Lemma 3.1.2 and Lemma 5.2.3, $\mathbf{vlns}^\infty$ and $\mathbf{vlms}^\infty$ are closed under taking subgraphs and topological minors. However, they are not closed under taking minors, since every graph G is a minor of some graph H with $\Delta(H) \leqslant 3$. G may have arbitarily large search numbers but, by Lemma 3.1.1, $\mathbf{vlns}^\infty(H) \leqslant 4$ and $\mathbf{vlms}^\infty(H) \leqslant 3$.

Lemma 5.2.3 cannot be extended to $\mathbf{vlns}^s$ or $\mathbf{vlms}^s$ for finite s. For $s \in \mathbb{N}$, there are graphs G with topological minors H such that $\mathbf{vlns}^s(H) > \mathbf{vlns}^s(G) = 3$ or $\mathbf{vlms}^s(H) > \mathbf{vlms}^s(G) = 2$: for example, take H to be any graph and replace the edges with independent $(s+1)$-paths to make G. (See also Figure 4.2.)

6. Algorithms and complexity. The proof of Theorem 5.1 also indicates that Algorithm 1 can be used to compute $\delta^s(G)$ for any $s \in \mathbb{N}^+$. The algorithm attempts to construct a (k,s)-layout of G greedily, which would show that $\delta^s(G) \leqslant k$. If this fails, a (k,s)-hide-out of G has been detected and, since the hide-out itself has no (k,s)-layout, nor does G. A straightforward modification of the algorithm, replacing $\mathbf{sep}^s_G$ with $\mathbf{msep}^s_G$, determines whether $\delta^s_{\mathrm{m}}(G) \leqslant k$, giving either a mixed (k,s)-layout or a mixed (k,s)-hide-out. By Lemmata 4.1 and 4.3, both variants of the algorithm run in polynomial time for $s = \infty$ and $s \leqslant 3$.

THEOREM 6.1. *Let $s \in \mathbb{N}^+$. Given a graph G and $k \in \mathbb{N}$, the problem of determining whether* $\mathbf{vlns}^s(G) \leqslant k$ *(respectively,* $\mathbf{vlms}^s(G) \leqslant k$*) is computable in time*

Algorithm 1 check-s-degen: check whether $\delta^s(G) \leqslant k$.

Input: an n-vertex graph G and an integer $k \geqslant 1$.
Output: if $\delta^s(G) \leqslant k$, a (k,s)-layout; if not, a (k,s)-hide-out.

$S \leftarrow V(G)$.
for $i = n, \ldots, 1$,
  if there is $x \in S$ with $\mathbf{sep}^s_G(x, S - x) \leqslant k$ then $v_i \leftarrow x$;
  else output "$\delta^s(G) > k$, witnessed by hide-out S."
  $S \leftarrow S - v_i$.
Output "$\delta^s(G) \leqslant k$, witnessed by layout $v_1, \ldots, v_n$."

polynomial in $|V(G)|$ *if* $s \in \{1, 2, 3, \infty\}$ *and is* NP*-complete otherwise.*

Proof. As we commented above, the polynomial-time cases are covered by Algorithm **check-s-degen** and Lemmata 4.1 and 4.3. The remaining cases for node search are in NP because the fact that $\mathbf{vlns}^s(G) \leqslant k$ is witnessed by a (k,s)-layout $v_1, \ldots, v_n$, along with an $(s, v_i, \{v_1, \ldots, v_{i-1}\})$-separator of size at most k for each $i \in \{1, \ldots, n\}$ and the validity of such a witness can be checked in polynomial time. The same argument can be applied for mixed search.

It remains to prove that it is NP-hard to determine whether $\mathbf{vlns}^s(G) \leqslant k$ (resp., $\mathbf{vlms}^s(G) \leqslant k$) for finite $s \geqslant 4$. We first examine the node search case. In particular, we reduce the problem of determining whether $\mathbf{sep}^s_G(x, X) \leqslant k$ to the problem of determining whether $\delta^s(G) \leqslant k$. By Theorem 5.1, this is equivalent to the problem of determining whether $\mathbf{vlns}^s(G) \leqslant k$.

Given a graph G with n vertices, a vertex $x \in V(G)$, a set $X \subseteq V(G) - x$ and a non-negative integer k, we will construct a new graph G' such that $\mathbf{sep}^s_G(x, X) + |V(G)| = \delta^s(G')$. Take $k+n+1$ copies of G and, for each $v \in X$, identify the copies of v. We refer to the resulting set of vertices in G' as X and we write X^* for the set of copies of x. Then, add n new vertices (we call this set W) and add a new path of length s from each vertex in W to each vertex in X^*. We claim that $\delta^s(G') = \mathbf{sep}^s_G(x, X) + n$. For this, it is enough prove the following two claims for each $k \geqslant 0$.
Claim 1: $\mathbf{sep}^s_G(x, X) \leqslant k$ implies that $\delta^s(G') \leqslant k + n$.
Claim 2: $\mathbf{sep}^s_G(x, X) > k$ implies that $\delta^s(G') > k + n$.

For Claim 1, assume, on the contrary, that G' contains a $(n+k+1, s)$-hide-out R. We must have $R \subseteq W \cup X \cup X^*$, as every other vertex in G' has degree at most $n-1$. Notice also that, if $x \in X^* \cap R$, then x can be separated by $R - x$ by removing $|W| + \mathbf{sep}^s_G(x, X) \leqslant n + k$ vertices. Therefore, $R \subseteq W \cup X$. Notice now that each vertex in W is at least $s+1$ edges away from any other vertex in W or in X, therefore, $R \subseteq X$. This means that $|R| \leqslant |X| \leqslant n-1$ contradicting the fact that any $(n+k+1, s)$-hide-out R must contain at least $n+k+2$ vertices.

For Claim 2, first let X' be the set containing all vertices of X that are at distance at most s from x. As $\mathbf{sep}^s_G(x, X) \geqslant k+1 \geqslant 1$, X' contains at least one vertex. Notice also that $\mathbf{sep}^s_G(x, X') \geqslant k+1$, therefore $\mathbf{sep}^s_{G'}(x^*, X') \geqslant k+1$, for each $x^* \in X^*$. We claim that $R = W \cup X^* \cup X'$ is an $(n+k+1, s)$-hide-out in G'. For this, notice first that each vertex w of W is the beginning of $n+k+1$ disjoint paths to the vertices of X^*, therefore $\mathbf{sep}^s_{G'}(w, R - w) \geqslant \mathbf{sep}^s_{G'}(w, X^*) \geqslant k+n+1$. Now, let $x^* \in X^*$. Note that $\mathbf{sep}^s_{G'}(x^*, R - x^*) \geqslant \mathbf{sep}^s_{G'}(x^*, W \cup X') \geqslant n + \mathbf{sep}^s_{G'}(x^*, X') \geqslant n+k+1$. Finally, let $x \in X'$. As there are internally disjoint paths of length $\leqslant s$ from x to each vertex in X^* and $|X^*| \geqslant n+k+1$, we deduce that $\mathbf{sep}^S_{G'}(x, R - x) \geqslant \mathbf{sep}^S_{G'}(x, X^*) \geqslant n+k+1$.

We conclude that R is indeed a $(n+k+1, s)$-hide-out of G'; therefore $\delta^s(G') \geqslant n+k+1$.

For the NP-hardness of the mixed search case, we consider the following construction. Let G^* be the graph obtained by taking the disjoint union of two copies G^1 and G^2 of G and, for $i = 1, 2$, adding edges as follows: for each $v^i \in V(G^i)$, make v^i adjacent with all the vertices of $N_{G^{2-j}}[v^{3-i}]$ where v^{3-i} is the counterpart of v^i in G^{3-i} (see Figure 6.1 for an example).

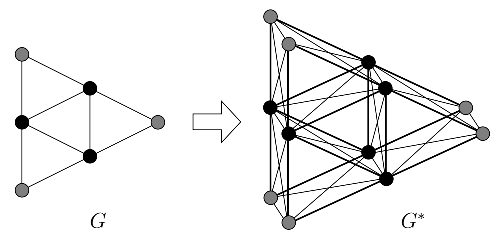

FIGURE 6.1. *The reduction for the NP-hardness of the mixed search case.*

We claim that $\mathbf{vlns}^s(G) = \lceil \frac{\mathbf{vlms}^s(G^*)}{2} \rceil$. From, Lemma 5.2.1, if $\mathbf{vlms}^s(G^*) = x$, then

$$x \leqslant \mathbf{vlns}^s(G^*) \leqslant x + 1. \tag{6.1}$$

Our argument now is based on the fact that $\mathbf{vlns}^s(G^*) = 2 \cdot \mathbf{vlns}^s(G)$, which is proved below. As then $\mathbf{vlns}^s(G^*)$ takes only even values, Relation (6.1) gives an exact estimation of $\mathbf{vlns}^s(G)$ and this proves the correctness of the claimed formula.

It remains to prove that $\mathbf{vlns}^s(G^*) = 2 \cdot \mathbf{vlns}^s(G)$. We present the proof for the case where $s \geqslant 2$ that is enough for our purposes (the case where $s = 1$ is very similar and simpler). According to Theorem 5.1, this requires the proof of the following two claims.

Claim 3: If G has a (k, s)-hide-out then G^* has a $(2k + 1, s)$-hide-out.
Claim 4: If G has a (k, s)-layout then G^* has a $(2k + 1, s)$-layout.

For Claim 3, assume that R is a (k, s)-hide-out of G. As for every $x \in R$, $\mathbf{sep}^s_G(x, R - x) \geqslant k$, we deduce that for every set $S \subseteq V(G) - x$ where $|S| < k$, there is some path of G, of length $\leqslant s$, from x to some vertex of $R - x$, avoiding S. For each $S \subseteq V(G) - x, |S| < k$, we pick such a path and we denote it by P_S. We also define the set $R^* = \bigcup_{v \in R} \{v^1, v^2\}$, containing the counterparts in G^1 and G^2 of each vertex in R and we set $R^h = R^* \cap V(G_h), h = 1, 2$.

Our aim is to prove that R^* is a $(2k + 1, s)$-hide-out of G^*. For this, let x^i be a vertex of R^* for some $i \in \{1, 2\}$. For each P_S, we denote by P^1_S and P^2_S the paths of G^* starting from x^i and continuing with the counterparts of the rest of the vertices of P_S in the graphs G_1 and G_2 respectively.

We have to prove that for every set $S^* \subseteq V(G^*) - x^i$ where $|S^*| \leqslant 2k$, G^* contains a path of length $\leqslant s$, from x^i to some vertex of $R^* - x^i$, avoiding S^*. Let S^* be such a set. Notice that $|N_G(x)| \geqslant k$, therefore, in G^*, x^i and its counterpart x^{3-i}, apart from being adjacent, have $\geqslant 2k$ common neighbors with x^i. This in turn implies that there are at least $2k + 1$ paths of length at most $2 \leqslant s$ from x^i to x^{3-i} in G^*. Clearly S^* cannot meet all these paths and, as $x^{3-i} \in R^*$, the claim holds in the case where $x^{3-i} \notin S^*$.

Suppose now that $x^{3-i} \in S^*$. We set $S^{*h} = (S^* - x^{3-h}) \cap V(G^i), h = 1, 2$. Notice that $|S^{*h} - x^{3-h}| \leqslant 2k - 1$, and therefore $|S^{*1}| + |S^{*2}| \leqslant 2k - 1$. This means that at least one, say S^{*1}, of S^{*1} and S^{*2} has at most $k - 1$ vertices. Recall that S^{*1} corresponds to a set S^1 of vertices in G. Therefore, P_{S^1} is a path of G^*, of length $\leqslant s$, from x^i to $R^* - x^i$, avoiding S^{*1}. As P_{S^1} avoids also S^{*2} and therefore S^* as well, the claim follows.

For Claim 4, we assume that $v_1, \ldots, v_n$ is an ordering of the vertices of G such that $\mathbf{sep}^s_G(v_i, \{v_1, \ldots, v_{i-1}\}) \leqslant k$ for every $i \in \{1, \ldots, n\}$. It is enough to prove that $v_1^1, v_1^2, \ldots, v_n^1, v_n^2$ is an ordering of the vertices of G^* such that for every $i \in \{1, \ldots, n\}$,

$$\mathbf{sep}^s_G(v_i^1, \{v_1^1, v_1^2, \ldots, v_{i-1}^1, v_{i-1}^2\}) \leqslant 2k + 1 \quad \text{and} \tag{6.2}$$
$$\mathbf{sep}^s_G(v_i^2, \{v_1^1, v_1^2, \ldots, v_{i-1}^1, v_{i-1}^2, v_i^1\}) \leqslant 2k + 1 \tag{6.3}$$

Recall that G contains an $(s, v_i, \{v_1, \ldots, v_{i-1}\})$-separator of size $\leqslant k$ for each $i \in \{1, \ldots, n\}$, i.e. a set, say $S_i \subseteq V(G) - v_i$, that meets all paths of G of length $\leqslant s$ from v_i to $\{v_1, \ldots, v_{i-1}\}$. For every $i \in \{1, \ldots, n\}$, we define S_i^1 and S_i^2 as the subsets of $V(G^*)$ containing the counterparts of the vertices of S_i in G^1 and G^2 respectively. Let $S_i^* = S_i^1 \cup S_i^2$ and fix some integer $i \in \{1, \ldots, n\}$. Observe that $v_i^1, v_i^2 \notin S^*$ and that $|S_i^*| \leqslant 2k$. To prove (6.2) and (6.3), it suffices to show that, for each $h \in \{1, 2\}$, $S_i^* \cup \{v_i^{3-h}\}$ meets all paths of G^* of length $\leqslant s$ from v_i^h to $\{v_1^1, v_1^2, \ldots, v_{i-1}^1, v_{i-1}^2\}$. Equivalently, we show that S_i^* meets all such paths that start from v_i^1 or v_i^2 and do not contain both v_i^1 and v_i^2. Indeed, suppose to the contrary, that S_i^* does not meet some path P_i^* of G^* that has length $\leqslant s$, starts from v_i^1 or v_i^2 but does not contain both v_i^1 and v_i^2, and finishes in some vertex in $\{v_1^1, v_1^2, \ldots, v_{i-1}^1, v_{i-1}^2\}$. By considering the counterparts of the vertices of P_i^* in G, we obtain a walk W_i in G from v_i to $\{v_1, \ldots, v_{i-1}\}$, of length $\leqslant s$. Certainly, we can extract from W_i a path P_i with the same properties. Recall that P_i contains some vertex $x \in S_i$. But then, both counterparts x^1 and x^2 of x will be vertices of S_i^* and at least one of them will be a vertex of P_i^*, a contradiction. □ In the preliminaries of this paper we restricted our study to simple graphs. If we drop this restriction, we may replace the last reduction of the proof of Lemma 6.1 by the following simpler one, suggested in [4]: Given any graph G, we define G^e as the graph created if we replace each edge in G by a double edge. Then it holds that $\mathbf{vlns}^s(G) = \mathbf{vlms}^s(G^e)$, which follows trivially using the arguments of [4].

7. Comparisons. In this section, we show that the two graph parameters that we have introduced, $\mathbf{vlns}^s$ and $\mathbf{vlms}^s$, along with tree-width are, as far as possible, independent for any subsequence of the parameter hierarchies they define. We concentrate our analysis on the simpler node search parameters as the same conclusions can be directly derived for the for the mixed search variant using the same ideas.

PROPOSITION 7.1. *For any graph G,*

$$\delta^1(G) \leqslant \delta^2(G) \leqslant \delta^3(G) \leqslant \cdots \leqslant \delta^\infty(G) \leqslant \mathbf{tw}(G)\,.$$

Proof. Immediate from the game characterizations of the parameters. Moving from left to right, the fugitive becomes stronger: from lazy with unit speed through lazy with increasing but still bounded speed, to lazy with unbounded speed, to active with unbounded speed. □

REMARK 1. *Proposition 7.1 indicates that the existence of a (k, ∞)-hide out can easily provide lower bounds for the treewidth of graphs. For instance, the graph H in*

Figure 4.2 contains a $(6, \infty)$-hide-out and therefore its treewidth is at least six. Since H is a minor of G, we have $\mathbf{tw}(G) \geqslant 6$. It is easily seen that a visible, active fugitive can be caught in G with seven cops, which establishes that $\mathbf{tw}(G) = 6$.

Recall that the $(n \times n)$ grid graph is the graph grid_n with vertices $\{1, \dots, n\}^2$ and all edges of the form $\{(i, j), (i+1, j)\}$ and $\{(i, j), (i, j+1)\}$. We will also require the following variant of the grid graphs. Let $\mathsf{hexgrid}_n$ be the graph that results from applying the following operations to every vertex (i, j) of degree 4 in grid_n:

1. add a new vertex $(i, j)'$;
2. delete the edges $\{(i-1, j), (i, j)\}$ and $\{(i, j-1), (i, j)\}$;
3. add edges $\{(i, j), (i, j)'\}$, $\{(i-1, j), (i, j)'\}$ and $\{(i, j-1), (i, j)'\}$.

The graphs grid_4 and $\mathsf{hexgrid}_4$ are illustrated in Figure 7.1. For $n \geqslant 4$, it can be seen that the "centre" of $\mathsf{hexgrid}_n$ is a hexagonal mesh.

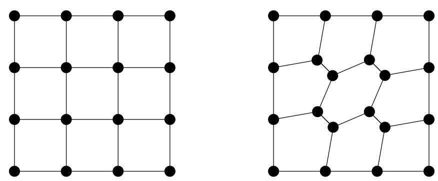

FIGURE 7.1. *The graphs* grid_4 *and* $\mathsf{hexgrid}_4$.

LEMMA 7.2. $\mathbf{tw}(\mathsf{grid}_n) = \mathbf{tw}(\mathsf{hexgrid}_n) = n$.

Proof. It is known (e.g., see [6]) that the treewidth of a $(\ell \times m)$-grid is equal to $\min\{\ell, m\}$. The result follows using the fact that grid_n is a minor of $\mathsf{hexgrid}_n$ and $\mathsf{hexgrid}_n$ is a minor of the $(2n \times n)$-grid as demonstrated in Figure 7.2. □

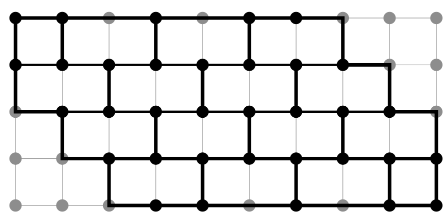

FIGURE 7.2. *The graph* $\mathsf{hexgrid}_5$ *seen as a minor of the* (10×5)*-grid.*

THEOREM 7.3. *Let* $3 \leqslant d_1 \leqslant d_2 \leqslant \cdots \leqslant d_r \leqslant d_{r+1} \leqslant d_{r+2}$. *There is a graph G such that* $\delta^1(G) = d_1$, $\delta^2(G) = d_2$, $\dots$, $\delta^r(G) = d_r$, $\delta^\infty(G) = d_{r+1}$ *and* $\mathbf{tw}(G) = d_{r+2}$.

Proof. For $i \in \{1, \dots, r+1\}$, let G_i be the graph obtained by independently replacing each edge xy of the complete graph K_{d_i+1} by an x–y path of length i. Let G_{r+2} be the graph obtained by independently replacing each edge xy of $\mathsf{hexgrid}_{d_{r+2}}$ by an x–y path of length $r+1$.

For any $i \in \{1, \dots, r+1\}$ and $j \in \{1, \dots, r\}$, it is easy to see that

$$\delta^j(G_i) = \begin{cases} 2 & \text{if } j < i \\ d_i & \text{if } j \geqslant i\,. \end{cases} \tag{7.1}$$

We have $\delta^\infty(G_{r+2}) = 3$. The upper bound follows from Lemma 3.1.1, because $\Delta(G_{r+2}) = 3$; the lower bound follows from Lemma 5.2.3 and the fact that G_{r+2}

contains K_4 as a topological minor. It is easily seen that, for all $i \in \{1, \dots, r+2\}$, $\mathbf{tw}(G_i) = d_i$.

Finally, let G be the union of disjoint copies of $G_1, \dots, G_{r+2}$. By Lemma 3.1.2, we have that, for any $s \in \{1, \dots, r\}$,

$$\begin{aligned}\delta^s(G) &= \max\{\delta^s(G_i) \mid 1 \leqslant i \leqslant r+2\} \\ &= \max\{d_1, \dots, d_s, 2, \dots, 2\} \\ &= d_s\,.\end{aligned}$$

Similarly, we have $\delta^\infty(G) = d_{r+1}$ and $\mathbf{tw}(G) = d_{r+2}$, as required. ◻

COROLLARY 7.4. *For any sequence* $3 \leqslant d_1 \leqslant d_2 \leqslant \cdots \leqslant d_{r+2}$*, there is a connected graph with the properties of the previous theorem.*

Proof. Take the graph G constructed in the theorem and arbitrarily choose vertices $x_i \in V(G_i)$ for each $i \in \{1, \dots, r+2\}$. Let $Y = \{y_1, \dots, y_{r+1}\}$ be new vertices and let G' be the graph formed from G by adding the vertices in Y and the edges $x_i y_i$ and $x_{i+1} y_i$ for all $i \in \{1, \dots, r+1\}$.

Let $s \in \mathbb{N}^+$ and let $k = \delta^s(G) \geqslant 3$. G is a subgraph of G' so, by Lemma 3.1.2, $\delta^s(G') \geqslant k$. Let $v_1, \dots, v_n$ be a (k,s)-layout of G. Since $k \geqslant 3$ and each y_i has degree two in G', $\mathbf{sep}^s_{G'}(y_i, X) \leqslant 2$ for any $i \in \{1, \dots, r+1\}$ and $X \subseteq V(G') - y_i$. Therefore $v_1, \dots, v_n, y_1, \dots, y_{r+2}$ is a (k,s)-layout of G' so $\delta^s(G') = k = \delta^s(G)$.

It is easily seen that, for $d \geqslant 1$, if disjoint graphs H and H' have treewidth at most d then any graph formed from $H \cup H'$ by adding a 2-path between the two constituent graphs also has treewidth at most d. Since $1 \leqslant \mathbf{tw}(G_i) \leqslant \mathbf{tw}(G)$, a simple induction shows that $\mathbf{tw}(G') = \mathbf{tw}(G)$. ◻

Theorem 7.3 and Corollary 7.4 hold also if we replace δ^s by δ^s_m. We omit the details as arguments are very similar.

8. Extending contraction degeneracy. A popular approach to estimating treewidth is to look for algorithms or heuristics that compute lower bounds for it. By Proposition 7.1, degeneracy gives such a lower bound. However, $\delta^1(\mathsf{hexgrid}_n) = 2$ for any $n \geqslant 2$ but the class of these hexagonal grids has unbounded treewidth.

Bodlaender, Koster and Wolle [7, 19, 30] define the *contraction degeneracy* of a graph to be

$$\delta\mathrm{C}(G) = \max\{\delta^1(H) \mid H \text{ is a non-trivial minor of } G\}\,.$$

Contraction degeneracy seemed to be a good lower bound for treewidth — notice that $\delta^1(G) \leqslant \delta\mathrm{C}(G) \leqslant \mathbf{tw}(G)$. Bodlaender et al. prove the problem of determining, given G and k, whether $\delta\mathrm{C}(G) = k$ is NP-complete and propose heuristics for computing the parameter [7].

We have defined the hierarchy δ^s for $s \in \mathbb{N}^+$, which can be seen as an extension of degeneracy. As $\delta^1(G) \leqslant \delta^\infty(G) \leqslant \mathbf{tw}(G)$, δ^∞ is, itself, a better lower bound for treewidth than degeneracy and is still polynomial-time computable, though the same examples as before show that there are graphs G with $\delta^\infty(G) = 3$ and arbitrary treewidth. However, we can follow the approach of Bodlaender et al. and define, for any $s \in \mathbb{N}^+$, the parameter

$$\delta^s\mathrm{C}(G) = \max\{\delta^s(H) \mid H \text{ is a non-trivial minor of } G\}\,.$$

Note that $\delta^1\mathrm{C}(G) = \delta\mathrm{C}(G)$. The following is immediate from Proposition 7.1.

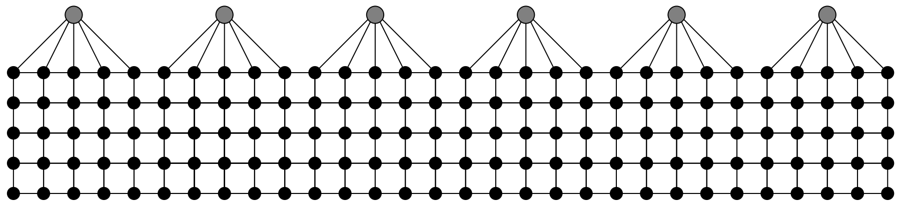

FIGURE 8.1. *The construction of Theorem 8.2 for $r = 5$; the $(5, \infty)$-hide-out consists of the grey vertices.*

PROPOSITION 8.1. *For any graph G,*

$$\delta^1\mathrm{C}(G) \leqslant \delta^2\mathrm{C}(G) \leqslant \delta^3\mathrm{C}(G) \cdots \leqslant \delta^\infty\mathrm{C}(G) \leqslant \mathbf{tw}(G)\,.$$

Thus, one can expect $\delta^\infty\mathrm{C}(G)$ to give a better lower bound for treewidth than contraction degeneracy. Unfortunately, $\delta^\infty(G)$ can, itself, be shown to be NP-complete — the proof is almost identical to the proof for contraction degeneracy in [7]. However, treewidth is bounded above by a function of $\delta^\infty\mathrm{C}(G)$, while contraction degeneracy gives only a lower bound, because $\delta\mathrm{C}(G) \leqslant 5$ for any planar G [7] but $\mathbf{tw}(G)$ can be arbitrarily large.

THEOREM 8.2. *There is a function $f : \mathbb{N} \to \mathbb{N}$ such that, for all graphs G, $\delta^\infty\mathrm{C}(G) \leqslant \mathbf{tw}(G) \leqslant f(\delta^\infty\mathrm{C}(G))$.*

Proof. The idea of the proof is that, for any r, the $(r \cdot (r+1) \times (r+1))$-grid graph (defined analogously to the $(r \times r)$-grid graph grid_r) can be contracted to a graph that contains an (r, ∞)-hide-out. To see this, remove the edges in $\{\{(i \cdot r, 1), (i \cdot r + 1, 1)\} \mid i = 1, \ldots, r\}$ and then contract all remaining edges of the form $\{(1, j), (1, j+1)\}$. The results of these contractions are $r + 1$ vertices that form the claimed (r, ∞)-hide-out (see Figure 8.1).

It is easy to verify that $(r \cdot (r+1), r+1)$ grid is a minor of an $(\mathcal{O}(r^{3/2}) \times \mathcal{O}(r^{3/2}))$-grid. If $\delta^\infty\mathrm{C}(G) \leqslant r$, then no minor of G contains $(r + 1, \infty)$-hide-out and, therefore, G does not contain an $(\mathcal{O}(r^{3/2}) \times \mathcal{O}(r^{3/2}))$-grid as a minor.

From [11,25], there is a function g such that, if G does not contain grid_k as a minor, then $\mathbf{tw}(G) \leqslant g(k)$. We conclude that, if $\delta^\infty\mathrm{C}(G) \leqslant k$, then $\mathbf{tw}(G) \leqslant g(\mathcal{O}(k^{3/2}))$. □

If a planar graph does not contain the $(k \times k)$-grid as a minor, then $\mathbf{tw}(G) \leqslant \mathcal{O}(k)$ [25]. Therefore, for planar G, $\mathbf{tw}(G) \leqslant \mathcal{O}((\delta^\infty\mathrm{C}(G))^{3/2})$. The same observation (with the same polynomial dependence) can be extended to any class of graphs with an excluded minor using the main result from [8].

9. Concluding remarks. We have studied the number of cops required to catch a lazy, visible fugitive moving with bounded or unbounded speed in a graph, using node search and the more general mixed search. We have shown that the associated search numbers correspond to graph parameters that are generalizations of the classical notion of degeneracy and characterized these parameters in terms of forbidden substructures, which we call hide-outs. Most parameters associated with graph searching are NP-complete in the case of fugitives with unbounded speed. However, our parameters are polynomial-time computable for fugitives with unbounded speed or speed at most three, and NP-complete for all other finite speeds.

As we mentioned in Section 4, checking whether $\mathbf{sep}^s_G(x, X) \leqslant k$ is an NP-complete problem for every fixed $s \geqslant 4$ because of the results in [3, 15, 20]. Recent results in [14] imply that, for every fixed s, this problem can be solved by an $s^k \cdot n^{O(1)}$-step algorithm (i.e., an FPT-algorithm). Apparently, using the algorithm of Section 6, the same type of algorithm can be derived for checking whether $\mathbf{vlns}^s(G) \leqslant k$ or $\mathbf{vlms}^s(G) \leqslant k$.

For most graph searching parameters, an important issue is to prove or disprove their monotonicity [13]. In the monotone versions of the games, the cops are only allowed to use strategies that gradually restrict the fugitive to smaller regions of the graph such that, once he has been cut off from a vertex in the graph, he can never return there. A game is said to be *monotone* if restricting the cops to using montone strategies does not increase the number of cops required on any graph. This property is highly desirable for searching games as it directly implies that the corresponding decision problem belongs in NP. Clearly, this is not the case for $\mathbf{vlns}^\infty$ and $\mathbf{vlms}^\infty$, since both corresponding decision problems are in P.

The natural way to define the monotonicity of the games that we examine this this paper is to consider only strategies where the fugitive never occupies or crosses a vertex that has already been occupied by a cop. Unfortunately, the strategies constructed in the first part of the proof of our min-max theorem (Theorem 5), are not monotone when $s \geqslant 2$, and it is not clear whether they could be replaced by monotone ones. Consequently, we leave monotonicity as an open issue. In fact, we conjecture that the games of this paper are monotone when $s = \infty$ and non-monotone when $2 \leqslant s < \infty$.

We have shown that δ^∞ can serve as a lower bound for treewidth and pathwidth and that δ^∞C approximates treewidth. It would be interesting to know whether there are graph classes where δ^∞ approximates treewidth or pathwidth or on which δ^∞C serves as a good approximation.

Acknowledgments.. We thank the anonymous reviewers for their valuable comments, remarks, and suggestions.

REFERENCES

[1] B. Alspach. Searching and sweeping graphs: A brief survey. *Matematiche (Catania)*, 59(1–2):5–37, 2004.
[2] S. Arnborg, D. G. Corneil, and A. Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic and Discrete Mathematics*, 8:277–284, 1993.
[3] G. Baier, T. Erlebach, A. Hall, E. Köhler, H. Schilling, and M. Skutella. Length-bounded cuts and flows. In *Proceedings of 33rd Intl. Colloquium on Automata, Languages and Programming (ICALP 2006), Part I*, volume 4051 of *LNCS*, pages 679–690. Springer, 2006.
[4] D. Bienstock. Graph searching, path-width, tree-width and related problems (a survey). In *Reliability of computer and communication networks*, volume 5 of *DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, pages 33–49. AMS, 1991.
[5] D. Bienstock and P. D. Seymour. Monotonicity in graph searching. *Journal of Algorithms*, 12(2):239–245, 1991.
[6] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.
[7] H. L. Bodlaender, T. Wolle, and A. M. C. A. Koster. Contraction and treewidth lower bounds. *Journal of Graph Algorithms and Applications*, 10(1):5–49, 2006.
[8] E. D. Demaine and MohammadTaghi Hajiaghayi. Linearity of grid minors in treewidth with applications through bidimensionality. *Combinatorica*, 28(1):19–36, 2008.
[9] N. D. Dendris, L. M. Kirousis, and D. M. Thilikos. Fugitive-search games on graphs and related parameters. *Theoretical Computer Science*, 172(1–2):233–254, 1997.
[10] R. Diestel. *Graph Theory.* Springer, 3rd edition, 2005.
[11] R. Diestel, T. R. Jensen, K. Yu. Gorbunov, and C. Thomassen. Highly connected sets and the excluded grid theorem. *Journal of Combinatorial Theory, Series B*, 75(1):61–73, 1999.

[12] F. V. Fomin and N. N. Petrov. Pursuit-evasion and search problems on graphs. In *27th Southeastern International Conference on Combinatorics, Graph Theory and Computing*, volume 122 of *Congr. Numer.*, pages 47–58, 1996.

[13] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3):236–245, 2008.

[14] P. A. Golovach and D. M. Thilikos. Paths of bounded length and their cuts: Parameterized complexity and algorithms. In *17th Annual European Symposium on Algorithms (ESA '09)*, LNCS. Springer, 2009.

[15] A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982.

[16] N. G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42(6):345–350, 1992.

[17] L. M. Kirousis and C. H. Papadimitriou. Interval graphs and searching. *Discrete Mathematics*, 55(2):181–184, 1985.

[18] L. M. Kirousis and D. M. Thilikos. The linkage of a graph. *SIAM Journal on Computing*, 25(3):626–647, 1996.

[19] A. M. C. A. Koster, T. Wolle, and H. L. Bodlaender. Degree-based treewidth lower bounds. Technical Report UU-CS-2004-050, Department of Information and Computing Sciences, Utrecht University, 2004.

[20] L. Lovász, V. Neumann-Lara, and M. Plummer. Mengerian theorems for paths of bounded length. *Periodica Math. Hungarica*, 9(4):269–276, 1978.

[21] D. W. Matula. A min–max theorem for graphs with application to graph coloring. *SIAM Reviews*, 10:481–482, 1968.

[22] T. D. Parsons. Pursuit-evasion in a graph. In *Proceedings of Intl. Conference on Theory and Applications of Graphs, 1976*, volume 642 of *Lecture Notes in Math.*, pages 426–441. Springer, 1978.

[23] N. N. Petrov. A problem of pursuit in the absence of information on the pursued. *Differentsial'nye Uravneniya*, 18(8):1345–1352, 1468, 1982.

[24] D. M. Richerby and D. M. Thilikos. Graph searching in a crime wave. In *Proceedings of 33rd Intl. Workshop on Graph-Theoretic Concepts in Computer Science (WG 2007)*, volume 4769 of *LNCS*, pages 21–32. Springer, 2007.

[25] N. Robertson, P. D. Seymour, and R. Thomas. Quickly excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 62(2):323–348, 1994.

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

[27] Y. C. Stamatiou and D. M. Thilikos. Monotonicity and inert fugitive search games. *Electronic Notes in Discrete Mathematics*, 3:184, 1999.

[28] A. Takahashi, S. Ueno, and Y. Kajitani. Mixed searching and proper-path-width. *Theoretical Computer Science*, 137(2):253–268, 1995.

[29] D. M. Thilikos. Algorithms and obstructions for linear-width and related search parameters. *Discrete Applied Mathematics*, 105:239–271, 2000.

[30] T. Wolle, A. M. C. A. Koster, and H. L. Bodlaender. A note on contraction degeneracy. Technical Report UU-CS-2004-042, Department of Information and Computing Sciences, Utrecht University, 2004.