Monotonicity and Inert Fugitive Search Games

Yannis C. Stamatiou [1]

Department of Computer Engineering and Informatics, Patras University, Rio, 265 00 Patras, Greece. e-mail: stamatiu@ceid.upatras.gr

Dimitrios M. Thilikos [2]

Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord – Mòdul C5, Desp. 211b, c/Jordi Girona Salgado, 1-3. E-08034, Barcelona, Spain. email: sedthilk@lsi.upc.es

Abstract

In general, graph search can be described in terms of a sequence of searchers' moves on a graph, able to capture a fugitive resorting on its vertices/edges. Several variations of graph search have been introduced, differing on the abilities of the fugitive as well as of the search. In this paper, we examine the case where the fugitive is inert, i.e., it moves only whenever the search is about to capture it. Mainly, there are two variants for "clearing" an edge during a search: when a sliding of a searcher occurs along the edge or when both its endpoints are simultaneously occupied by searchers. These variants define the inert edge search and the inert node search respectively. A third search variant, the inert mixed search, is defined when both ways of clearing an edge are possible. As we show, inert search and inert mixed search are equivalent (surprisingly, this is not the case if we discard the inertness property). Moreover, we prove that, in any case, by restricting the searches to only those that always reduce further the fugitive's possible resorts, does not give any advantage to the fugitive (this monotonicity property is usually expressed as: "recontamination does not help"). So far, the only monotonicity result on inert search concerns inert node search and our results yield a much simpler proof of that result as well. Furthermore, we define a new graph-theoretic parameter, the *proper-treewidth*, in analogy to the parameter *proper-pathwidth*, and prove it equivalent to the inert mixed search game. Last, we prove that proper-treewidth, in turn, is equivalent to a known graph theoretic parameter related to treewidth.

Technical Report of the "Departament de Llenguatges i Sistemes Informàtics"

1 Introduction

We consider loopless, undirected connected graphs that may have multiple edges. On an informal level, we intend to capture a fugitive that resides on the vertices or edges of a graph. For the purposes of our game, it is convenient to view the graph as a model representing a system of tunnels where a fugitive hides. A search is a sequence of searchers' moves that can be of the following three types:

(i) Removal of a searcher from vertex v.
(ii) Placement of a searcher on vertex u.
(iii) Sliding of a searcher that currently resides on v along an edge $\{v, u\}$ (after this move, the searcher is supposed to reside on vertex u).

We consider the fugitive to be *inert*: it moves *only* when the search is about to capture it and this can happen when it currently resides on some vertex v and the next step of the search involves a move of type (ii) or (iii) that will have as a result the placement of a searcher on v. Moreover, the fugitive is *fast*: whenever it moves, it can go to any unguarded vertex moving along any unguarded path. We also assume that the fugitive is *invisible*: the search is given in advance and no further knowledge based on the moves of the fugitive can be exploited. Finally, the fugitive is *omniscient*: it has complete knowledge of our search and whenever it *moves*, it always chooses the most advantageous position for it.

After each step of the search, we call the locations where the fugitive could resort *contaminated*, otherwise we call them *clear*. (A location can be a vertex or an edge.) Initially, only the fugitive resides on the graph and all the locations of the graph are contaminated. After each move, new locations are declared clear and it is possible that locations that have already been cleared, to be recontaminated. Certainly, a vertex is declared clear whenever a searcher is placed on it. On the other side, there are two possible ways for an edge to be declared clear:

A: *if both its endpoints are occupied by a searcher* or

[1] The research of the first author was performed while he was visiting the School of Computer Science of Carleton University, and was supported by the Greek Ministry of National Economy through a NATO scholarship for conducting postdoctoral studies (contract number 106384/ΔOO 1222/2-7-98).
[2] The research of the second author was partially supported by the Natural Sciences and Engineering Research Council of Canada (while he was visiting the Department of Computer Science of the University of Waterloo, Waterloo, Canada) and by the Ministry of Education and Culture of Spain – Grant no MEC-DGES SB97 0K148809.

B: *if a searcher slides along it*, i.e., a searcher moves from one endpoint of the edge to the other.

The object of an inert mixed search is to clear all the locations of a graph. The inert mixed search number of an inert mixed search, is the maximum number of searchers on the graph during any move. The *inert mixed search number* of a graph G, $\mathrm{ims}(G)$, is the minimum inert mixed search number, over all the possible successful inert mixed searches on it. A move causes *recontamination* of a clear location, if it causes the appearance of a path without searchers connecting it with an already contaminated location. (Recontaminated locations must be cleared again.) A search without recontamination is called *monotone*.

The *inert node (edge) search*, is defined similarly to the inert mixed search with the difference that an edge can be cleared only if **A** (**B**) happens. We define the *inert node (edge) search number* of a graph G analogously and we denote it as $\mathrm{ins}(G)(\mathrm{ies}(G))$.

Obviously, the inert edge search and the inert node search are special cases of the inert mixed search. Moreover, given an inert mixed search, we can transform it to an inert edge search that uses the same number of searchers, in the following way: whenever an edge is cleared because of **A**, we, instead, clear it by sliding a searcher residing on one of its endpoints towards the other, i.e. using **B**. The result of this action will be the same, as the fugitive is inert and no recontamination can occur during it.

As the essential variations of the inert search are the mixed search and the node search, we may focus our attention on clearing only the vertices of the graph and assuming that an edge is automatically considered clear whenever both its endpoints are clear. We can now give formal definitions of these two versions.

Given two subsets A, B of a set V, we denote their symmetric difference as $A \bigtriangledown B$.

An inert mixed search on a graph G is a sequence $\mathcal{S} = (S_0, \ldots, S_r)$ of sets of vertices ($S_i \subseteq V(G), i = 1, \ldots, r$) where $S_0 = \emptyset$ and for all $i = 1, \ldots, r$ either $|S_i \bigtriangledown S_{i-1}| = 1$ or $S_i \bigtriangledown S_{i-1}$ induces a single edge in G (we exclude the case where a multiple edge is induced).

The set of free locations for an inert mixed search is defined as follows:

- $F_0 = V(G)$.
- For $i = 1, \ldots, r$, $F_i = (F_{i-1} - S_i) \cup \{v \in V(G) - S_i :$ there is a path from a vertex $u \in F_{i-1} \cap (S_i - S_{i-1})$ to v whose vertices are not in $S_i - \{u\}$ and whose edges are not in $E(G[S_i \bigtriangledown S_{i-1}])\}$.

Intuitively, an inert fugitive is allowed to move only when a searcher is about to be placed on the vertex it currently occupies. This is so, because the fugitive can move away from a vertex u only if $u \in F_{i-1} \cap (S_i - S_{i-1})$. A search $\mathcal{S}$ is a search for a vertex set $V \subseteq V(G)$ if $F_i = V(G) - V$. An inert mixed search is complete (or it is a inert mixed search on G), if it is a search for $V(G)$. Observe that the inert mixed search number of $\mathcal{S}$ is equal to $\max\{|S_i| \mid i = 0, \ldots, r\}$ and that $\mathrm{ims}(G)$ is the minimum inert mixed search number over all the possible complete inert mixed searches on it. A search is called *monotone* if $\forall_{1 \leq i \leq r} F_i \subseteq F_{i-1}$.

If, in the above definitions, we further demand that $|S_{i-1} \bigtriangledown S_i| \leq 1$, we have the search version for the inert node search introduced and studied in [6]. Clearly, in this case, $E(G[S_{i-1} \bigtriangledown S_i])$ is always empty and this expresses the fact that sliding a searcher is no more possible. If we discard the requirement for the fugitive to be inert, we have the classic versions of graph search, namely the (agile) edge-, node-, and mixed- search, defining the parameters $\mathrm{es}(G)$, $\mathrm{ns}(G)$, and $\mathrm{ms}(G)$ respectively (the crucial difference with our games is that the fugitive is agile, i.e. it always moves, no matter if the search threatens it or not). We stress that, in contrast with the inert case agile edge-search is not equivalent to agile mixed-search.

Edge search was the search game to be defined first, introduced by Breisch [4] and Parsons [18] (see also [16]). Node search appeared as the first variant of edge search and was introduced by Kirousis and Papadimitriou in [14]. Finally, mixed search was introduced in [2] and [23]. It is worth mentioning that $\mathrm{ns}(G) - 1$ and $\mathrm{ins}(G) - 1$ are equal to the pathwidth and the treewidth of G respectively (see [6,7,12–14,17]). For surveys concerning graph searching and related parameters see [1,3,9].

The recontamination question for a search game asks whether it is equivalent to its monotone version, i.e., whether excluding all the non-monotone searches, reduces the searchers' ability. If the answer is no, we say that the corresponding search game is monotone or that recontamination does not help. A lot of research has been done on the recontamination question. The first monotonicity proof was concerned with edge-search and was given by LaPaugh in [15]. The proof of the monotonicity of node-search was given by Kirousis and Papadimitriou in [14] and the one for mixed-search, by Bienstock and Seymour in [2], where a unifying and relatively simpler proof for all the previous versions was presented. As far as the inert search games are concerned, the only version known to be monotone is the inert node search proved in [6]. The proof is strongly based on the equivalence of $\mathrm{ins}(G)$ and treewidth and the existence of a min-max theorem for treewidth proven by Seymour and Thomas in [21].

In this paper, we prove that recontamination does not help for inert edge search and, therefore, for the equivalent mixed search as well. Our proof is relatively simple and, as a corollary, yields the monotonicity of inert node search in a simpler and more direct way than the one used in [6]. The above are summarized in the following table.

	Edge search	Node search	Mixed search
Agile fugitive	LaPaugh ([15])	Kirousis and Papadimitriou ([14])	Bienstock and Seymour ([2])
Inert fugitive	This paper	Dendris et al. ([6])	This paper

In our proof, we follow an approach parallel to the one used by Bienstock and Seymour in [2]. Our main tool is the concept of an *expansion* on a graph. In the next section we give its formal definition and prove the relevant monotonicity result. In sections 3 and 4 we prove the equivalence of the inert mixed search number with the graph theoretic parameters of proper-treewidth and la (for the definitions see Sections 3 and 4 respectively). Our results constitute the "tree" counterpart of the known equivalence of the agile mixed search number and proper-pathwidth and are based on a complete characterization of the chordal graphs that have inert mixed search number $\leq k$. In Section 5, we summarize our results as well as their counterparts for other search parameters.

2 The concept of an expansion on a graph

In this section we will define the concept of an *expansion* on a graph, that will be our main tool to derive the monotonicity result for the inert edge/mixed search game. In what follows, we consider only graphs without multiple edges, since if a graph has multiple edges between a pair of vertices v, u, we may subdivide all the multiple edges connecting v and u without harming the inert search number.

We first give some helpful auxiliary definitions. Given a graph G, a set $F \subseteq V(G)$, and a vertex $v \in F$, we define as $C^v[F]$ the connected component of $G[F]$ that contains v. We also set $\overline{F} = V(G) - F$. If $E \subseteq E(G)$, we define as $\partial_E F$ the set of vertices in $\overline{F}$ that are adjacent to vertices in F through edges *not* in E (we may sometimes abuse the notation and let F be a graph instead of a set of vertices). Notice that $\partial_E V(C^v[F]) \subseteq \overline{F}$, i.e., it does not contain vertices in $F - V(C^v[F])$.

We now define an *expansion on* G to be any sequence $\mathcal{E} = (X_0, \ldots, X_r)$ of subsets of $V(G)$, where $X_0 = \emptyset$, $X_r = V(G)$, and such that

(1) $\forall_{1 \leq i \leq r}\ |X_i - X_{i-1}| \leq 1$.

(2) $\forall_{1\leq i\leq r}$ if $X_i \subseteq X_{i-1}$, then all the vertices of $X_{i-1} - X_i$ belong in the same component of $G[\overline{X}_i \cup \overline{X}_{i-1}]$.

For $i = 1, \ldots, r$, we define $\phi(X_i)$ as follows:

• If $X_i - X_{i-1} = \emptyset$, then $\phi(X_i) = \max\{|\partial V(C)|$, where C is a connected component of $\overline{X}_i\}$.
• If $X_i - X_{i-1} = \{p\}$, then $\phi(X_i) = \min\{|\partial_E V(C^p[\overline{X}_i \cup \{p\}])|$, where E is either empty or it contains an edge connecting p with some vertex in the neighborhood of v in $G[X_i]\} + 1$.

If $\mathcal{E} = (X_0, \ldots, X_r)$ is an expansion on G, we will say that $\mathcal{E}$ uses $\leq k$ guards, if $\forall_{1\leq i\leq r}\phi(X_i) \leq k$. We call an expansion $\mathcal{E} = (X_0, \ldots, X_r)$ on G *monotone* if, for any $i = 1, \ldots, r$, $X_{i-1} \subseteq X_i$ and $|X_i - X_{i-1}| = 1$.

Given an expansion $\mathcal{E}$ on a graph G, we define $M(\mathcal{E}) = \{|\mathcal{E}|\} \cup \{1 \leq i < |\mathcal{E}| \mid X_{i-1} \not\subseteq X_i\}$ and we set $m(\mathcal{E}) = \min M$. Clearly, an expansion $\mathcal{E}$ is monotone if and only if $m(\mathcal{E}) = |\mathcal{E}|$. In case $m(\mathcal{E}) < |\mathcal{E}|$, we set $h(\mathcal{E}) = \max\{0 \leq j < m(\mathcal{E}) \mid X_{m(\mathcal{E})} \not\subseteq X_j\}$.

Lemma 1 *If there exists an inert edge search (possibly non-monotone) on a graph G that uses $\leq k$ searchers, then there exists an expansion on G that uses $\leq k$ guards.*

PROOF. Let $\mathcal{E} = (\overline{F}_0, \overline{F}_1, \ldots, \overline{F}_r)$. Let X_{i-1}, X_i be two consecutive elements of $\mathcal{E}$ for some $i = 1, \ldots, r$. We claim that when $\overline{F}_{i-1} \neq \overline{F}_i$ expansion requirements 1 and 2 hold. We set $\overline{F}_{i-1} = \overline{F}_i$. Otherwise, we set $X_{i-1} = \overline{F}_{i-1}, X_i = \overline{F}_i$.

In the case where $S_i \subseteq \overline{F}_{i-1}$, then $S_i - S_{i-1} \subseteq \overline{F}_{i-1}$ and thus $(S_i - S_{i-1}) \cap F_{i-1} = \emptyset$. From the definition of $\overline{F}_i$ we have that $\overline{F}_i = \overline{F}_{i-1} \cup S_i = \overline{F}_{i-1}$.

Suppose now that $S_i \not\subseteq \overline{F}_{i-1}$. We may assume that $S_i - S_{i-1} = \{p_i\}$. Notice that $p_i \notin S_{i-1}$ and $S_{i-1} \subseteq \overline{F}_{i-1}$ imply that $p_i \notin \overline{F}_{i-1}$. Combining this with the fact that $p_i \in S_i \subseteq \overline{F}_i$, we have that $\{p_i\} \subseteq \overline{F}_i - \overline{F}_{i-1}$. From the definition of F_i we have that $\overline{F}_i \subseteq \overline{F}_{i-1} \cup S_i$ and as $S_{i-1} \subseteq \overline{F}_{i-1}$ we can argue that $\overline{F}_i \subseteq \overline{F}_{i-1} \cup (S_i - S_{i-1}) = \overline{F}_{i-1} \cup \{p_i\}$. From the last relation, we have that if a vertex belongs to $\overline{F}_i$ but not in $\overline{F}_{i-1}$ it should be u. Therefore $\overline{F}_i - \overline{F}_{i-1} \subseteq \{p_i\}$. We can now conclude that $\overline{F}_i - \overline{F}_{i-1} = \{p_i\}$ and thus requirement 1 holds. Towards proving requirement 2, notice that $F_{i-1} \cap (S_i - S_{i-1}) = \{p_i\}$. Let $R_i = X_i - X_{i-1}$. Then we have that any vertex v in $R_i \cup \{p_i\}$ is connected with u via a path of vertices in $R_i \cup \{p_i\}$. As $\overline{F}_{i-1} - \overline{F}_i \subseteq R_i \cup \{p_i\}$, $\{p_i\} = \overline{F}_i - \overline{F}_{i-1}$ and $R_i \subseteq F_i \cup F_{i-1}$, requirement 2 follows.

We claim that, if $\{p_i\} = S_i - S_{i-1}$ and $E = E(S_i \bigtriangledown S_{i-1})$, then $|\partial_E V(C^{p_i}[F_i \cup$

$\{p_i\}])| \leq k-1$.

By the definition of R_i we have that $G[R_i \cup \{p_i\}]$ is one of the connected components of $G[F_i \cup \{p_i\}]$. As $R_i \cup \{p_i\} \subseteq F_i \cup \{p_i\}$, we have that $G[R_i \cup \{p_i\}] = C^{p_i}[F_i \cup \{p_i\}]$.

Notice now that $\partial_E V(G[R_i \cup \{p_i\}])$ is a subset of $\overline{F}_i - \{p_i\}$ containing any vertex $v \in \overline{F}_i$ that is connected with p_i with a path whose vertices, except v, are in $R_i \cup \{p_i\}$ and without edges in $E(G[S_i \bigtriangledown S_{i-1}])$. From the definition of R_i, we have that each such vertex must be a member of $S_i - \{p_i\}$ and, as $|S_i - \{p_i\}| \leq k-1$, the claim follows.

It is now easy to see that, by replacing, in $\mathcal{E}$, any maximal subsequence $(\overline{F}_i, \ldots, \overline{F}_{i+\sigma})$, $\sigma > 0$, where $\overline{F}_i = \cdots = \overline{F}_{i+\sigma}$ by $\overline{F}_i$, we obtain a new sequence $\mathcal{E}' = \{X_1, \ldots, X_{r'}\}$ that is an expansion on G that uses $\leq k$ guards. $\square$

Lemma 2 *If there exists an expansion (possibly non-monotone) on G that uses $\leq k$ guards, then there exists a monotone expansion on G that uses $\leq k$ guards.*

PROOF. Lemma 2 is based on the algorithm that appears in Figure 1 that transforms a non-monotone expansion to a monotone one.

Algorithm: Conversion to monotone expansion.
Input: An expansion $\mathcal{E}$ on a graph $G(V,E)$ using k guards.
Output: A monotone expansion $\mathcal{E}$ on $G(V,E)$ using k guards.

1. **begin**
2.   **if** $m(\mathcal{E}) = |\mathcal{E}|$ **then output** $\mathcal{E}$
3.   **if** $h(\mathcal{E}) < m(\mathcal{E}) - 1$ **then**
   $\mathcal{E} \leftarrow (X_0, \ldots, X_{h(\mathcal{E})}, X_{m(\mathcal{E})}, \ldots, X_{|\mathcal{E}|})$ and **goto** 2
4.   **if** $X_{h(\mathcal{E})} = X_{h(\mathcal{E})-1}$ **then**
   $\mathcal{E} \leftarrow (X_0, \ldots, X_{h(\mathcal{E})-1}, X_{m(\mathcal{E})}, \ldots, X_{|\mathcal{E}|})$ and **goto** 2
5.   **let** $\{p'\} = X_{m(\mathcal{E})-1} - X_{m(\mathcal{E})-2}$ and $\{p\} = X_{m(\mathcal{E})} - X_{m(\mathcal{E})-1}$
6.   **if** $p \notin V(C^{p'}[\overline{X}_{i-1})])$ **then**
   $\mathcal{E} \leftarrow (X_0, \ldots, X_{m(\mathcal{E})-2}, X_{m(\mathcal{E})} - \{p'\}, X_{m(\mathcal{E})}, \ldots X_{|\mathcal{E}|})$ and **goto** 2
7. $\mathcal{E} \leftarrow (X_0, \ldots, X_{m(\mathcal{E})-2}, X_{m(\mathcal{E})} - \{p\}, X_{m(\mathcal{E})}, \ldots X_{|\mathcal{E}|})$
   and **goto** 2
8. **end**

Fig. 1. Converting a non-monotone expansion to a monotone one

Setting $\mathcal{E}^{(0)} = \mathcal{E}$, the input expansion, let $\mathcal{E}^{(j)}$, $j > 0$, denote the input expansion after the j-th return to line 2 of the algorithm, if the algorithm has not already terminated before completing j such returns. Otherwise, $\mathcal{E}^{(j)}$ is undefined. From now on, when we refer to $\mathcal{E}^{(j)}$ it is implied that $\mathcal{E}^{(j)}$ is defined.

We will prove that the following properties hold for the algorithm given above:

I_1: For each j, $\mathcal{E}^{(j)}$ is an expansion.

I_2: For each j, if $\mathcal{E}^{(j)} = X_0, \ldots, X_r$, then $\phi(X_i) \leq k$, $0 \leq i \leq r$.
I_3: The algorithm terminates.

Let us first prove I_1. The defining properties of an expansion need only be checked locally, at the position where the algorithm modifies $\mathcal{E}^{(j-1)}$ to give $\mathcal{E}^{(j)}$. We will be concerned with property 1 first.

Consider step 3. Then $|X_{m(\mathcal{E})} - X_{h(\mathcal{E})}| \leq 1$ for, otherwise, it would hold $|X_{h(\mathcal{E})+1} - X_{h(\mathcal{E})}| > 1$ since, from the definition of $h(\mathcal{E})$, $X_{m(\mathcal{E})} \subseteq X_{h(\mathcal{E})}$. However, this contradicts the fact that $\mathcal{E}^{j-1}$ was an expansion.

For step 4, property 1 holds trivially. Let us then consider step 6. It holds that $|(X_{h(\mathcal{E})} - \{p'\}) - X_{h(\mathcal{E})-2}| = \{p\}$. Also, it is easy to see that $X_{m(\mathcal{E})} - (X_{h(\mathcal{E})} - \{p'\}) = \{p\}$. Therefore, property 1 holds.

For step 7 now, $(X_{m(\mathcal{E})} - \{p\}) - X_{m(\mathcal{E})-2} = (X_{m(\mathcal{E})} - X_{m(\mathcal{E})-2}) \cap \overline{\{p\}}$. However, the difference $X_{m(\mathcal{E})} - X_{m(\mathcal{E})-2}$ can only be equal to $\emptyset$ or p', if $p' \in X_{m(\mathcal{E})}$. Therefore property 1 holds again.

Let us turn now to property 2 and consider line 3 first. It holds that $X_{h(\mathcal{E})} - X_{m(\mathcal{E})} \subseteq X_{m(\mathcal{E})-1} - X_{m(\mathcal{E})}$, because $X_{h(\mathcal{E})} \subseteq X_{m(\mathcal{E})-1}$. Then if there were two vertices in $X_{h(\mathcal{E})} - X_{m(\mathcal{E})}$ such that they belonged to two different components in $G[\overline{X_{h(\mathcal{E})} \cap X_{m(\mathcal{E})}}]$, then they would also belong to two different components in $G[\overline{X_{m(\mathcal{E})} \cap X_{m(\mathcal{E})-1}}]$, contradicting the fact that $\mathcal{E}^{(j-1)}$ was an expansion.

Property 2 also holds trivially for $\mathcal{E}^{(j)}$, after the execution of line 4. Let us then consider line 6 of the algorithm and the difference $X_{m(\mathcal{E})-2} - (X_{m(\mathcal{E})} - \{p'\}) = (X_{m(\mathcal{E})-1} \cup \{p'\}) - (X_{m(\mathcal{E})} - \{p'\}) = X_{m(\mathcal{E})-1} - X_{m(\mathcal{E})}$. Also, $X_{m(\mathcal{E})-2} \cap (X_{m(\mathcal{E})} - \{p'\}) = (X_{m(\mathcal{E})-1} - \{p'\}) \cap (X_{m(\mathcal{E})} - \{p'\}) = X_{m(\mathcal{E})-1} \cap X_{m(\mathcal{E})}$. Therefore, for property 2 to hold, all vertices in $X_{m(\mathcal{E})-1} - X_{m(\mathcal{E})}$ should belong to the same component of $G[\overline{X_{m(\mathcal{E})-1} \cap X_{m(\mathcal{E})}}]$, which is true since $\mathcal{E}^{(j-1)}$ was an expansion.

Finally, consider line 7 of the algorithm. Then $X_{m(\mathcal{E})-2} - (X_{m(\mathcal{E})} - \{p\})$ is again equal to $X_{m(\mathcal{E}-1)} - X_{m(\mathcal{E})}$. Also, $X_{m(\mathcal{E})-2} \cap (X_{m(\mathcal{E})} - \{p\}) = X_{m(\mathcal{E})-1} \cap X_{m(\mathcal{E})}$. Therefore, like in step 6 of the algorithm, property 2 holds.

We will now show that all modifications made by the algorithm on expansion $\mathcal{E}^{(j-1)}$, produce an expansion $\mathcal{E}^{(j)}$ that uses at most that many guards as $\mathcal{E}^{(j-1)}$. Consider first line 3. If $h(\mathcal{E}) = m(\mathcal{E})$ in $\mathcal{E}^{(j-1)}$, then no modification is effected by line 3. Otherwise, let let $\{p\} = X_{m(\mathcal{E})} - X_{h(\mathcal{E})}$. In $\mathcal{E}^{(j-1)}$, $X_{m(\mathcal{E})} - X_{h(\mathcal{E})+1} = \emptyset$ (again from the definition of h). Therefore, in $\mathcal{E}^{(j-1)}$, it was true that $\phi(X_{m(\mathcal{E})}) = \max\{|\partial V(C)|\} \leq k$, where C is a connected component of $\overline{X}_{m(\mathcal{E})}$. We must now show that $\phi(X_{m(\mathcal{E})})$ is still at most k, but now under the second branch of the definition of ϕ, since $X_{m(\mathcal{E})} - X_{h(\mathcal{E})} = \{p\}$. If $h(\mathcal{E}) = m(\mathcal{E})$, then $\phi(X_{m(\mathcal{E})}) \leq k$, since line 3 is not executed. If in the second branch of the definition of ϕ, E is empty, then clearly $\phi(X_{m(\mathcal{E})}) \leq k$. Otherwise, consider the set S_p of all edges that join p with vertices of $X_{m(\mathcal{E})}$. We claim that there exists an edge $e \in S_p$, such that that $\phi(X_{m(\mathcal{E})}) = \min\{|\partial_{\{e\}} V(C^p[\overline{X}_{m(\mathcal{E})} \cup \{p\}])|\} \leq k-1$, where e connects p with some vertex in the neighborhood of p in $G[X_{m(\mathcal{E})}]$. Suppose in contrary that for every e, $\phi(X_{m(\mathcal{E})}) \geq k$. This means that if we exclude p, there are at least k vertices in $\{\partial V(C)\}$, where C is the connected component of $\overline{X_{m(\mathcal{E})} - \{p\}}$ that includes p. If p belonged in this component, we would have $\phi(X_{m(\mathcal{E})}) \geq k+1$ in $\mathcal{E}^{(j-1)}$, which is not true since we assumed that the previous expansion used k guards. Therefore, an edge e such that $\phi(X_{m(\mathcal{E})}) \leq k-1$ in the new expansion $\mathcal{E}^{(j)}$ must exist.

Property I_2 holds trivially for step 4. Let us consider step 6. Now $X_{m(\mathcal{E})} - (X_{m(\mathcal{E})} - \{p'\}) = \{p\}$ and we must show that, according to the second branch of the definition of ϕ, $\phi(X_{m(\mathcal{E})}) \leq k$. This holds because if we consider the previous expansion $\mathcal{E}^{(j-1)}$ and the sets $X_{m(\mathcal{E}-1)}$ and $X_{m(\mathcal{E})-2}$ for which it holds $X_{m(\mathcal{E}-1)}$ - $X_{m(\mathcal{E})-2} = \{p'\}$, we have that $\phi(X_{m(\mathcal{E})-1}) \leq k$, again with the second branch of the definition of ϕ. Therefore, this must hold for $\mathcal{E}^{(j)}$ too, since the components that contain p' in $\overline{X_{m(\mathcal{E})-1} - \{p'\}}$ and $\overline{X_{m(\mathcal{E})} - \{p'\}}$ are the same. Moreover, for the sets $(X_{m(\mathcal{E})} - \{p'\})$ and $X_{m(\mathcal{E})-2}$ whose difference is equal to $\{p\}$, it also holds that $\phi(X_{m(\mathcal{E})} - \{p'\}) \leq k$ for the same reason.

In step 7 of the algorithm now, the situation is similar to step 6. The difference $X_{m(\mathcal{E})} - X_{m(\mathcal{E})-1}$ is a set with exactly one element, the element p. Since $\mathcal{E}^{(j-1)}$ uses k guards, it holds that $\phi(X_{m(\mathcal{E})}) \leq k$, using the second branch of the definition of ϕ. After the execution of step 7, it is easy to see that $\phi(X_{m(\mathcal{E})}) \leq k$ since the second branch of the definition of ϕ can be used again on the same set, $X_{m(\mathcal{E})}$. Now consider the sets $X_{m(\mathcal{E})} - \{p\}$ and $X_{m(\mathcal{E})-2}$. Their difference can be either $\{p'\}$ or $\emptyset$. In the first case, we observe that in $\mathcal{E}^{(j-1)}$, it holds that $\phi(X_{m(\mathcal{E})}) \leq k$, according to the second branch of the definition of ϕ, for some edge singleton E. This means that when p is introduced in $X_{m(\mathcal{E})})$, k guards are used again since the vertex adjacent to p via the edge in E, does not contribute anymore to ϕ. Now the same reasoning is applied, as for step 6, to prove that there exists an edge set E for which the second branch of the definition of ϕ gives k guards in the new expansion and for the vertex p'. Now suppose $(X_{m(\mathcal{E})} - \{p\}) - X_{m(\mathcal{E})-2} = \emptyset$. Then $p' \notin X_{m(\mathcal{E})}$ and $(X_{m(\mathcal{E})} - \{p\}) \subseteq X_{m(\mathcal{E})-2}$. Then we should use the first branch of the definition of ϕ. Using again the edge set E from the second branch of the definition, for the expansion $\mathcal{E}^{(j-1)}$ and the sets $X_{m(\mathcal{E})}$ and $X_{m(\mathcal{E})} - \{p\}$ that differ in the vertex p, when p is introduced again at most k guards are used in $\mathcal{E}^{(j)}$ since the vertex adjacent to p via the edge in E, does not participate anymore in the calculation of ϕ.

Finally, in order to prove that the algorithm terminates, it suffices to observe that each of the modifications in lines 3, 4, 6 and 7 either decrease the size of a set by 1 (6 and 7) or decrease the number of sets in an expansion (lines 3 and 4). $\square$

Before we proceed, we will introduce some usefull notation. Let $\mathcal{E} = (X_0, \ldots, X_r)$ be a monotone expansion. We set $\mathcal{P} = (p_1, \ldots, p_r)$, where $\{p_i\} = X_i - X_{i-1}, 1 \leq i \leq r$. We can now define, for each particular G and $\mathcal{E}$, a function ρ where $\rho(X_i)$ is chosen as a set in

> $\{\partial_{E_i} V(C^{p_i}[\overline{X}_i \cup \{p_i\}])$ where E_i is either empty or it contains an edge connecting p_i with some vertex in the neighborhood of p_i in $G[X_i]$ $\}$

whose cardinality determines the value of $\phi(X_i)$ (in particular, $\phi(X_i) = |\rho(X_i)| + 1$). Clearly, the definition of $\rho(X_i)$ can be accompanied with the corresponding sequence $(E_1, \ldots, E_r)$ of edge sets that give to ϕ its optimal values.

Lemma 3 *If there exists a monotone expansion on G that uses $\leq k$ guards, then there exists a monotone inert edge search on G that uses $\leq k$ searchers.*

PROOF. Let $\mathcal{E} = (X_0, \ldots, X_r)$ be a monotone expansion on G. It is enough to prove that, for any $i = 1, \ldots, r$ there exists a monotone inert edge search $\mathcal{S}_i$ for X_i using $\leq k$ searchers. Notice that this is trivial when $i = 1$. We assume now that $\mathcal{S}_{i-1}$ is a monotone inert edge search for X_{i-1} using $\leq k$ searchers, i.e., $F_{i-1} = \overline{X}_{i-1}$. We construct a new strategy $\mathcal{S}_i$ for X_i as follows. We initialize $\mathcal{S}_i$ so that it is the same as $\mathcal{S}_{i-1}$. Let S be the last set of $\mathcal{S}_{i-1}$. Let $\{q_1, \ldots, q_\sigma\}$ be an arbitrary ordering of S. We extend $\mathcal{S}_i$ by concatenating the following sequence of sets: $(\{q_2, \ldots, q_\sigma\}, \{q_3, \ldots, q_\sigma\}, \ldots, \{q_\sigma\}, \emptyset\})$ (i.e. we remove, one by one, all the searchers in S). Notice that the new sequence is a search as each new set added has one element less than the one before it. Moreover, the new sets are all subsets of S and therefore, they all have cardinality at most k. Finally, as $S \subseteq X_{i-1}$, the free locations corresponding to the new sets are all equal to F_{i-1}, i.e., no recontamination happens. We further enhance $\mathcal{S}_i$ by concatenating the following sequence of sets: $(\{s_1\}, \{s_1, s_2\}, \ldots, \{s_1, \ldots, s_{\tau-1}\}, \rho(X_i))$. As $\mathcal{E}$ is monotone, we have that $\overline{X}_i - \{p_i\} = \overline{X}_{i-1}$. Notice now that $\rho(X_i) \subseteq X_i - \{p_i\} = X_{i-1}$ and, as before, we have that the occurring sequence is a search, the new sets have all cardinality at most k and the free locations corresponding to the new sets are all equal to F_{i-1}, i.e., no recontamination happens.

We now examine two cases. In case $E_i = \emptyset$, we concatenate to $\mathcal{S}_i$ the set $R \cup \{p_i\}$. It is clear that $\mathcal{S}_i$ remains a search using $\leq k$ searchers. Moreover, $F_i = F_{i-1} - \{p_i\}$ as any path in G connecting vertices in $\overline{F}_{i-1} = \overline{X}_{i-1} = \overline{X}_i \cup \{p_i\}$ with p_i should contain a vertex in $\rho(X_i) = \partial_{E_i} V(C^{p_i}[\overline{X}_i \cup \{p_i\}])$, i.e., no recontamination happens.

It remains to examine the case where $E_i = \{\{p'_i, p_i\}\}$. If $p'_i \in \rho(X_i)$, this case is identical to the previous one. If $p'_i \notin \rho(X_i)$, we concatenate to $\mathcal{S}_i$ the set $\rho(X_i) \cup \{p'_i\}$. As $p'_i \in N_{G[X_i]}(p_i) \subseteq X_i - \{p_i\} = X_{i-1}$, and $|\rho(X_i)| = \phi(X_i) - 1 = \leq k - 1$, we easily conclude that $\mathcal{S}$ remains a monotone search using $\leq k$ searchers. Finally, we further concatenate to $\mathcal{S}$ the set $\rho(X_i) \cup \{p_i\}$.

Notice that $(\rho(X_i) \cup \{p'_i\}) \bigtriangledown (\rho(X_i)) \cup \{p_i\}) = \{p'_i, p_i\} \in E(G)$ and therefore $\mathcal{S}$ is again a search. It is also obvious that $|\rho(X_i) \cup \{p_i\}| \leq k$. Finally, we have that $F_i = F_{i-1} - \{p_i\}$ as any path in G avoiding $\{p'_i, p_i\}$ and connecting vertices in $\overline{F}_{i-1} = \overline{X}_{i-1} = \overline{X}_i \cup \{p_i\}$ with p_i should contain a vertex in $\rho(X_i) = \partial_{E_i} V(C^{p_i}[\overline{X}_i \cup \{p_i\}])$, i.e., no recontamination happens.

In both cases, we have that $F_i = F_{i-1} - \{p_i\} = \overline{X}_{i-1} \cup \{p_i\} = \overline{X}_i$ and $\mathcal{S}_i$ is a monotone inert edge search $\mathcal{S}_i$ for X_i using $\leq k$ searchers. □

Now, combining Lemmata 1, 2 and 3 we have the following theorem:

Theorem 1 *Monotonocity does not help in the inert mixed search.*

Notice that the monotonicity of the inert node search is a consequence of the monotonicity of inert mixed search and the fact that the inert node search can be reduced to the inert mixed search using the following easy lemma:

Lemma 4 *If G^n is the graph resulting after replacing every edge in G with two edges in parallel, then* $\text{ins}(G) = \text{ims}(G^n)$.

In the case where we are restricted to graphs without multiple edges, we can instead define G^n in the above lemma as the graph resulting after replacing every edge in G with two paths of length 2 connecting its endpoints. We mention that a direct consequence of Lemma 4 and the the monotonicity of inert mixed search is that computing $\text{ims}(G)$ (or equivalently, $\text{ies}(G)$) is a NP-complete problem.

Finally, the inert mixed search number, as well as all the other search parameters, is closed under taking of minors. This follows by the monotonicity of inert mixed search and the following lemma.

Lemma 5 *Let G be a graph and $\mathcal{S}$ be a monotone inert mixed search on G that uses $\leq k$ searchers. Then, the graph obtained after the removal or the contraction of any edge e of G has a monotone inert mixed search that uses $\leq k$ searchers.*

PROOF. The case where H is the result of the removal of e follows from the easy observation that $\mathcal{S} = (S_0, \ldots, S_r)$ is monotone inert mixed search of H as well. Suppose now that H is the result of the contraction of e. Let v be the endpoint of e that appears first in some set in in $\mathcal{S}$ and let u be the other, i.e. $\mathcal{S}$ clears first u and then v. W.l.o.g. we assume that the result of the contraction of e is v and we form a search $\mathcal{S}' = (S'_0, \ldots, S'_r)$ of H where, for $i = 0, \ldots, r$, $S'_i = S_i - \{u\}$. It is now easy to see that $\mathcal{S}'$ is a monotone inert mixed search of H that uses $\leq k$ searchers. □

3 Proper-treewidth

In this section we introduce the parameter of proper-treewidth and we will prove a series of lemmas able to prove its equivalence with the inert mixed search number.

A chordal graph is any graph with no induced cycle of length ≥ 4. A chordal graph is called K_{k+1}-chordal, $k \geq 1$, if its maximum size clique contains $k+1$ vertices. Alternatively, we can define K_{k+1}-chordal graphs in the following recursive way: A $(k+1)$-clique is a K_{k+1}-chordal graph; the graph obtained from a K_{k+1}-chordal graph if we introduce a new vertex and connect it with $\leq k$ vertices of one of its cliques is also a K_{k+1}-chordal graph. If in the above definition we replace "$\leq$" with "$=$", then we define the class of k-trees. We also define the treewidth of a graph G as the least k for which G is a subgraph of a k-tree. We will need the following known (and easy) theorem about the relation of k-trees and K_{k+1}-chordal graphs (for standard definitions and results about chordal graphs see [10]).

Theorem 2 *The k-trees are exactly the k-connected K_{k+1}-chordal graphs.*

Given two graphs G_1, G_2 and two cliques K_1, K_2 of G_1 and G_2 respectively where $|V(K_1)| = |V(K_2)| \leq k$, we define the *$k$-clique sum of G_1 and G_2* the graph obtained if we take the disjoint union of G_1 and G_2, and we identify the vertices and the edges of K_1 with the vertices and the edges of K_2. We will need the following known result (see [25], pages 328–329).

Theorem 3 *A graph is a K_{k+1}-chordal iff it can be obtained after applying a sequence of k-clique sums on a series of cliques of size at most $k+1$.*

Let G be a K_{k+1}-chordal graph. We call a clique in G *simplicial* when it contains a simplicial vertex of G. We will call the cliques of G of size $\leq k$ *small* and the cliques of size $k+1$ *big*. We call a small clique of G *separating* if its vertex set is a minimal separator of G. We call an edge of G *linking*, when it does not belong to any separating clique of G. We call a big clique K of G *proper* if it contains some linking edge.

We omit the proofs of the two following lemmas as they are direct consequences of the definitions.

Lemma 6 *All the simplicial big cliques of G are proper.*

Lemma 7 *Let G be a K_{k+1}-chordal graph and $e = \{v_1, v_2\}$ a linking edge of some $(k+1)$-clique of it. Then, e is not induced by the vertices of any minimal separator of G and there exists no other $(k+1)$-clique that contains e.*

We call a K_{k+1}-chordal graph *proper* if all of its $(k+1)$-cliques are proper. We call a graph *partial proper k-tree*, or equivalently we say that it has proper-treewidth $\leq k$, if it is a subgraph of a proper k-tree.

We say that a graph H is a vertex subgraph of G if $V(G) = V(H)$ and $E(H) \subseteq E(G)$.

Lemma 8 *Any graph G with $\text{ims}(G) \leq k$ is a subgraph of a proper K_{k+1}-chordal graph.*

PROOF. From Lemma 1 and Theorem 1 there exists a monotone expansion $\mathcal{E} = (X_0, \ldots, X_r)$ on G that uses $\leq k$ guards. We set $\{p_i\} = X_i - X_{i-1}, 1 \leq i \leq r$. For any $i, 1 \leq i \leq r$, let $\mathcal{C}(X_i) = \{\partial V(C) \mid C$ is a connected component of $G[\overline{X}_i]\}$. We set $S_i = \partial V(C^{p_{i+1}}[\overline{X}_i]), 0 \leq i \leq r-1$ and we define the sequence $\bar{G}_1, \ldots, \bar{G}_r$ as follows: $\bar{G}_1 = G[\{p_1\}]$, and $\bar{G}_i = (V(\bar{G}_{i-1}) \cup \{p_i\}, E(\bar{G}_{i-1}) \cup E_i)$ where E_i contains the edges connecting the vertices of S_{i-1} with p_i. We claim that for any $j, 1 \leq j \leq r$,

(i) $\bar{G}_j$ is a K_{k+1}-chordal graph that contains $G[X_j]$ as a vertex subgraph.
(ii) All the members of $\mathcal{C}(X_j)$ induce cliques in $\bar{G}_j$ of size $\leq k$.
(iii) $\mathcal{E}$ is an expansion of $\tilde{G}_i = G \cup \bar{G}_i$ that uses $\leq k$ guards.

Clearly, (i) and (ii) hold for $j = 1$. We assume that they hold for any $j, 1 \leq j < i \leq r$ and we will prove that they hold for $j = i$ as well.

From (i), we have that $\bar{G}_{i-1}$ is a K_{k+1}-chordal graph and contains $G[X_{i-1}]$ as a vertex subgraph. Notice that $S_{i-1} = \partial V(C^{p_i}[\overline{X}_{i-1}]) \in \mathcal{C}(X_{i-1})$ and, from (ii), S_{i-1} induces in $\bar{G}_{i-1}$ a clique of size $\leq k$. By construction, $\bar{G}_i$, is a chordal graph of maximum clique size $\leq k+1$ that contains $G[X_i]$ as a vertex subgraph.

It remains to prove that all the members of $\mathcal{C}(X_i)$ induce cliques in $\bar{G}_i$ of size $\leq k$. This is a direct consequence of (ii) in the special case where $V(C^{p_i}[\overline{X}_{i-1}]) = \{p_i\}$, as $\mathcal{C}(X_i) = \mathcal{C}(X_{i-1}) - \{N_G(p_i)\}$ (notice that $G[\overline{X}_i]$ has the same connected components as $G[\overline{X}_{i-1}]$ except from $G[\{p_i\}]$).

We can now assume that $V' = V(C^{p_i}[\overline{X}_{i-1}]) - \{p_i\}$ is not empty. Let $\mathcal{C}^{p_i}(X_i) = \{\partial V(C) \mid C$ is a connected component of $G[V']\}$. Clearly, the connected components of $G[V']$ are the only connected components in $\mathcal{C}(X_i)$ that are not also connected components in $\mathcal{C}(X_{i-1})$. Therefore, $\mathcal{C}^{p_i}(X_i) = \mathcal{C}(X_i) - \mathcal{C}(X_{i-1})$, and it is enough to prove that, each member of $\mathcal{C}^{p_i}(X_i)$ induces a clique in $\bar{G}_i$ of size $\leq k$. Notice that each member of $\mathcal{C}^{p_i}(X_i)$ is a subset of $\partial V'$. Therefore, it is enough to prove that $\partial V'$ induces a clique in $\bar{G}_i$ of size $\leq k$. Recall that $\rho(X_i) \leq k-1$. This means that either $\rho(X_i) = S_{i-1}$ or that there is a vertex $a \in S_{i-1}$ such that $\rho(X_i) = S_{i-1} - \{a\}$ and such that, apart from p_i, there is no other vertex in $\overline{X}_{i-1} \supseteq V'$ connected with a in G (clearly in this case

$E_i = \{\{a, p_i\}\})$. In the first case we obtain that $\partial V' \subseteq S_{i-1} \cup \{p_i\}$ and thus $|\partial V'| \leq k$. In the second case, $\partial V' \subseteq S_{i-1} \cup \{p_i\} - \{a\}$ and thus $|\partial V'| \leq k$. In both cases, S_{i-1} induces a clique in $\bar{G}_{i-1}$ and therefore, $\partial V'$ induces a clique in $\bar{G}_i$ of size $\leq k$.

In order to prove (iii) we claim that $\rho_{\tilde{G}_i}(X_h) = \rho_{\tilde{G}_{i-1}}(X_h), 1 \leq h \leq r$ (the index on function ρ denotes the graph on which it is defined). Clearly, $\rho_{\tilde{G}_{i-1}}(X_h) \subseteq \rho_{\tilde{G}_i}(X_h), 1 \leq h \leq r$ as $\tilde{G}_{i-1}$ is a vertex subgraph of $\tilde{G}_i$. From the induction hypothesis, we have that $\rho_{\tilde{G}_i}(X_h) \subseteq \rho_{\tilde{G}_{i-1}}(X_h), i < h \leq r$. Let v be a vertex in $\rho_{\tilde{G}_i}(X_h)$ for some $1 \leq h \leq i$. There exists a path P in $\tilde{G}_i$ that starts from v, such that all its vertices after v belong in $\overline{X}_{h-1}$, and finishes on p_h. Clearly, if P exists also in $\tilde{G}_{i-1}$, we have that $v \in \rho_{\tilde{G}_{i-1}}(X_h)$ as well and thus $\rho_{\tilde{G}_i}(X_h) \subseteq \rho_{\tilde{G}_{i-1}}(X_h)$. If P is not a path in $\tilde{G}_{i-1}$ then some of the edges of this path are edges in $E' = E(\tilde{G}_i) - E(\tilde{G}_{i-1})$ or edges connecting S_{i-1} with p_i that do not belong in $\tilde{G}_{i-1}$. We call these edges *critical* (notice that, as all the edges in E' are incident to p_i, the critical edges can be at most two). Observe now that the existence of some critical edge $\{b, p_i\}$ in $\tilde{G}_i$ implies the existence of a path in G (and thus in $\tilde{G}_{h-1}$ as well) that starts from b, has all the following vertices in $\overline{X}_{i-1}$ (and thus in $\overline{X}_{h-1}$) and finishes on p_i. Resuming, there exists a path in $\tilde{G}_{h-1}$ that starts on a, all the following vertices are in $\overline{X}_{h-1}$, and finishes in v. This means that $v \in \rho_{\tilde{G}_{i-1}}(X_h)$ as well. Therefore, $\rho_{\tilde{G}_i}(X_h) \subseteq \rho_{\tilde{G}_{i-1}}(X_h), 1 \leq h \leq i$ and this completes the proof of the claim.

According to the claims (i) – (iii) above, G is a vertex subgraph of the K_{k+1}-chordal graph $\bar{G} = \bar{G}_r = \tilde{G}_r$. Moreover, $\mathcal{E}$ is an expansion on $\bar{G}$ that uses $\leq k$ guards. It now remains to show that any big clique in $\bar{G}$ is proper.

Suppose in contrary that there exists a k-clique K in $\bar{G}$ that is not proper. From Lemma 6, K will be a non-simplicial clique and, as K is not proper, all of its edges are inducing edges of separating cliques (i.e. it does not have linking edges). As $\mathcal{E}$ is a monotone expansion on $\bar{G}$ that uses $\leq k$ guards, Lemma 3 implies that there exists a monotone search $\mathcal{S} = \{S_0, \ldots, S_t\}$ on $\bar{G}$ that uses at most k searchers. We call *appearance number* of a vertex v of K, the smallest $i, 1 \leq i \leq t$ for which $v \in S_i$. Let now u be the vertex of $V(K)$ with the biggest appearance number (intuitively u is the vertex of K that is cleared last). Let i be the appearance number of u. We first exclude that $S_i = S_{i-1} \cup \{u\}$. Indeed, in this case, $|S_{i-1}| \leq k-1$. Therefore, there exists an edge from u to a vertex in $V(K)$ that is not in $|S_{i-1}|$ and his will cause recontamination. The only remaining case is when $G[S_i \bigtriangledown S_{i-1}]$ induces a single edge $\{v, u\}$ in G. W.l.o.g. we assume that $u \in S_i$. Notice that, if there exists any other path connecting u and v, whose internal vertices are not in $V(K)$ we have recontamination. Notice now that, as K is not proper, e is not a linking edge and therefore, its endpoints are vertices of a separating clique of K. This implies the existence of a path connecting v and u whose internal vertices are not in K, a contradiction. Therefore all the big cliques of $\bar{G}$ are proper and the lemma holds. □

We will now prove that the proper K_{k+1}-chordal graphs can be viewed as vertex subgraphs of proper k-trees.

Lemma 9 *Let G be a K_{k+1}-chordal graph and K a big clique of G. Let also E be the set of linking edges of K. If K contains a minimal separator S of size $< k$ then there exists a vertex $v \in V(K) - S$ such that there exists an edge in E not induced by $S \cup \{v\}$.*

PROOF. Let $e = \{w, u\}$ be a linking edge of K. From Lemma 7, we have that e is not induced by S and thus at least one of its endpoints is not in S. If $e \cap S = \emptyset$, then clearly for any choice of $v \in V(K) - S$, $S \cup \{v\}$ cannot induce e. Suppose now that one, say w, of w, u is in S. Notice that $V(K) - (S \cup \{u\})$ is non empty as $|V(K)| = k + 1$ and $|S| < k$. Therefore, for any choice of $v \in V(K) - (S \cup \{u\})$, $S \cup \{v\}$ cannot induce e. □

Lemma 10 *For any $r, 1 \leq r \leq k - 1$, any r-connected proper K_{k+1}-chordal graph is a vertex subgraph of a $(r + 1)$-connected proper K_{k+1}-chordal graph.*

PROOF. We apply the following procedure as long as there exists in G a separator S of size r:

> Let C be one of the connected components of $G[V(G) - S]$. Let $G_1 = G[V(C) \cup V(S)]$ and $G_2 = G[V(G) - V(C)]$. Let $K_i, i = 1, 2$ be a maximal clique in G_i that contains S as a subclique. We distinguish 3 cases: (a) If both $K_i, i = 1, 2$ are small, then add in G an edge that connects any vertex v_1 of K_1 with any vertex v_2 of K_2. (b) If one, say K_1, of $K_i, i = 1, 2$ is small, then add in G an edge that connects any vertex v_1 of K_1 with a vertex v_2 of K_2 such that $\{v_2\} \cup S$ is not inducing all the linking edges of K_2. (c) If both $K_i, i = 1, 2$ are big cliques, then add in G an edge $\{v_1, v_2\}$ where, for $i = 1, 2$, $\{v_i\} \cup S$ is not inducing all the linking edges of K_i.

As, the input of the above transformation is a proper K_{k+1}-chordal graph, the transformation is always doable as long as a separator S of size r exists because of Lemma 9. We now claim that the property of the proper K_{k+1}-chordality is an invariant of the above transformation. Let G be an input and let G be an output of the above transformation. Observe that G' is a K_{k+1}-chordal graph as it is the clique sum of three K_{k+1}-chordal graphs: G_1, the clique formed if we add in $G[S \cup \{v_1, v_2\}]$, edge $\{v_1, v_2\}$, and G_2 (notice that $|S \cup \{v_1, v_2\}| \leq k + 1$). It remains to prove that all the big cliques of G' are proper. Clearly, if the newly appearing clique is a big clique then it should be proper as, by construction, it contains $\{v_1, v_2\}$ as linking edge. It remains now to see that each one of the old big cliques of G' still have at least one linking edge. Clearly, a linking edge of a big clique has some chance of losing this status after the above transformation, only in cases (b) and (c) where K_2 (in case (b)) and $K_i, i = 1, 2$ (in case (c)) are big cliques. We examine first case (b). Clearly, a clique that could lose all its linking edges after the above transformation is K_2 as all the edges induced by $S \cup \{v\}$ are edges of K_2. Moreover, no other big clique can loose any of its linking edges because no linking edge is shared by two big cliques (Lemma 7). Notice now that, from Lemma 9 the choice of v makes it so that there will be always a linking edge in K not induced by $S \cup \{v\}$, thus K is proper. The case (c) is very similar.

Clearly, the above transformation will not be further applicable only if the resulting graph does not contain any minimal separator of size r. Therefore, it will finally produce a $(r+1)$-connected proper K_{k+1}-chordal graph as required.
□

Applying now inductively Lemma 10 and using Theorem 2, we have the following result.

Lemma 11 *Any proper K_{k+1}-chordal graph is a vertex subgraph of a proper k-tree.*

A corollary of Lemmata 8 and 11 is the following.

Lemma 12 *For any graph G,* $\text{proper-treewidth}(G) \leq \text{ims}(G)$.

4 The parameter la

In this section we will prove that the inert mixed search number and the proper-treewidth are equal to the graph theoretic parameter la, defined as the last k for which a graph is a minor of the product $T \times K_k$ for some tree T.

The graph $T \times K_k$ for a tree T with n vertices, is the graph obtained by replacing each tree vertex with a copy of a k-labeled clique K_k and joining with an edge two vertices of different cliques if they have the same label and their corresponding tree vertices were previously adjacent in T. $\text{la}(G)$ is a graph theoretic parameter introduced in [26] and [5] and used for studying invariants of graphs related with the multiplicities of eigenvalues of elliptic self-adjoint differential operators.

Lemma 13 *For any graph G,* $\text{la}(G) \leq \text{proper-treewidth}(G)$.

PROOF. We have to prove that any proper k-tree H is a minor of $T \times K_k$ for some tree T. We introduce first some terminology for the products of the form $T \times K_k$. We distinguish the edges of $T \times K_k$ as follows. If they are edges of k-cliques we call them *strong*. Otherwise we call them *weak*.

Let S be a big clique of H. Notice that in the non-trivial case where the cardinality of the vertex set of H is at least $k+2$, S contains at least one and at most two separating cliques (in case H is a k-clique, the result is obvious). Let S_1 be the one of them and if there exists a second one we denote it as S_2, otherwise we consider as S_2 an arbitrary k-clique of S that is different than S_1. Observe now that the $(k+1)$-clique $G[S]$ is isomorphic to a graph obtained by $K_2 \times K_k$ after a sequence of contractions applied to all the weak edges except from one. We rename the vertices of the two k-cliques in $K_2 \times K_k$ so that they are S_1 and S_2. We also assume that the unique weak edge of $K_2 \times K_k$ that is not contracted is the one connecting the vertices in the symmetric difference of S_1 and S_2. We call M_S, the renamed version of $K_2 \times K_k$ corresponding to S. Clearly, by undoing the contractions following the inverse order, $G[S]$ can be transformed to M_S. We call this process *inverse contractions for S*. It is now easy to verify that if we apply the inverse contractions for any big clique in G, we will obtain a graph isomorphic to $T \times K_k$ for some tree T and the result follows. □

Lemma 14 *For any tree T, $T \times K_k$ has a monotone inert mixed search that uses k searchers.*

PROOF. We prove the following stronger statement: for any tree T, $T \times K_k$ has a monotone inert mixed search $\mathcal{S}$ that uses k searchers and such that for any $v \in V(G)$ the vertices of the clique – we denote it by K^v – corresponding to v, appear as a set of $\mathcal{S}$. The proof uses induction on the number of edges of T. If $E(T) = \emptyset$, then the result is obvious. If $E(T) = \{e\}$ where $e = \{v, u\}$ then we set $V(K^v) = \{v_1, \ldots, v_k\}$, $V(K^u) = \{u_1, \ldots, u_k\}$ and the required search is the concatenation of $\mathcal{S}_{u+}\ \mathcal{S}_{u \to v}\ \mathcal{S}_{v-}$ where

$$\mathcal{S}_{u+} = (\emptyset, \{u_1\}, \{u_1, u_2\}, \ldots, V(K^u) - \{u_k, u_{k-1}\}, V(K^u) - \{u_k\}, V(K^u)),$$

$$\mathcal{S}_{u \to v} = (V(K^u) \cup \{v_1\} - \{u_1\}, V(K^u) \cup \{v_1, v_2\} - \{u_1, u_2\}, \ldots, V(K^v) \cup \{u_k\} - \{v_k\})$$ and

$$\mathcal{S}_{v-} = (V(K^v), V(K^v) - \{v_k\}, V(K^v) - \{v_k, v_{k-1}\}, \ldots, \{v_1, v_2\}, \{v_1, \}, \emptyset).$$

Suppose now that the claim is correct for $|E(T)| < n$ and let T be a tree where $|E(T)| = n$. Let v be a leaf of T and let u be its unique neighboring vertex. Let $T' = T(V(T) - \{v\})$. From the induction hypothesis, $T' \times K_k$ has a monotone inert mixed search $\mathcal{S}'$ that uses k searchers and such that $V(K_u) = S_i$ where $\mathcal{S}' = (S_0, \ldots, S_i, \ldots, S_r)$. We set $\mathcal{S}_1 = (S_0, \ldots, S_i)$ and $\mathcal{S}_2 = (S_{i+1}, \ldots, S_r)$. It is now enough to see that the required search of $T \times K_k$ is the concatenation, in series, of $\mathcal{S}_1$, $\mathcal{S}_{u \to v}$, $\mathcal{S}_{v-}$, $\mathcal{S}_{u+}$, and $\mathcal{S}_2$. $\square$

A consequence of Lemmata 14 and 5 is the following.

Lemma 15 *For any graph G,* $\mathrm{ims}(G) \leq \mathrm{la}(G)$.

From Lemmata 12,13, and 15 we conclude the following.

Theorem 4 *For any graph G,* $\mathrm{ims}(G) = \text{proper-treewidth}(G) = \mathrm{la}(G)$.

5 Conclusions

The equivalence between the inert mixed search number and la fits in a more general framework of relations between search parameters and "width" type parameters. In particular, we proved that the graphs that are subgraphs of proper k-trees are exactly the graphs with inert mixed search number at most k. It is known that if we alter the restriction of properness we have the parameter of inert node search number which can be, in turn, reduced to the inert mixed search number using the transformation of Lemma 4. In this way, it appears that the problem of computing the treewidth of a graph (or, equivalently, the inert node search number) can be reduced to the problem of computing the proper-treewidth (or, equivalently, the la). The "path" counterpart of this relation has already been revealed by A. Takahashi, S. Ueno, and Y. Kajitani, in [24] where they define the parameter of proper-pathwidth [1] as an equivalent parameter to the agile mixed search number. According to their results, a loopless graph without multiple edges has agile mixed search number at most k iff it has proper-pathwidth $\leq k$ or, equivalently, if it is a subgraph of a k-path (a k-path can be viewed as a k-tree that either has $\leq k+1$ vertices or it has only two simplicial vertices). Moreover, it is known (see [12,17,13]) that the inert node search number of a loopless graph without multiple edges is the least k for which a graph is a subgraph of a k-caterpillar (a k-caterpillar is a k-tree that is also an interval graph – for definitions and results on k-caterpillars see e.g. [19]). Finally, it is possible to prove that if we modify the definition of la so that the tree involved is simply a path, then we have its "path" counterpart, equivalent to proper-pathwidth and the agile mixed search number. The proof of the equivalence of this modified la and the proper-pathwidth is mainly based on the "path" analogues of Lemmata 13 and 14. The proofs of those lemmata are just simplified versions of their "tree" counterparts, proved in this paper,

[1] We point out that the parameter of proper-pathwidth defined in [24] is different from the parameter of proper-pathwidth (equivalent to bandwidth) defined by Kaplan and Shamir in [11].

and are omitted. We summarize the whole landscape of equivalences between parameters in the following table:

	Node search	Mixed search
Agile	(a) pathwidth$\leq k$ (b) subgraphs of k-caterpillars	(a) proper-pathwidth (b) subgraphs of k-paths (c) minors of $T \times K_k$ where T is a line
Inert	(a) treewidth$\leq k$ (b) subgraphs of k-trees	(a) proper-treewidth (b) subgraphs of proper k-trees (c) minors of $T \times K_k$ where T is a tree

6 Open problems

The first search game concerning a "tree" type parameter was given by Seymour and Thomas in [22] and was equivalent to treewidth. In the setting of that game the fugitive was agile but "visible", i.e. the searchers' moves depend on the knowledge of the fugitive's moves. Clearly, according to the equivalence of treewidth and node search for an "invisible but inert" fugitive, proven in [6], these two games are equivalent. We believe that it is possible to define an "agile but visible" equivalent of the inert mixed search as well. However, it is an open problem whether the obstruction type characterization of treewidth, given in [22], can be extended to one of proper treewidth. An additional feature of the "invisible but inert" framework introduced in [6] and is adopted in this paper, is that it can be parameterized in terms of the speed of the fugitive, i.e. the length of a maximum unguarded path that the inert fugitive can cross during some "threatening" move (a move of type (i) or (ii)). In this direction, it appears challenging to extend the results of [6] for the parametrized version of the (more general) inert mixed search.

An interesting variant of any search game can be defined when the search number of a search $\mathcal{S} = (S_0, \ldots, S_r)$ is given by $\sum_{1\leq i\leq r} |S_i|$ instead of the classical $\min_{0\leq i\leq r} |S_i|$. The monotonicity question, as well as the identification of known graph theoretical parameters connected with these search variants, appears to be a challenging problem. In this direction, there are results only for the case of agile node search. In particular, Fomin and Golovach in [8] prove the monotonicity of the agile node search variant as well as its equivalence with the interval graph completion problem. We believe that our results, combined with the results in [8], can produce results related to other search variants as well.

The fact that the inert mixed search number is closed under minors, indicates (see e.g. [20]) that, for any fixed k, there exists a polynomial time algorithm deciding, for any given graph G, whether $\mathrm{ims}(G) \leq k$ or not. However such an algorithm is unknown for $k \geq 3$. A solution for $k = 2$ is straightforward as it can be easily seen that $\mathrm{ims}(G) \leq 2$ iff G does not contain K_4 or K_3^n as a minor. However, such an obstruction-based characterization for bigger values of k appears to be a difficult open problem.

References

[1] D. Bienstock. Graph searching, path-width, tree-width and related problems (a survey). *DIMACS Ser. in Discrete Mathematics and Theoretical Computer Science*, 5:33–49, 1991.

[2] D. Bienstock and P. Seymour. Monotonicity in graph searching. *J. Algorithms*, 12:239 – 245, 1991.

[3] H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comp. Sc.*, 209:1–45, 1998.

[4] R. Breisch. An intuitive approach to speleotopology. *A publication of the Southwestern Region of the National Speleological Society*, VI:72–78, 1967.

[5] Y. C. de Verdière. Multiplicities of eigenvalues and tree-width of graphs. *Journal of Combinatorial Theory, Series B*, 74:121–146, 1998.

[6] N. D. Dendris, L. M. Kirousis, and D. M. Thilikos. Fugitive-search games on graphs and related parameters. *Theor. Comp. Sc.*, 172:233–254, 1997.

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

[8] F. Fomin and P. Golovach. Interval completion via graph searching. Manuscript, submitted to SIAM Journal on Computing, 1999.

[9] F. Fomin and N. Petrov. Pursuit-evasion and search poblems on graphs. *Congressus Numerantium*, 122:47–58, 1996.

[10] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York, 1980.

[11] H. Kaplan and R. Shamir. Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques. *SIAM J. Comput.*, 25:540–561, 1996.

[12] N. G. Kinnersley. The vertex separation number of a graph equals its path width. *Inform. Proc. Letters*, 42:345–350, 1992.

[13] L. M. Kirousis and C. H. Papadimitriou. Interval graphs and searching. *Disc. Math.*, 55:181–184, 1985.

[14] L. M. Kirousis and C. H. Papadimitriou. Searching and pebbling. *Theor. Comp. Sc.*, 47:205–218, 1986.

[15] A. S. LaPaugh. Recontamination does not help to search a graph. *J. ACM*, 40:224–245, 1993.

[16] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. ACM*, 35:18–44, 1988.

[17] R. H. Möhring. Graph problems related to gate matrix layout and PLA folding. In E. Mayr, H. Noltemeier, and M. Sysło, editors, *Computational Graph Theory, Computing Suppl. 7*, pages 17–51. Springer Verlag, 1990.

[18] T. D. Parsons. Pursuit evasion in a graph. In Y. Alavi and D. R. Lick, editors, *Theory and Application of Graphs*, pages 426–441, Berlin, 1976. Springer Verlag.

[19] A. Proskurowski. Separating subgraphs in k-trees: Cables and caterpillars. *Disc. Math.*, 49:275–285, 1984.

[20] N. Robertson and P. D. Seymour. Graph minors — a survey. In I. Anderson, editor, *Surveys in Combinatorics*, pages 153–171. Cambridge Univ. Press, 1985.

[21] P. D. Seymour and R. Thomas. Graph searching and a minimax theorem for tree-width. *J. Comb. Theory Series B*, 58:239–257, 1993.

[22] P. D. Seymour and R. Thomas. Graph searching and a minimax theorem for tree-width. *J. Comb. Theory Series B*, 58:239–257, 1993.

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

[24] A. Takahashi, S. Ueno, and Y. Kajitani. Mixed-searching and proper-path-width. *Theor. Comp. Sc.*, 137:253–268, 1995.

[25] C. Thomassen. Embeddings and Minors, *in Handbook of Combinatorics, ed. R. L. Graham, M. Grötschel and L. Lovasz, North Holland*, 1990.

[26] H. van der Holst. *Topological and Spectral Sraph Characterizations.* PhD thesis, Amsterdam University, 1996.