Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Discrete Applied Mathematics 127 (2003) 565–580
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

On the domination search number ☆

Fedor V. Fomin[a,b,*], Dieter Kratsch[c], Haiko Müller[d]

[a] *Heinz Nixdorf Institut, University of Paderborn, 33102 Paderborn, Germany*
[b] *Faculty of Mathematics and Mechanics, St. Petersburg State University, St. Petersburg, Russia*
[c] *Université de Metz, Laboratoire d'Informatique Théorique et Appliquée, 57045 Metz Cedex 01, France*
[d] *School of Computing, University of Leeds, Leeds, LS2 9JT, UK*

Received 18 April 2000; received in revised form 17 May 2001; accepted 4 March 2002

Abstract

We introduce the domination search game which can be seen as a natural modification of the well-known node search game. Various results concerning the domination search number of a graph are presented. In particular, we establish a very interesting connection between domination graph searching and a relatively new graph parameter called dominating target number.

Keywords: Graph searching; Domination; Asteroidal triple-free graph; Dominating target

1. Introduction

Graph searching problems have attracted the attention of researchers from Discrete Mathematics and Computer Science for a variety of nice and unexpected applications in different and seemingly unrelated fields. There is a strong resemblance of graph searching to certain pebble games [21] that model sequential computation. Other applications of graph searching can be found in the VLSI theory: the game-theoretic approach to some important parameters of graph layouts such as the cutwidth [30], the topological

☆ Most of this research was done during a visit of FVF at the F.-Schiller-Universität Jena which was supported by a fellowship of the DAAD (Kennziffer: A/99/09594). FVF acknowledges support by EC contract IST-1999-14186, Project ALCOM-FT (Algorithms and Complexity—Future Technologies) and by RFBR grants N98-01-00934, N01-01-00235.

* Corresponding author.
E-mail addresses: fomin@uni-paderborn.de (F.V. Fomin), kratsch@lita.sciences.univ-metz.fr (D. Kratsch), hm@comp.leeds.ac.uk (H. Müller).

0166-218X/03/$ - see front matter
PII: S0166-218X(02)00389-X

bandwidth [29], the bandwidth [14], the profile [15], and the vertex separation number [13] is very useful for the design of efficient algorithms. Also let us mention the connection between graph searching, pathwidth, and treewidth. These parameters play a very important role in the theory of graph minors developed by Robertson and Seymour [4,12,36]. Some search problems also have applications in problems of privacy in distributed environments with mobile eavesdroppers ('bugs') [18].

In this paper, we introduce a domination search game which can be regarded as a natural generalization of the well-known node search game (the formal definitions are given in Section 2). In this version of searching at every step some searchers are placed or are removed from vertices of a graph G. The purpose of searching is to find an invisible and fast fugitive moving from vertex to vertex in G. In the node search problem the searchers find the fugitive if some of them succeeded to occupy the same vertex as the fugitive. In the domination search problem the searchers have more power, they find the fugitive if one of them can 'see' it, i.e. the fugitive stands on a vertex of the closed neighborhood of a vertex occupied by the searcher. Similar 'see-catch' problems on graphs with searchers having 'radius of capture' in different graph metrics were studied in [33,34,16]. Some variants of 'see-catch' pursuit-evasion games on grids were studied in [37], see [17] for a survey.

Recently 'see-catch' search problems in polygonal environments attracted much attention in Computational Geometry and Robotics. This variation of searching was introduced by Suzuki and Yamashita in [38]. More on polygon searching problems with different variations can be found in [19,26,11,27]. The study of these problems is motivated by robotic applications, such as surveillance with a mobile robot equipped with a camera that must find a moving target in a cluttered workspace. Domination searching can also be regarded as a natural transformation of polygon searching problems into graph searching.

The paper is organized as follows. In Section 2 we give definitions and preliminaries. In Section 3 we establish relations between the domination search game and the well-known node search game. In this section we also observe some complexity results. In Section 5 we discuss upper bounds for the domination search number that can be obtained by making use of spanning trees. The last section contains the main theorem of the paper which settles a very interesting connection between domination graph searching and a relatively new graph parameter called dominating target number. The theorem gives an upper bound for the domination search number of a connected graph in terms of its dominating target number.

2. Preliminaries

We use standard graph-theoretic terminology compatible with [6], to which we refer the reader for basic definitions. $G=(V,E)$ is an undirected, simple (without loops and multiple edges) and finite graph with the vertex set V and the edge set E. We denote by $G[W]$ the subgraph of $G=(V,E)$ induced by $W \subseteq V$.

As customary we consider *connected components* (or short *components*) of a graph as maximal connected subgraphs as well as vertex subsets. A vertex set $S \subseteq V$ of a

graph G is said to be *connected* if the subgraph of G induced by S is connected. We denote by $\bar{G}$ the complement of a graph G. The (*open*) *neighborhood* of a vertex v is $N(v)=\{u\in V\colon \{u,v\}\in E\}$ and the *closed neighborhood* of v is $N[v]=N(v)\cup\{v\}$. For a vertex set $S\subseteq V$ we put $N[S]=\bigcup_{v\in S} N[v]$ and $N(S)=N[S]\setminus S$. A vertex set $D\subseteq V$ of a graph $G=(V,E)$ is said to be a *dominating set* of G if for every vertex $u\in V\setminus D$ there is a vertex $v\in D$ such that $\{u,v\}\in E$. Thus D is a dominating set iff $N[D]=V$. The minimum cardinality of a dominating set of a graph G is denoted by $\gamma(G)$. We also say that $A\subseteq V$ *dominates* $B\subseteq V$ in the graph $G=(V,E)$ if $B\subseteq N[A]$.

The *distance* $d_G(u,v)$ between two vertices u and v of G is the length of the shortest path between u and v in the graph G. For a graph $G=(V,E)$ let G^k be the graph with vertex set V and two vertices u and v are adjacent in G^k if and only if $d_G(u,v)\leqslant k$.

For our purpose it is convenient to describe the graph searching in terms of clearing graph vertices. Initially, all vertices are contaminated (uncleared or are occupied by invisible fugitive). A contaminated vertex is cleared once after placing a searcher on a vertex from its closed neighborhood. A clear vertex v is recontaminated if there is a path avoiding closed neighborhoods of vertices occupied by searchers leading from v to a contaminated vertex.

More precisely. A *domination search program* Π on a graph $G=(V,E)$ is a sequence of pairs (also considered as the *steps* of Π)

$$(D_0,A_0),(D_1,A_1),\ldots,(D_{2m-1},A_{2m-1})$$

such that

(1) for all $i\in\{0,1,\ldots,2m-1\}$, $D_i\subseteq V$ and $A_i\subseteq V$;
(2) $D_0=\emptyset$, $A_0=\emptyset$;
(3) for every $i\in\{1,2,\ldots,m\}$ at the $(2i-1)$th step we *place new searchers and clear vertices*: $D_{2i-2}\subset D_{2i-1}$ and $A_{2i-1}=A_{2i-2}\cup N[D_{2i-1}]$;
(4) for every $i\in\{1,2,\ldots,m-1\}$ at the $2i$th step we *remove searchers with possible recontamination*: $D_{2i-1}\supset D_{2i}$ and A_{2i} is the subset of A_{2i-1} satisfying that for every vertex $v\in A_{2i}$ every path containing v and a vertex from $V\setminus A_{2i-1}$ contains a vertex from $N[D_{2i}]$. If $A_{2i-1}\setminus A_{2i}\neq\emptyset$ we say that the vertices of $A_{2i-1}\setminus A_{2i}$ are *recontaminated* at the $2i$th step and that *recontamination occurs* at the $2i$th step.

It is useful to consider D_i as the set of vertices occupied by searchers and A_i as the set of *cleared vertices after the ith step*. Hence $V\setminus A_i$ is the set of *contaminated vertices* after the ith step.

Notice that a program $\Pi=((D_0,A_0),(D_1,A_1),\ldots,(D_{2m-1},A_{2m-1}))$ is fully determined by the sequence $(D_0,D_1,\ldots,D_{2m-1})$ thus we shall mainly use the shorter description $\Pi=(D_0,D_1,\ldots,D_{2m-1})$.

All proofs in our paper can be given in terms of the formal definition of a search program $\Pi=(D_0,D_1,\ldots,D_{2m-1})$ but often we prefer a more 'informal' (but equivalent) way of proving.

We call a domination search program Π *winning* if $A_{2m-1}=V$ (i.e. all vertices are cleared). A search program $\Pi=(D_0,D_1,\ldots,D_{2m-1})$ is *monotone* if $A_i\subseteq A_{i+1}$ for all $i\in\{0,1,\ldots,2m-2\}$, i.e. no recontamination occurs. We say that a domination search program Π uses k searchers if $\max_{i\in\{0,\ldots,2m-1\}}|D_i|=k$. We define the *domination*

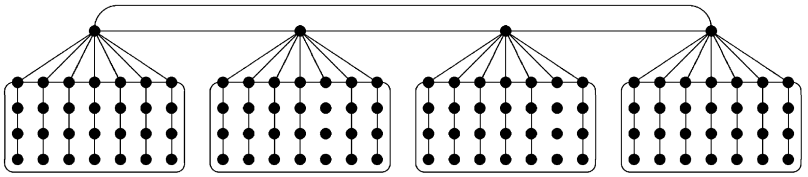

Fig. 1. The Dobrev example.

search number by

$$\mathrm{ds}(G) := \min_{\Pi} \max_{i \in \{0,\ldots,2m-1\}} |D_i|,$$

where the minimum is taken over all winning programs Π.

Problem 1. *Is there a graph G such that every monotone winning domination search program on G uses more than* $\mathrm{ds}(G)$ *searchers*? *In other words, does recontamination help to search a graph*?

This question was recently answered by Dobrev who constructed the graph G depicted in Fig. 1. A search program clearing G by only two searchers is easy to find. However, in each such search program one vertex in the 4-cycle is recontaminated. To see the latter fact observe that the only way to clear a 7-wheel by two searchers is to place one of them on the central vertex.

The domination search number of a disconnected graph G is equal to the maximum domination search number taken over all components of G. Hence from now on we assume that all considered graphs are connected.

There is an obvious relation between domination search number and domination number: $\mathrm{ds}(G) \leqslant \gamma(G)$. There is also a more interesting connection. Let $G=(V,E)$ be a graph and $S_2(G)=(V(S_2(G)),E(S_2(G)))$ be the graph obtained from G by replacing every edge $\{u,v\}$ of G by an (u,v)-path of length three. We call the vertices of $V \subseteq V(S_2(G))$ *original* vertices and the vertices of $V(S_2(G)) \setminus V$ *middle* vertices.

Theorem 2. *For any graph $G=(V,E)$, let H be the graph with vertex set $V(S_2(G))$ and edge set $E \cup E(S_2(\bar{G}))$, i.e. H is obtained from G by connecting every two nonadjacent vertices by a path of length three. Then*

$$\gamma(G) \leqslant \mathrm{ds}(H) \leqslant \gamma(G)+1.$$

Proof. Let D be a dominating set in G. We first put $|D|$ searchers on the vertices of D in H and then clear all middle vertices of $S_2(\bar{G})$ by one additional searcher. Hence $\mathrm{ds}(H) \leqslant \gamma(G)+1$.

Let us prove now that $\gamma(G)-1$ searchers cannot clear the graph H by showing that at every step of searching there exists a contaminated original vertex. In fact, let $\Pi=(D_0,D_1,\ldots,D_{2m-1})$ be a domination search program on the graph H using at most $\gamma(G)-1$ searchers. $|D_1| \leqslant \gamma(G)-1$ implies that there is at least one contaminated original vertex after the first step of Π. If Π is winning then there is an $i \in \{1,2,\ldots,m\}$

such that after the $(2i-1)$th step all original vertices are cleared and no original vertex is contaminated until the end of Π. W.l.o.g. we may assume that at the $(2i-1)$th step a contaminated original vertex v is cleared. From $|D_{2i-1}| \leqslant \gamma(G)-1$ we conclude that there is an original vertex $w \notin N[D_{2i-1}]$ such that either w is adjacent to v (of course this implies that $v \notin D_{2i-1}$) or there is a (v,w)-path of length 3 in H with two interior middle vertices such that no vertex of this path belongs to $N[D_{2i-2}]$. In both cases w is contaminated after the $(2i-1)$th step. This contradiction concludes the proof. □

3. Node search number

The domination search game can be regarded as a generalization of the well known *node search game*, see [4] for a survey. In the node search game at every step some searchers are placed on or removed from vertices. Initially, all vertices are contaminated (uncleared). The difference between node and domination searching is that a contaminated vertex v in node searching is cleared once after placing a searcher on v. A clear vertex v is recontaminated if there is a path avoiding vertices occupied by searchers leading from v to a contaminated vertex. More formally, a *node search program* Π on a graph $G=(V,E)$ is a sequence of pairs

$$(D_0,B_0),(D_1,B_1),\ldots,(D_{2m-1},B_{2m-1})$$

such that

(1) for all $i \in \{0,1,\ldots,2m-1\}$, $D_i \subseteq V$ and $B_i \subseteq V$;
(2) $D_0=\emptyset$, $B_0=\emptyset$;
(3) for every $i \in \{1,2,\ldots,m\}$ at the $(2i-1)$th step we *place new searchers and clear vertices*: $D_{2i-2} \subset D_{2i-1}$ and $B_{2i-1}=B_{2i-2} \cup D_{2i-1}$;
(4) for every $i \in \{1,2,\ldots,m-1\}$ at the $2i$th step we *remove searchers with possible recontamination*: $D_{2i-1} \supset D_{2i}$ and B_{2i} is the subset of B_{2i-1} satisfying that for every vertex $v \in B_{2i}$ every path containing v and a vertex from $V \setminus B_{2i-1}$ contains a vertex from D_{2i}. Vertices of $B_{2i-1} \setminus B_{2i}$ are said to be *recontaminated*.

Analogously to domination searching, a node search program Π is *winning* if $B_{2m-1}=V$ (all vertices are cleared). A node search program $\Pi = ((D_0,B_0),(D_1,B_1),\ldots,(D_{2m-1},B_{2m-1}))$ is *monotone* if $B_i \subseteq B_{i+1}$ for all $i \in \{0,1,\ldots,2m-2\}$. The number of searchers used in this program is $\max_{i\in\{0,\ldots,2m-1\}} |D_i|$.

The smallest number of searchers that are necessary to be used in a winning node search program on a graph G is denoted by $\mathrm{ns}(G)$ and is said to be the *node search number* of G.

Using results of La Paugh [25], Kirousis and Papadimitriou [21] obtained the following fundamental result about node searching (see also [5] for an alternative proof).

Theorem 3 (Kirousis–Papadimitriou). *For any graph G, there exists a* monotone *search program using* $\mathrm{ns}(G)$ *searchers, i.e. G can be cleared by* $\mathrm{ns}(G)$ *searchers without recontamination of previously cleared vertices.*

Theorem 4. *For any graph* $G=(V,E)$, $\mathrm{ns}(G)=\mathrm{ds}(S_2(G))$.

Proof. Let us prove that $\mathrm{ns}(G) \geqslant \mathrm{ds}(S_2(G))$. Let $\Pi = (D_0, D_1, \ldots, D_{2m-1})$ be a monotone node search program on G. We claim that Π is also a monotone domination search program on $S_2(G)$. Assume the converse and suppose that the first recontamination in domination searching occurs when we remove a searcher from a vertex v. Since Π is monotone for node searching, all neighbors of v in G are cleared before this step. Moreover, every neighbor u of v in G was visited by a searcher. Since no recontamination occurs before removing the searcher from v in $S_2(G)$, all vertices of every (u,v)-path in $S_2(G)$ are also cleared. Therefore no neighbor of v in $S_2(G)$ is adjacent to an uncleared vertex and no recontamination occurs when removing the searcher from v.

To prove $\mathrm{ns}(G) \leqslant \mathrm{ds}(S_2(G))$ we consider a winning domination search program Π on $S_2(G)$. We simulate Π by a winning node search program Π_G on G as follows. If a searcher is removed from $S_2(G)$ in Π then we remove it from G. If a searcher in Π is placed on an original vertex v of $S_2(G)$ then we place this searcher on v in Π_G. If a searcher in Π is placed on a middle vertex v of $S_2(G)$ then we place this searcher in Π_G on the original vertex adjacent to v. Notice that there is only one such vertex.

Let A_i be the set of cleared vertices after the ith step of Π and B_i be the set of cleared vertices after the ith step of Π_G. We claim that for every $i \in \{0, 1, \ldots, 2m-1\}$,

$$A_i \cap V \subseteq B_i. \qquad (*)$$

This is obvious for $i = 0$. We proceed by induction assuming that for all $i < k$ the inclusion $(*)$ holds. If at the kth step searchers are placed on $V(S_2(G))$ then $(*)$ is clearly true. Suppose that at the kth step searchers are removed and that $A_k \cap V \nsubseteq B_k$. Then there is a vertex $v \in V$ that is recontaminated at the kth step in Π_G but is cleared in Π. Therefore after the $(k-1)$th step in Π_G there is a contaminated vertex u and there is also a (u,v)-path P in G containing no vertex of D_k.

By assumption, u is contaminated in Π after the $(k-1)$th step. It can be seen easily that the (u,v)-path P in G that does not contain vertices occupied by searchers after the kth step of Π_G can be transformed into an (u,v)-path P' in $S_2(G)$ by replacing each edge $\{a,b\}$ in the path P by its (a,b)-path of length 3 with two interior middle vertices. Therefore no vertex of the path P' in $S_2(G)$ is 'seen' by a searcher in Π. Consequently v is contaminated after the kth step in Π. This contradiction proves $(*)$.

Since Π is winning, we obtain that after step $2m-1$ of Π all original vertices of $S_2(G)$ are cleared. Then by $(*)$, Π_G is also winning. □

Monien and Sudborough [31] have shown that a variant of node searching, namely edge searching, is NP-hard even for planar graphs with vertex degree at most three. Since the transformation of Kirousis and Papadimitriou from node to edge search problems and our transformation in Theorem 4 from domination search to node search problems preserve planarity and degree constraints, we have the following.

Corollary 5. *The problem* Domination Searching: '*Given a graph G and an integer k, decide whether* $\mathrm{ds}(G) \leqslant k$ *or not*' *is* NP*-hard even for planar graphs with vertex degree at most three.*

Furthermore, unless $P = \mathrm{NP}$, there is no polynomial time algorithm approximating the domination number by a constant factor [2,28] or by a factor of $c \log n$ for some $c > 0$ [35]. Combined with Theorem 2 this implies

Corollary 6. *There is a constant $c > 0$ such that there is no polynomial time algorithm to approximate the domination search number of a graph within a factor of $c \log n$ unless $P = \mathrm{NP}$.*

Problem 7. *Is it true that for every graph G with n vertices $\mathrm{ds}(G)$ searchers can clear G in $O(n)$ steps?*

4. Graphs with large search numbers

Using the relation between the search numbers we are able to find graphs with large domination search numbers.

We need the following technical lemma.

Lemma 8. *For every graph G, $\mathrm{ns}(G) \leqslant \Delta(G)\,\mathrm{ds}(G) + 1$.*

Proof. In fact, let Π be a domination search program using k searchers. This program leads to the node search program Π_Δ using $k \cdot \Delta(G) + 1$ searchers: when we place a searcher on a vertex v in Π, in Π_Δ we place searchers on all vertices of $N[v]$ and then remove the searcher from v. When we remove a searcher in Π from a vertex v then in Π_Δ we remove the searchers from all vertices of $N(v)$ that are not adjacent to a vertex occupied by some other searcher (at this step). Obviously, if Π is a winning domination search program then Π_Δ is a winning node search program and the lemma follows. □

Let B^k be the k-dimensional cube, i.e. the graph with the vertex set consisting of all binary vectors in $\mathbb{R}^k$; two vertices u, v in this graph are adjacent if and only if $\|u - v\| = 1$, where $\|x\|$ is the number of nonzero entries in x (the Hamming norm of x). For $S \subseteq B^k$ we define

$$\partial(S) := \{u \in S\text{: there exists a } w \in B^k \setminus S \text{ such that } u \text{ is adjacent to } w\}.$$

A vertex set $A \subseteq B^k$ is *optimal* if $|\partial A| \leqslant |\partial B|$ for each $B \subseteq B^k$ of cardinality $|A|$. Define a vertex ordering $L = \{v_1, v_2, \ldots, v_{2^k}\}$ of B^k as follows. Vertex v_i precedes v_j on L if and only if $\|v_i\| < \|v_j\|$ or $\|v_i\| = \|v_j\|$ and v_i precedes v_j lexicographically. Harper [20] has shown that for each i the set $\{v_1, v_2, \ldots, v_i\}$ is optimal (see [3] for a survey).

Lemma 9. $\mathrm{ns}(B^k) \geqslant \binom{k}{k/2} + 1$.

Proof. Harper's result implies that for every subset S with $\sum_{i=1}^{k/2} \binom{k}{i}$ vertices $|\partial(S)| \geqslant \binom{k}{k/2}$ since the first $\sum_{i=1}^{k/2} \binom{k}{i}$ vertices in the ordering L form the $k/2$-dimensional ball

in B^k. The number of 'border vertices' in this ball is equal to $\binom{k}{k/2}$ and by Harper's result this ball is the optimal set.

Consider a monotone winning node search program Π on B^k. Let j be the step of Π at which the set S of cardinality $\sum_{i=1}^{k/2}\binom{k}{i}$ is cleared. Then every vertex of $\partial(S)$ contains a searcher at this step. Since every vertex of $\partial(S)$ is adjacent to a vertex from $B^k \setminus S$ and the program is monotone, we have that at the $(j+1)$th step no searcher can be removed from $\partial(S)$. Hence Π uses at least $\binom{k}{k/2}+1$ searchers at the $(j+1)$th step. □

Theorem 10. *For every $\epsilon > 0$ and every integer m there exists a graph G on $n \geqslant m$ vertices such that* $\mathrm{ds}(G) = \Omega(n^{1-\epsilon})$.

Proof. For $\epsilon > 0$ we choose k such that $2^{k\epsilon} \geqslant k^{3/2}$. Let G be the k-dimensional cube B^k with $n = 2^k$ vertices. By the previous lemma $\mathrm{ns}(G) \geqslant \binom{k}{k/2}+1$. Applying Stirling's formula $k! \sim \sqrt{2\pi k}(k/e)^k$ we obtain $\mathrm{ns}(G) \geqslant 2^k/\sqrt{k}+1$.

By Lemma 8, $\mathrm{ns}(G) \leqslant \Delta(G)\,\mathrm{ds}(G)+1$. Therefore,

$$\mathrm{ds}(G) \geqslant \frac{2^k}{\sqrt{k}}\frac{1}{k} = \frac{2^k}{k^{3/2}} \geqslant \frac{2^k}{2^{k\epsilon}} = n^{1-\epsilon}. \qquad \square$$

We do not know whether there is a graph class satisfying $\mathrm{ds}(G) = \Omega(n)$ and we leave that question open.

5. Spanning trees

Let $\delta = (v_1, v_2, \ldots, v_n)$ be a vertex ordering of a graph $G = (V, E)$. The width of the ordering δ of G is

$$\mathrm{bw}(G, \delta) := \max\{|i-j|\colon \{v_i, v_j\} \in E\},$$

and the *bandwidth* of G is

$$\mathrm{bw}(G) := \min\{\mathrm{bw}(G, \delta)\colon \delta \text{ is a vertex ordering of } G\}.$$

In the proof of the next theorem we use a theorem of Ando et al. [1] stating that for any tree T with l leaves $\mathrm{bw}(G) \leqslant \lceil l/2 \rceil$. We denote the set of leaves, i.e. vertices of degree 1, of a tree T by $V_1(T)$.

Theorem 11. *Let $T = (V, E(T))$ be a spanning tree of a graph $G = (V, E)$ such that $G \subseteq T^{k+1}$ and let l be the number of leaves of the tree $T_1 = T - V_1(T)$. Then*

$$\mathrm{ds}(G) \leqslant \lceil \tfrac{l}{2} \rceil (k+1) + 1.$$

Proof. Let $\delta = (v_1, v_2, \ldots, v_r)$ be an ordering of the vertices of T_1 such that $bw(T_1, \delta) \leqslant b = \lceil l/2 \rceil$. Note that such an ordering exists by [1].

Our domination search program works as follows. We put searchers on the first $b(k+1)+1$ vertices of δ. Then we remove the searcher from v_1 and place it on

$v_{b(k+1)+2}$. Suppose that after removing the searcher from v_1 recontamination occurs. Hence there is a vertex x such that $\{x, v_1\} \in E$ and there is a contaminated vertex y such that $\{x, y\} \in E$. Since searchers occupy all vertices with indices at most $bk+1$ unless v_1, there is a vertex v_i, $i > b(k+1)+1$, such that $y \in N_T[v_i]$. Now $G \subseteq T^{k+1}$ implies that the (x, y)-path in T has length at most $k+1$.

We distinguish three cases. In Case 1 we assume that x is a leaf of T and $\{x, v_1\} \in E(T)$. Then the length of the (v_1, v_i)-path in T is at most k since the (x, y)-path in T passes through v_1. In Case 2 we assume that x is a leaf of T and $\{x, v_1\} \notin E(T)$. Hence the neighbor of x in T is a vertex v_j for some $j > b(k+1)+1$. $G \subseteq T^{k+1}$ implies $d_T(v_1, x) \leqslant k+1$. Hence the length of the (v_1, v_j)-path in T is at most k since the (v_1, x)-path in T passes through v_j. Finally in Case 3 we assume that x is not a leaf of T. Hence $x = v_j$ for some $j > b(k+1)+1$. Therefore $\{v_1, v_j\} \in E$ which implies $d_T(v_1, v_j) \leqslant k+1$.

Therefore in all three cases there is a $j > b(k+1)+1$ such that $d_T(v_1, v_j) \leqslant k+1$. By the pigeonhole principle there is an edge $\{v_p, v_q\}$ in the (v_1, v_j)-path of T such that $|p-q| > b$ which contradicts the choice of ordering δ.

By the same arguments we can remove a searcher (without recontamination) from v_2 and put it on $v_{b(k+1)+3}$, and so on. Finally, every vertex of G is cleared once by a searcher because T is a spanning tree of G. Since no recontamination occurs when a searcher is placed on v_n all vertices of G are cleared. □

If the spanning tree T is a caterpillar then the results of Theorem 11 can be slightly improved. A *caterpillar* is a tree which consists of a path, called the *backbone*, and leaves adjacent to vertices of the backbone.

Theorem 12. *Let T be a spanning caterpillar of a graph G and let k be an integer such that G is a subgraph of T^{k+1}. Then* $\mathrm{ds}(G) \leqslant \max\{2, k\}$.

Proof. The backbone $P = (v_1, v_2, \ldots, v_m)$ of T is a dominating path of G, i.e., every vertex of G is adjacent to a vertex of P. First we suppose $k \geqslant 2$. Consider the following domination search program Π using k searchers. Initially we put searchers on the first k vertices of P. Then we remove the searcher from v_1 and put it on v_{k+1}, then remove searcher from v_2 and put it on v_{k+2} and so on.

Let us show that Π is a monotone program. Assume the converse. Suppose that after removing a searcher from a vertex, say v_j, the first recontamination occurs. Then there is a vertex x such that $\{x, v_j\} \in E$ and $\{x, y\} \in E$ for some contaminated vertex y. Because y is not cleared yet, we conclude that $y \in N_T[v_i]$ for some $i > j+k-1$. Moreover, if $y = v_i$ then $i > j+k$. Therefore the length of the (x, y)-path in T is at least $k+2$. But this contradicts the definition of T; hence Π is monotone. Since every vertex of G was once cleared by a searcher we obtain that Π is winning.

If $k \leqslant 1$ then $G \subseteq T^{k+1} \subseteq T^2 \subseteq T^3$ and two searchers are sufficient to clear G as shown above. □

An independent set of three vertices is called an *asteroidal triple* if every two of them are connected by a path avoiding the neighborhood of the third. A graph is *AT-free* if it

does not contain an asteroidal triple. Asteroidal triples were introduced to characterize interval graphs and comparability graphs, see [7] for references. In their fundamental paper [9] Corneil et al. investigate AT-free graphs. Among others properties they prove that every connected AT-free graph contains a *dominating pair*, i.e. a pair of vertices u and v such that every (u,v)-path is dominating. A dominating pair of a connected AT-free graph can be detected by a simple linear time algorithm, called 2LexBFS [10]. In [24] it was shown how to use a dominating path found by 2LexBFS as backbone of a caterpillar T such that $T \subseteq G \subseteq T^4$. We obtain the following corollary of Theorem 12.

Corollary 13. *Let G be an AT-free graph. Then* $\mathrm{ds}(G) \leqslant 3$.

We could not find an AT-free graph with domination search number 3. Moreover, we state the following conjecture.

Conjecture 14. *Let G be an AT-free graph. Then* $\mathrm{ds}(G) \leqslant 2$.

For some classes of AT-free graphs we are able to prove that the domination search number of every graph in this class is at most two.

A graph G is a comparability graph if and only if G has a transitive orientation of its edges. Cocomparability graph are the complements of comparability graphs. Every interval graph is a cocomparability graph, and every cocomparability graph is AT-free, see [7] for references and a survey on different graph classes.

Theorem 15. *Let G be a cocomparability graph. Then* $\mathrm{ds}(G) \leqslant 2$.

Proof. There is an ordering $(v_1, v_2, \ldots, v_n)$ of the vertices of a cocomparability graph $G = (V,E)$ such that $i < j < k$ and $\{v_i, v_k\} \in E$ implies that v_j is adjacent to v_i or v_k, see, e.g. [7].

Let us describe a winning program which uses two searchers. First we place one searcher on v_1 and one on the rightmost neighbor of v_1, say v_i. Now every vertex in the interval $[v_1, v_i] = \{v_k\colon 1 \leqslant k \leqslant i\}$ is cleared by these two searchers. Then we remove the searcher from v_1. We claim that no recontamination occurs. In fact, if a vertex v_k, $1 < k < i$ is adjacent to a contaminated vertex v_l, $l > i$, then v_l is not adjacent to v_i (otherwise v_l is cleared by the searcher on v_i). Hence $v_k \in N[v_i]$ which means that v_k is 'seen' by the searcher on v_i. Hence no recontamination occurs.

Thus we can place the searcher removed from v_1 on the rightmost neighbor of v_i. Then we remove the searcher from v_i (i.e. the leftmost vertex occupied by a searcher) and place it on the rightmost neighbor of the only vertex occupied by a searcher. We repeat these actions until the searchers reach a vertex, say v_j, without right neighbor.

Now one searcher is placed on v_j and the other one is removed from G. If $v_j = v_n$ then all vertices are cleared and searching was successful. Otherwise we place a searcher on v_{j+1} and remove then the searcher from v_j. We claim that no vertex is recontaminated after removing the searcher from v_j. Indeed, let v_l be adjacent to v_j. Then $l < j$. Suppose that v_l is adjacent to a contaminated vertex v_p. Then $p > l$ and v_p is not

adjacent to v_{j+1}. Consequently $\{v_l, v_{j+1}\} \in E$ and v_l is cleared by the searcher on v_{j+1}. Hence no recontamination occurs.

Then we again place a searcher on the rightmost neighbor until a vertex without right neighbor is reached, etc. When one of the searchers is placed on v_n all vertices are cleared. □

A graph isomorphic to $K_{1,3}$ is referred to as a *claw*, and a graph that does not contain an induced claw is said to be *claw-free*.

Corollary 16. *The domination search number of AT-free claw-free graphs is at most two.*

Proof. As it was observed by Kloks et al. in [22] every connected AT-free claw-free graph G is a claw-free cocomparability graph or is a complement of a triangle-free graph. If a graph G is a cocomparability graph then by Theorem 15 it can be searched by two searchers. Otherwise $\bar{G}$ is triangle-free which implies $\gamma(G) \leqslant 2$ and $\mathrm{ds}(G) \leqslant 2$. □

Parra and Scheffler [32] proved that for every AT-free claw-free graph G, $\mathrm{ns}(G) - 1 = \mathrm{bw}(G)$. Combining this with Corollary 16 one can obtain the following interesting result.

Corollary 17. *For any AT-free claw-free graph G,*

$$\tfrac{1}{2}\Delta(G) \leqslant \mathrm{bw}(G) \leqslant 2\Delta(G) - 1,$$

where $\Delta(G)$ denotes the maximum degree of a vertex in G.

Proof. It is well-known and easy to see that for every graph G, $\Delta(G) \leqslant 2\mathrm{bw}(G)$, see e.g. [8].

For every AT-free claw-free graph G, $\mathrm{bw}(G) = \mathrm{ns}(G) - 1$ and by Lemma 8, $\mathrm{ns}(G) \leqslant \Delta(G) \cdot \mathrm{ds}(G) + 1$ holds.

In the winning search program on cocomparability graphs given in Theorem 15 we put searchers at distance at most 2 from each other. Using this fact one can observe that $\mathrm{ns}(G) - 2 \leqslant \Delta(G) \cdot \mathrm{ds}(G)$ on cocomparability graphs. The same is true for graphs with triangle-free complement. By Corollary 16, $\mathrm{ds}(G) \leqslant 2$ and we obtain $\mathrm{bw}(G) \leqslant 2\Delta(G) - 1$. □

6. Dominating targets

We start with a result on graphs with dominating pair, a class of graphs containing as we have mentioned all connected AT-free graphs. Let us recall the definition of a dominating pair. A *dominating pair* is a pair of two (not necessarily different) vertices u and v of a connected graph G such that the vertex set of every (u, v)-path in G is a dominating set of G.

Lemma 18. *The domination search number of a connected graph with dominating pair is at most* 4.

Proof. Let d_1, d_2 be a dominating pair in G and let P be a shortest (d_1, d_2)-path. Then P is the backbone of a spanning caterpillar T in G. Since P is a shortest path, we have that $G \subseteq T^5$. Then by Theorem 12, $\mathrm{ds}(G) \leqslant 4$. □

Problem 19. *Determine the maximum domination search number of a connected graph with dominating pair.*

Dominating targets have been introduced in [23] as a generalization of dominating pairs. A *dominating target* is a vertex set $T \subseteq V$ of a connected graph $G = (V, E)$ such that every connected superset of T is a dominating set. The *dominating target number* of a graph G, denoted by $\mathrm{dt}(G)$, is the smallest size of dominating target of G. Hence graphs with dominating pair have dominating target number at most 2. The following theorem that extends Lemma 18 is one of the main results of our paper. To prove it we shall need some technical results about dominating targets as well as a notion of a dominating target for a vertex set of a graph.

A vertex set $T \subseteq V$ of a graph $G = (V, E)$ is a *dominating target for a vertex set* $B \subseteq V$ *in* G if $B \subseteq N[S]$ for every connected superset S of T. Clearly T is a dominating target of a graph $G = (V, E)$ if and only if T is a dominating target for V in G.

Lemma 20. *Let T be a dominating target for a vertex set $B \subseteq V$ of a graph $G=(V,E)$ and let D be a set of vertices. Then for any vertex $v \in B \setminus N[T \cup D]$ the following two statements hold*:

(1) *The number of connected components of $G - N[v]$ containing vertices of T is at least two.*

(2) *Let $C_1, C_2, \ldots, C_k$ be the connected components of $G - N[v]$ with $T \cap C_i \neq \emptyset$. For every $i \in \{1, 2, \ldots, k\}$ let $y_i \in N[v]$ be a vertex with $N[y_i] \cap C_i \neq \emptyset$. Then for every $i \in \{1, 2, \ldots, k\}$ the set $T_i = (T \cap C_i) \cup \{v\}$ is a dominating target for the vertex set $B_i = (B \cap C_i) \setminus N[D_i]$ in the graph $G[C_i \cup N[v]]$, where $D_i = (D \cap (C_i \cup N[v])) \cup \{v\} \cup \{y_j : j \neq i\}$.*

Proof. Let $v \in B \setminus N[T \cup D]$. Suppose there is a connected component C of $G - N[v]$ with $T \subseteq C$. Then C is a connected superset of T that does not dominate the vertex v—a contradiction. Hence at least two components of $G - N[v]$ contain vertices of T.

For the proof of the second statement suppose that T_i is not a dominating target for B_i in the graph $G[C_i \cup N[v]]$. Then there is a connected superset $S_i \subseteq C_i \cup N[v]$ of T_i and a vertex $w \in B_i$ that has no neighbor in S_i. We extend S_i to a connected superset S of T in G by adding for every $j \neq i$ the vertex y_j and (for simplicity) C_j. Hence $w \in B$ is not adjacent to a vertex of S which implies that T is not a dominating target for B in G—a contradiction. □

Theorem 21. *For every connected graph G, $\mathrm{ds}(G) \leqslant 2\mathrm{dt}(G) + 3$.*

Proof. By induction on $k=|T|$, we prove the following stronger statement that implies the theorem when taking a dominating target T of G such that $|T|=\mathrm{dt}(G)$, i.e. $B=V$ and $D=\emptyset$.

Let T be a dominating target for $B \subseteq V$ in the graph $G=(V,E)$ and let $D \subseteq V$ be a vertex set such that $N[D] \supseteq V \setminus B$. Then there is a winning domination search program Π on G using at most $|D|+2|T|+3$ searchers such that at the first step of Π searchers are placed on all vertices of D and after that on each vertex of D there is a searcher throughout all steps of the search program Π.

If T is a dominating set of $G[B]$ then $T \cup D$ is a dominating set of G. Hence the statement is true since we simply place searchers on all vertices of $T \cup D$. Notice that $|T|=1$ implies that T is a dominating set of $G[B]$. If $|T|=2$ then we place a searcher on each vertex of D and by Lemma 18, we can clear all vertices of B using four additional searchers to be moved along a shortest path P between the two vertices of T in G.

Suppose inductively that for all $D \subseteq V$ and all dominating targets T for B in G with $N[D] \supseteq V \setminus B$ the statement is true if $|T| \leqslant k-1$. Let T be a dominating target for $B \subseteq V$ in the graph $G=(V,E)$ with $|T|=k$ and let $D \subseteq V$ be a vertex set such that $N[D] \supseteq V \setminus B$.

We may assume that $T \cup D$ is a not a dominating set of $G[B]$, thus $S=B \setminus N[T \cup D] \neq \emptyset$. Then by the first part of Lemma 20, for every vertex $v \in S$ at least two components of $G-N[v]$ contain vertices of T.

Consider first the easy case in which there is a vertex $v \in S$ such that every component of $G-N[v]$ contains at least 2 vertices of T. Let $C_1, C_2, \ldots, C_m$, $m \geqslant 2$, be the components of $G-N[v]$ with $T \cap C_i \neq \emptyset$ such that $k-2 \geqslant |C_1 \cap T| \geqslant |C_2 \cap T| \geqslant \cdots \geqslant |C_m \cap T| \geqslant 2$. For every $j \in \{1,2,\ldots,m\}$ choose a vertex $y_j \in N[v]$ with $N[y_j] \cap C_j \neq \emptyset$.

The domination search program works as follows. We place searchers on all vertices of D and a searcher on v that will never be removed. Then the components $C_1, C_2, \ldots, C_m$ are cleared individually one by one, where each component C_i is cleared as follows:

Place searchers on all vertices y_j with $j \neq i$. By the second part of Lemma 20, we may conclude that $T_i = (T \cap C_i) \cup \{v\}$ is a dominating target for the vertex set $B_i = ((B \cap C_i) \setminus \bigcup_{j \neq i} N[y_j]) \setminus N[D_i]$ in the graph $G[C_i \cup N[v]]$ where $D_i = (D \cap (C_i \cup N[v_i])) \cup \{v\} \cup \{y_j : j \neq i\}$ satisfies $N[D_i] \supseteq C_i \setminus B_i$. Since $|C_i \cap T|+1 \leqslant k-1$ the graph $G[C_i \cup N[v]]$ can be searched (resp. C_i can be cleared) using $2(|C_i \cap T|+1)+3+|D_i|$ searchers by our inductive assumption.

Since the components are cleared individually and $|D_i| \leqslant |D|+1+(m-1)$ we have that the number of searchers needed is at most $|D|+m+2(|C_1 \cap T|+1)+3$. Each component C_i contains at least two vertices of T and we obtain $|C_1 \cap T|+m \leqslant k$. Combined with $|C_1 \cap T| \leqslant k-2$ this implies that the number of searchers is at most $|D|+2k-2+5=|D|+2k+3$.

Now we can concentrate our efforts on the hard case in which for every vertex $v \in S = B \setminus N[T \cup D]$ at least one component of $G-N[v]$ contains exactly one vertex of T.

We say that a vertex $v \in S$ is *t-separating* in G for a vertex $t \in T$ if there is a component C of $G-N[v]$ such that $C \cap T = \{t\}$.

Suppose that the dominating target T for B in G contains a vertex t for which no vertex $v \in B$ is t-separating. Then $T \setminus \{t\}$ is a dominating target for $B \setminus N[t]$ in the graph G. Thus placing a searcher on vertex t and then inductively using the claim for a dominating target of cardinality $k-1$, we easily obtain that $1+|D|+2k+1=|D|+2k+2$ searchers are sufficient to search G.

From now on we may assume that for every vertex $t \in T$ there is a vertex $v \in S$ which is t-separating. Notice that every vertex $v \in S$ is t-separating for some $t \in T$ (otherwise we are in the easy case) and that a vertex v may be t-separating for various vertices of T. It will be convenient to assume that $T = \{t_1, t_2, \ldots, t_k\}$.

For every $i \in \{1, 2, \ldots, k\}$, let $R(t_i)$ be the set of all vertices $v \in S$ that are t_i-separating. Notice that $R(t_i) \neq \emptyset$ for all $i \in \{1, 2, \ldots, k\}$ and that $\bigcup_{1 \leqslant i \leqslant k} R(t_i) = V \setminus N[D \cup T]$.

For vertices $a, b \in R(t_i)$ we say that $a \prec_i b$ if a and t_i are in one component of $G-N[b]$ but b and t_i are not in one component of $G-N[a]$. For every $i \in \{1, 2, \ldots, k\}$, we choose a $\prec_i$-maximal element v_i, i.e. a maximal element of the partially ordered set $(R(t_i), \prec_i)$. (Notice that $v_i = v_j$ for $i \neq j$ is possible.) Let C_i be the component of $G - N[v_i]$ containing t_i, i.e. $T \cap C_i = \{t_i\}$.

The domination search program works as follows. We place searchers on all vertices of D and on all vertices of $\{v_1, v_2, \ldots, v_k\}$ that will never be removed throughout the search.

Then the components $C_1, C_2, \ldots, C_k$ will be cleared individually one by one, where each component C_i is cleared as follows. Let $C_1^i, C_2^i, \ldots, C_{m_i}^i$ be all components of $G - N[v_i]$ containing a vertex of T except the component C_i, thus $m_i \leqslant k-1$. For each component C_j^i, $j \in \{1, 2, \ldots, m_i\}$, we choose a vertex $y_j^i \in N[v_i] \cap N(C_j^i)$. By the second part of Lemma 20 and similar to the easy case we may conclude that for every $i \in \{1, 2, \ldots, k\}$, $\{v_i, t_i\}$ is a dominating target for the vertex set $B_i = ((B \cap C_i) \setminus \bigcup_{j \neq i} N[y_j^i]) \setminus N[D_i]$ in the graph $G[C_i \cup N[v_i]]$ where $D_i = (D \cap (C_i \cup N[v_i])) \cup \{v_i\} \cup \{y_j^i : j \neq i\}$ satisfies $N[D_i] \supseteq C_i \setminus B_i$. We place searchers on all the vertices $y_1^i, y_2^i, \ldots, y_{m_i}^i$. Finally as consequence of Lemma 18 (already obtained at the beginning of this proof), four additional searchers are sufficient to clear C_i. Finally we remove all searchers from $y_1^i, y_2^i, \ldots, y_{m_i}^i$. Altogether this clears C_i.

Hence all components $C_1, C_2, \ldots, C_k$ can be cleared by using at most $|D| + k + (k-1) + 4 = |D| + 2k + 3$ searchers. After clearing the components $C_1, C_2, \ldots, C_k$ there remain $|D| + k$ searchers on G. The current set of cleared vertices is $N[D] \cup \bigcup_{1 \leqslant i \leqslant k} C_i \cup \bigcup_{1 \leqslant i \leqslant k} N[v_i]$. Since $N[t_i] \subseteq C_i$ and $\bigcup_{1 \leqslant i \leqslant k} R(t_i) = V \setminus N[T \cup D]$ we may conclude that all contaminated vertices belong to the set $U = (\bigcup_{1 \leqslant i \leqslant k} R(t_i) \setminus (N[v_i] \cup C_i)) \setminus N[D]$.

Let us show how the vertices from the set U can be cleared by placing k additional searchers on vertices of G. For every $i \in \{1, 2, \ldots, k\}$, we choose a vertex $u_i \in N[v_i]$ having a neighbor in C_i. We claim that every vertex of U is adjacent to a vertex of the set $\{u_1, u_2, \ldots, u_k\}$ which would imply that additional k searchers placed on all vertices of $\{u_1, u_2, \ldots, u_k\}$ together with the k searchers already placed on $\{v_1, v_2, \ldots, v_k\}$ will clear all vertices of U. In fact, choose a vertex $w \in U$.

Then $w \in R(l)$ for some $l \in \{1, 2, \ldots, k\}$, thus w is t_l-separating. Choose a (t_l, v_l)-path P having all vertices in C_l except u_l and v_l. Since v_l is a $\prec_l$-maximal element, v_l does not belong to the component of $G - N[w]$ containing t_l and therefore w is adjacent to

a vertex p of the path P. On the other hand, since $w \notin C_l \cup N[v_l]$ we may conclude that $p \notin C_l$. Therefore $p = u_l$ and the claim follows.

Finally we place searchers on all vertices of $\{u_1, u_2, \ldots, u_k\}$ thus clearing vertices from U by our above claim. This needs at most $2k + |D|$ searchers.

Summarizing, the number of searchers needed is at most $|D| + 2k + 3$. This completes the proof. □

Corollary 22. *For every disconnected graph* G, $\mathrm{ds}(G) \leqslant 3 + 2\max_C \mathrm{dt}(C)$, *where the maximum is taken over all components* C *of* G.

To illustrate the strength of Theorem 21 let us consider the complete graph G on n vertices and the graph $S_2(G)$ obtained from G by replacing each edge by a path of length 3. Clearly $\mathrm{dt}(S_2(G)) = n$ and by Theorem 4 we have $\mathrm{ds}(S_2(G)) = n$. This leads to the following conjecture which is up to our knowledge the strongest possible strengthening of our theorem.

Conjecture 23. $\mathrm{ds}(G) \leqslant \mathrm{dt}(G)$ *for all graphs* G.

Acknowledgements

The authors are grateful to Stefan Dobrev (Bratislava) for pointing out the example given in Fig. 1.

References

[1] K. Ando, A. Kaneko, S. Gervacio, The bandwidth of a tree with k leaves is at most $\lceil k/2 \rceil$, Discrete Math. 150 (1996) 403–406.

[2] M. Bellare, S. Goldwasser, C. Lund, A. Russell, Efficient probabilistically checkable proofs and applications to approximation, in: Proceedings of the Annual ACM Symposium on Theory of Computing, Vol. 25, The Association for Computing Machinery, 1993, pp. 294–304.

[3] S.L. Bezrukov, Isoperimetric problems in discrete spaces, in: Extremal Problems for Finite Sets (Visegrád, 1991), János Bolyai Math. Soc., Budapest, 1994, pp. 59–91.

[4] D. Bienstock, Graph searching, path-width, tree-width and related problems (a survey) DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 5 (1991) 33–49.

[5] D. Bienstock, P. Seymour, Monotonicity in graph searching, J. Algorithms 12 (1991) 239–245.

[6] J.A. Bondy, Basic graph theory: paths and circuits, in: R.L. Graham, M. Grötschel, L. Lovász (Eds.), Handbook of Combinatorics, Vol. 1, Elsevier Science, Amsterdam, 1995, pp. 3–110.

[7] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph classes: a survey, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[8] P.Z. Chinn, J. Chvátalová, A.K. Dewdney, N.E. Gibbs, The bandwidth problem for graphs and matrices—a survey, J. Graph Theory 6 (1982) 223–254.

[9] D.G. Corneil, S. Olariu, L.K. Stewart, Asteroidal triple-free graphs, SIAM J. Discrete Math. 10 (1997) 399–430.

[10] D.G. Corneil, S. Olariu, L.K. Stewart, Linear time algorithms for dominating pairs in asteroidal triple-free graphs, SIAM J. Comput. 28 (1999) 1284–1297.

[11] D. Crass, I. Suzuki, M. Yamashita, Searching for a mobile intruder in a corridor—the open edge variant of the polygon search problem, Internat. J. Comput. Geom. Appl. 5 (1995) 397–412.

[12] N.D. Dendris, L.M. Kirousis, D.M. Thilikos, Fugitive-search games on graphs and related parameters, Theoret. Comput. Sci. 172 (1997) 233–254.
[13] J.A. Ellis, I.H. Sudborough, J. Turner, The vertex separation and search number of a graph, Inform. Comput. 113 (1994) 50–79.
[14] F.V. Fomin, Helicopter search problems, bandwidth and pathwidth, Discrete Appl. Math. 85 (1998) 59–71.
[15] F.V. Fomin, P.A. Golovach, Interval completion and graph searching, SIAM J. Discrete Math. 13 (2000) 454–464.
[16] F.V. Fomin, P.A. Golovach, N.N. Petrov, Search problems on 1-skeletons of regular polyhedrons, Internat. J. Math. Game Theory Algebra 7 (1997) 101–111.
[17] F.V. Fomin, N.N. Petrov, Pursuit-evasion and search problems on graphs, Congressus Numerantium 122 (1996) 47–58.
[18] M. Franklin, Z. Galil, M. Yung, Eavesdropping games: a graph-theoretic approach to privacy in distributed systems J. ACM 47 (2000) 225–243.
[19] L.J. Guibas, J.-C. Latombe, S.M. Lavalle, D. Lin, R. Motwani, A visibility-based pursuit-evasion problem, Internat. J. Comput. Geom. Appl. 9 (1999) 471–493.
[20] L.J. Harper, Optimal numberings and isoperimetric problem on graphs, J. Combin. Theory 8 (1966) 385–393.
[21] L.M. Kirousis, C.H. Papadimitriou, Searching and pebbling, Theore. Comput. Sci. 47 (1986) 205–218.
[22] T. Kloks, D. Kratsch, H. Müller, Approximating the bandwidth of asteroidal-triple free graphs, Forschungsergebnisse Math/95/6, Friedrich Schiller Universität, Jena, Germany, 1995.
[23] T. Kloks, D. Kratsch, H. Müller, On the structure of graphs with bounded asteroidal number, Technical Report Math/Inf/97/22, Friedrich-Schiller-Universität, Jena, Germany, 1997.
[24] T. Kloks, D. Kratsch, H. Müller, Approximating the bandwidth of asteroidal-triple free graphs, J. Algorithms 32 (1999) 41–57.
[25] A.S. LaPaugh, Recontamination does not help to search a graph, J. ACM 40 (1993) 224–245.
[26] S.M. LaValle, B.H. Simov, G. Slutzki, An algorithm for searching a polygonal region with a flashlight, in: Proceedings of the 16th Annual Symposium on Computational Geometry, Hong Kong, 2000, New York, 2000, ACM, pp. 260–269 (electronic).
[27] J.-H. Lee, S.-M. Park, K.-Y. Chwa, Searching a polygonal room with one door by a 1-searcher, Internat. J. Comput. Geom. Appl. 10 (2000) 201–220.
[28] C. Lund, M. Yannakakis, On the hardness of approximation problems, J. ACM 41 (1994) 960–981.
[29] F.S. Makedon, C.H. Papadimitriou, I.H. Sudborough, Topological bandwidth, SIAM J. Algebraic Discrete Methods 6 (1985) 418–444.
[30] F.S. Makedon, I.H. Sudborough, On minimizing width in linear layouts, Discrete Appl. Math. 23 (1989) 243–265.
[31] B. Monien, I.H. Sudborough, Min cut is NP-complete for edge weighted trees, Theoret. Comput. Sci. 58 (1988) 209–229.
[32] A. Parra, P. Scheffler, Characterizations and algorithmic applications of chordal graph embeddings, Discrete Appl. Math. 79 (1997) 171–188.
[33] N.N. Petrov, S. Starostina, On some problem of guaranteed search, Vestn. St.Petersb. Univ. Math 27 (1994) 50–51.
[34] N.N. Petrov, I. Ture, A pursuit problem on a graph, Vestn. Leningr. Univ., Math. (translation from Vestn. Leningr. Univ., Ser. I 4, 1990, 12–18) 23 (1990) 12–19.
[35] R. Raz, S. Safra, A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP, in: Proceedings of the Annual ACM Symposium on Theory of Computing, Vol. 29, The Association for Computing Machinery, 1997, pp. 475–484.
[36] N. Robertson, P.D. Seymour, Graph minors—a survey, in: I. Anderson (Ed.), Surveys in Combinatorics, Cambridge University Press, Cambridge, 1985, pp. 153–171.
[37] K. Sugihara, I. Suzuki, Optimal algorithms for a pursuit-evasion problem in grids, SIAM J. Discrete Math. 2 (1989) 126–143.
[38] I. Suzuki, M. Yamashita, Searching for a mobile intruder in a polygonal region, SIAM J. Comput. 21 (1992) 863–888.