



Searching expenditure and interval graphs [☆]

Fedor V. Fomin*

*St. Petersburg State University, Mechanical and Mathematical Department, Bibliotechnaya square 2,
198904 St. Petersburg, Russia*

Abstract

The problem of searching a fugitive in a graph by a team of pursuers is considered. A new criterion of optimal search called the searching expenditure is defined. It is proved that for each graph G , the searching expenditure is equal to the number of edges in the least (with respect to the number of edges) interval graph containing G as a subgraph. © 2002 Elsevier B.V. All rights reserved.

Keywords: Graph; Searching

1. Introduction

The problems of searching a graph attract specialists from different areas of discrete mathematics due to several reasons.

The first reason is the connection between some searching problems and pebble games [8], which are related to the problems of rational usage of computer memory. Second, it turned out that some graph invariants first occurred in the theory of superlarge chips, such as width in layouts [10], topological bandwidth [11], and the size of a vertex cut in a graph [4], in many cases have a game theory interpretation. The third reason is the connection between searching problems and the path-width and tree-width of graphs, the important parameters in the theory of graph minors developed by Robertson and Seymour [1,3].

Search problems occur also in problems of the coordination of robots' movements [17] and those of providing information privacy in bugged channels [6]. Detailed

[☆] Translated from *Discrete Analysis and Operations Research*, Ser. 1, Novosibirsk 5(3) (1998) 70–79. This work was supported by the RFBR grant N01-01-00235.

* Corresponding author. Heinz Nixdorf Institute, University of Paderborn, Furstentallee 11, 33102 Paderborn, Germany.

E-mail address: fomin@gamma.math.spbu.ru, fomin@upb.de (F.V. Fomin).

information on searching problems and their relatives can be found in the surveys [1,5,13]. See also [12,14–16] for further references.

Sometimes it is convenient to interpret a searching problem as the pursuit-evasion problem for a “diffused” fugitive (e.g., dust or gas). At each step of the search, the vertices and edges where the fugitive can appear are declared to be polluted, and all the other vertices and edges are declared to be clean. The pursuers clean the graph as follows: at each step, a pursuer is put to a vertex of the graph, and after that, some pursuers can be removed. It is supposed that at the beginning all edges of the graph are polluted. A polluted edge becomes clean if both its ends are occupied by pursuers. A clean edge e becomes polluted if after removing a pursuer a path occurs connecting it to a polluted edge and having no pursuers on interior vertices.

An interesting interpretation of searching problems related to cleaning a graph is described by Bienstock [1]. Consider the behavior of a computer virus in a network. We are informed of its presence but do not know how many computers are infected. Assuming the worst, we must suspect that all the network is infected and thus all the nodes must be checked and cleaned. Suppose that it is impossible or inconvenient to check all the nodes simultaneously; so the problem arises of developing the optimal (with respect to some criterion) cleaning strategy.

In the “traditional” searching problem, we look for the program involving the least number of pursuers. In this paper, we are interested in programs involving the least total number of pursuers, obtained by summing over all the steps of the program. This number will be called the searching expenditure.

One of the most important questions in the searching problems is that of repeated cleaning (or monotonicity). It turns out [2,9] that for some searching problems with the “traditional” optimality criterion the following fact holds: if k pursuers are enough to clean the graph, then k pursuers can clean it so that the edges having been cleaned are not polluted again.

In this paper, we first prove the monotonicity for searching programs of minimal expenditure. The constructions used for this proof are close to those by Bienstock and Seymour [2].

Then we use the monotonicity theorem to find out how computing the search expenditure is related to the problem of extending to an interval graph with the least number of edges, which is equivalent to the problem of the graph profile (see [7]). The problem of computing the graph profile often arises in computational mathematics when working with matrices.

2. The problem

In what follows, we work with loopless finite non-directed graphs without multiple edges. The set of vertices of a graph G is denoted by $V(G)$, and the set of edges is denoted by $E(G)$.

A *searching program* Π on a graph G is a sequence of pairs

$$(A_0^1, Z_0^1), (A_0^2, Z_0^2), (A_1^1, Z_1^1), (A_1^2, Z_1^2), \dots, (A_n^1, Z_n^1), (A_n^2, Z_n^2)$$

such that

- (I) $A_i^j \subseteq E(G)$ and $Z_i^j \subseteq V(G)$ for each i and j such that $0 \leq i \leq n$ and $j = 1, 2$;
- (II) for each i and j such that $0 \leq i \leq n$ and $j = 1, 2$, every vertex incident with an edge of A_i^j and an edge of $E(G) - A_i^j$ belongs to Z_i^j ;
- (III) $A_0^j = \emptyset$, $A_n^j = E(G)$ for $j = 1, 2$;
- (IV) (*adding a pursuer*) for each i , $1 \leq i \leq n$, there exists a vertex v such that $Z_i^1 = Z_{i-1}^2 \cup \{v\}$ and $A_i^1 = A_{i-1}^2 \cup E_v$, where E_v is the set of all edges connecting v and Z_{i-1}^2 ;
- (V) (*removing a pursuer*) for each i , $1 \leq i \leq n$, we have $Z_i^2 \subseteq Z_i^1$, and A_i^2 is the set of all edges $e \in A_i^1$ such that each path containing e and an edge from $E(G) - A_i^1$ has an interior vertex from Z_i^2 .

It is convenient to interpret Z_i^1 as the set of vertices occupied by pursuers after adding a pursuer at the i th step, Z_i^2 as the set of vertices occupied by pursuers just before the $(i + 1)$ st step, and A_i^1 and A_i^2 as the sets of cleaned edges.

In the known problem of finding the vertex searching number [8], we must find the program with the least $\max_{0 \leq i \leq n} |Z_i^1|$. This number can be interpreted as the largest number of pursuers situated on the graph at the same time. We are interested in another searching measure. Let us define the *expenditure of a searching program* Π as $\sum_{0 \leq i \leq n} |Z_i^2|$. The expenditure of a searching program can be interpreted as the total number of “person-steps” used for the search. The *searching expenditure* in the graph G is the minimal expenditure of a searching program on it. We shall say that the searching program $(A_0^1, Z_0^1), (A_0^2, Z_0^2), \dots, (A_n^1, Z_n^1), (A_n^2, Z_n^2)$ is *monotone* if for each i , $0 \leq i \leq n$, we have $A_i^1 = A_i^2$ (after removing pursuers, the edges are not recontaminated). The *monotone searching expenditure* in the graph G is the minimal expenditure of a monotone searching program on it. The searching expenditure and the monotone searching expenditure in G will be denoted by $\gamma(G)$ and $\gamma_m(G)$, respectively.

Searching programs can also be defined for pseudographs. Additional multiple edges and loops do not affect the searching expenditure.

3. Monotonicity and tangles in pseudographs

For a subset $X \subseteq E(G)$ of edges of a graph G , we define $\delta(X)$ as the set of vertices incident simultaneously with edges of X and those of $E(G) - X$. The set of vertices in the subgraph of G induced by the edges of $X \subseteq E(G)$ will be denoted by $V(X)$.

The notion of a tangle is defined for pseudographs. Let the pseudograph G^0 be obtained from a graph G by adding a loop to each vertex. A *tangle* in G^0 is a sequence (X_0, X_1, \dots, X_n) of edge subsets in G^0 such that

- (1) $X_0 = \emptyset$ and $X_n = E(G^0)$;
- (2) $|V(X_i) - V(X_{i-1})| \leq 1$, $1 \leq i \leq n$;
- (3) if $v \in V(X_i)$, $1 \leq i \leq n$, then the loop at v also belongs to X_i .

We define the *measure* of a tangle (X_0, X_1, \dots, X_n) to be $\sum_{0 \leq i \leq n} |\delta(X_i)|$. The tangle (X_0, X_1, \dots, X_n) is called *augmenting* if $X_0 \subseteq X_1 \subseteq \dots \subseteq X_n$ and $|V(X_i) - V(X_{i-1})| = 1$ for each i , $1 \leq i \leq n$.

Theorem 1. For each graph G and integer $k \geq 0$ the following statements are equivalent:

- (i) $\gamma(G) \leq k$;
- (ii) if the pseudograph G^0 is obtained from G by adding a loop to each vertex, then G^0 contains a tangle of measure at most k ;
- (iii) if the pseudograph G^0 is obtained from G by adding a loop to each vertex, then G^0 contains an augmenting tangle of measure at most k ;
- (iv) $\gamma_m(G) \leq k$.

Proof. (i) \Rightarrow (ii). As it has been mentioned, $\gamma(G) = \gamma(G^0)$. Let

$$(A_0^1, Z_0^1), (A_0^2, Z_0^2), (A_1^1, Z_1^1), (A_1^2, Z_1^2), \dots, (A_n^1, Z_n^1), (A_n^2, Z_n^2)$$

be a searching program in G^0 of expenditure at most k . We shall prove that $A_0^2, A_1^2, \dots, A_n^2$ is a tangle of measure at most k in G^0 .

Due to Property III in the definition of a searching program, we have $A_0^2 = \emptyset$, $A_n^2 = E(G)$. Property II of that definition means that $\delta(A_i^2) \subseteq Z_i^2$ for each i , $0 \leq i \leq n$. Thus, $\sum_{1 \leq i \leq n} |\delta(A_i^2)| \leq k$. Let us check that for each i , $1 \leq i \leq n$, the inequality $|V(A_i^2) - V(A_{i-1}^2)| \leq 1$ holds. If there exist an index i , $1 \leq i \leq n$, and distinct vertices u, v such that $u, v \in V(A_i^2) - V(A_{i-1}^2)$, then the loops e_u, e_v added to these vertices belong to $A_i^2 - A_{i-1}^2$. Using the inclusion $A_i^1 \supseteq A_i^2$ (Property V in the definition of a searching program), we obtain $e_u, e_v \in A_i^1 - A_{i-1}^2$, which contradicts Property IV in the definition of a searching program.

(ii) \Rightarrow (iii). Let us choose a tangle (X_0, X_1, \dots, X_n) in G such that the sum

$$\sum_{0 \leq i \leq n} |\delta(X_i)| \tag{1}$$

is minimum and, under this condition, the sum

$$\sum_{0 \leq i \leq n} (|X_i| + 1) \tag{2}$$

is minimum.

First, we prove that for each j , $1 \leq j \leq n$, the inclusion $X_{j-1} \subseteq X_j$ holds.

Since $V(X_{j-1} \cup X_j) - V(X_{j-1}) = V(X_j) - V(X_{j-1})$ and $V(X_{j+1}) - V(X_{j-1} \cup X_j) \subseteq V(X_{j+1}) - V(X_j)$, it follows that $(X_0, X_1, \dots, X_{j-1}, X_{j-1} \cup X_j, X_{j+1}, \dots, X_n)$ is a tangle. Then (1) implies the inequality

$$|\delta(X_{j-1} \cup X_j)| \geq |\delta(X_j)|. \tag{3}$$

It can be easily checked that $|\delta|$ satisfies the inequality

$$|\delta(X_{j-1} \cup X_j)| + |\delta(X_{j-1} \cap X_j)| \leq |\delta(X_{j-1})| + |\delta(X_j)|. \tag{4}$$

By (3) and (4), we conclude that

$$|\delta(X_{j-1} \cap X_j)| \leq |\delta(X_{j-1})|. \tag{5}$$

If v belongs to the set $V(X_{j-1}) \cap V(X_j)$, then the incident loop belongs to $X_{j-1} \cap X_j$. Thus, $v \in V(X_{j-1} \cap X_j)$. Consequently, $V(X_j) - V(X_{j-1} \cap X_j) \subseteq V(X_j) - V(X_{j-1})$.

Then, $V(X_{j-1} \cap X_j) - V(X_{j-2}) \subseteq V(X_{j-1}) - V(X_{j-2})$, and thus $(X_0, X_1, \dots, X_{j-2}, X_{j-1} \cap X_j, X_j, X_{j+1}, \dots, X_n)$ is a tangle. It follows from (5), (1) and (2) that $|X_{j-1} \cap X_j| \geq |X_{j-1}|$. Thus, $X_{j-1} \subseteq X_j$ for each j , $1 \leq j \leq n$.

If $|V(X_j) - V(X_{j-1})| = 0$, then $(X_0, X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n)$ is a tangle, which contradicts (2). Thus, (X_0, X_1, \dots, X_n) is an augmenting tangle.

(iii) \Rightarrow (iv). Let (X_0, X_1, \dots, X_n) be an augmenting tangle of measure at most k in G^0 . Let us define a searching program in G^0 by putting $Z_0^1 = Z_0^2 = \emptyset$ and $Z_i^1 = \delta(X_i) \cup \{V(X_i) - V(X_{i-1})\}$, $Z_i^2 = \delta(X_i)$ for each i , $1 \leq i \leq n$. Suppose that at the j th step the pursuers occupy the vertices of Z_j^1 and that all edges of X_j are clean. Obviously, these edges will not be recontaminated after removing the pursuers from the vertices of $Z_j^1 - Z_j^2$.

Let $v = V(X_{j+1}) - V(X_j)$. Each edge from $X_{j+1} - X_j$ either is the loop incident with v , or it is incident with v and a vertex from $\delta(X_j) = Z_j^2$. Then at the $(j + 1)$ st step the pursuer occupying v cleans all the edges of $X_{j+1} - X_j$. Finally, $Z_0 = X_0$, $X_n = E(G)$, and thus $\gamma_m(G) = \gamma_m(G^0) \leq k$.

The implication (iv) \Rightarrow (i) is obvious. Theorem 1 is proved. \square

4. Interval graphs

An *interval graph* is a graph whose set of vertices coincides with some set of intervals on the real line, and two vertices are adjacent if and only if the corresponding intervals meet. A given set of such intervals is called an *interval realization* of the graph.

The following easy lemma is well known:

Lemma 1. *Each interval graph has an interval realization in which the ends of intervals are distinct integers $1, 2, \dots, |V(G)|$.*

Such a realization will be called a *canonical representation*.

Let $\mathcal{I} = \{I_v = (l_v, r_v)\}_{v \in V(G)}$, $l_v < r_v$, be a canonical representation of an interval graph G . The *length* of the representation \mathcal{I} is the value $\sum_{v \in V} [r_v - l_v]$. We define the *length* $l(G)$ of an interval graph G to be the minimum length of its canonical representation. For an arbitrary graph G , we denote by $il(G)$ its *interval length*, which is the minimum length of an interval graph containing G as a subgraph.

The following property of canonical representations of interval graphs having minimum length will be used in the proof of Theorem 2.

Lemma 2. *Let I be an interval graph on n vertices, and $\mathcal{I} = \{I_v = (l_v, r_v)\}_{v \in V(G)}$, $l_v < r_v$, be its canonical representation of the minimum length. Then*

$$\sum_{1 \leq i \leq n} |P(i)| = \sum_{v \in V} [r_v - l_v] = |E(I)|,$$

where $P(i)$, $1 \leq i \leq n$, denotes the set of intervals I_v , $v \in V(I)$, containing the number i .

Proof. Since \mathcal{I} is a canonical representation of minimum length, it follows that the numbers r_v (the right ends of intervals from \mathcal{I}) cannot be integers and that they must be at most $n + 1$. Each interval $I_v = (l_v, r_v)$, $v \in V(I)$, contains $\lfloor r_v - l_v \rfloor$ integers. Each number $i \in \{1, \dots, n\}$ is contained in exactly $|P(i)|$ intervals. Thus,

$$\sum_{1 \leq i \leq n} |P(i)| = \sum_{v \in V} \lfloor r_v - l_v \rfloor.$$

For each i , $1 \leq i \leq n$, the degree of the vertex $v(i = l_v)$ in the graph I is equal to $|P(i)| + \lfloor r_v - l_v \rfloor$ (the number of intervals I_u , $l_u < i < r_u$, plus the number of intervals I_w , $i < l_w < r_w$). As a result, we have

$$2|E(I)| = \sum_{v \in V(I)} \deg(v) = \sum_{1 \leq i \leq n} |P(i)| + \sum_{v \in V} \lfloor r_v - l_v \rfloor,$$

where $\deg(v)$ is the degree of v . Lemma 2 is proved. \square

Theorem 2. For each graph G and integer $k > 0$, the following statements are equivalent:

- (i) $\gamma(G) \leq k$;
- (ii) $il(G) \leq k$;
- (iii) there exists an interval graph with at most k edges that contains G as a subgraph.

Proof. (i) \Rightarrow (ii). Let

$$(A_0^1, Z_0^1), (A_0^2, Z_0^2), (A_1^1, Z_1^1), (A_1^2, Z_1^2), \dots, (A_n^1, Z_n^1), (A_n^2, Z_n^2)$$

be a searching program in G with the expenditure at most k . According to Theorem 1, we suppose that this program is monotone. Then without loss of generality we may also assume that $n = |V(G)|$. Let us choose $\varepsilon < 1$, and to each vertex v of G assign the interval $(l_v, r_v + \varepsilon)$, where l_v is the number of step when a pursuer occurs at the vertex for the first time, and r_v is the number of step when the vertex is occupied by a pursuer for the last time (i.e., the largest i such that $v \in Z_i^1$).

After the searching program is terminated, all the edges of G become clean. Hence for each edge e in G there is a step when both ends of e are occupied by pursuers, i.e., the moment of cleaning e . So, the interval graph I whose canonical representation is $\mathcal{I} = \{I_v = (l_v, r_v + \varepsilon)\}_{v \in V(G)}$ contains G as a subgraph. Since

$$\sum_{v \in V} \lfloor r_v + \varepsilon - l_v \rfloor = \sum_{v \in V} \lfloor r_v - l_v \rfloor = \sum_{0 \leq i \leq n} |Z_i^2|$$

(each vertex v is counted $\lfloor r_v - l_v \rfloor$ times in the right-hand side), we have $il(G) \leq l(I) \leq k$.

The implication (ii) \Rightarrow (iii) is a corollary of Lemma 2.

(iii) \Rightarrow (i). Let $\mathcal{I} = \{I_v = (l_v, r_v)\}_{v \in V(G)}$, $l_v < r_v$, be a canonical representation of minimum length of an interval graph I containing G as a subgraph. We assume that $r_v < n + 1$ for each v , where $n = |V(I)|$.

Let us describe the following searching program on G :

- $Z_0^1 = \emptyset$;
- for each i , $1 \leq i \leq n$, define $Z_i^1 = Z_{i-1}^2 \cup \{v\}$, where v is the vertex corresponding to the interval with the left end i ;
- $Z_i^2 = P(i+1)$ for each i , $0 \leq i \leq n-1$;
- $Z_n^2 = \emptyset$.

Since each path in the interval graph I (and thus in G) connecting a vertex w , $l_w > i+1$, with a vertex u , $l_u < i+1$, contains a vertex from $P(i+1)$, it follows that no recontamination can happen. For each edge of I (and thus for each edge of G), there exists a set of vertices Z_i^1 such that both ends of this edge belong to it. So, after the program is terminated, all the edges of the graph will be cleaned.

The expenditure of the searching program built is equal to

$$\sum_{0 \leq i \leq n} |Z_i^2| = \sum_{1 \leq i \leq n} |P(i)|.$$

Due to Lemma 2, we have $\gamma(G) \leq |E(I)|$. Theorem 2 is proved. \square

References

- [1] D. Bienstock, Reliability of Computer and Communication Networks, Providence, RI; D. Bienstock, Graph searching, path-width, tree-width and related problems (a survey), Amer. Math. Soc., DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 5 (1991) 33–49.
- [2] D. Bienstock, P. Seymour, Monotonicity in graph searching, J. Algorithms 12 (2) (1991) 239–245.
- [3] N.D. Dendris, L.M. Kirousis, D.M. Thilikos, Fugitive-search games on graphs and related parameters, Theoret. Comput. Sci. 172 (1–2) (1997) 233–254.
- [4] J.A. Ellis, I.H. Sudborough, J.S. Turner, The vertex separation and search number of a graph, Inform. Comput. 113 (1) (1994) 50–79.
- [5] F.V. Fomin, N.N. Petrov, Pursuit-evasion and search problems on graphs, Congr. Numer. 122 (1996) 47–58.
- [6] M. Franklin, Z. Galil, M. Yung, Eavesdropping games: A graph-theoretic approach to privacy in distributed systems, J. ACM, 47 (2000) 225–243.
- [7] A.P. Golovach, F.V. Fomin, The total vertex separation number and profile of a graph, Discrete Math. Appl. 85 (1) (1998) 73–80.
- [8] L.M. Kirousis, C.H. Papadimitriou, Searching and pebbling, Theoret. Comput. Sci. 47 (2) (1986) 205–218.
- [9] A.S. LaPaugh, Recontamination does not help to search a graph, J. Assoc. Comput. Mach. 40 (2) (1993) 224–245.
- [10] F.S. Makedon, I.H. Sudborough, On minimizing width in linear layouts, Discrete Appl. Math. 23 (3) (1989) 243–265.
- [11] F.S. Makedon, C.H. Papadimitriou, I.H. Sudborough, Topological bandwidth, SIAM J. Algebraic Discrete Math. 6 (3) (1985) 418–444.
- [12] N. Megiddo, S.L. Hakimi, M.R. Garey, D.S. Johnson, C.H. Papadimitriou, The complexity of searching a graph, J. Assoc. Comput. Mach. 35 (1) (1988) 18–44.
- [13] R.H. Möhring, Graph problems related to gate matrix layout and PLA folding, Computational Graph Theory, Vol. 7, Computational Supplement, Springer, Vienna, 1990, pp. 17–51.
- [14] T.D. Parsons, Pursuit-Evasion in a Graph, Theory and Application of Graphs, Springer, Berlin, 1976, pp. 426–441.

- [15] N.N. Petrov, Some extremal searching problems on graphs, *Differentsial'nye Uravneniya* 18 (5) (1982) 821–827 (in Russian).
- [16] N.N. Petrov, Pursuit-evasion problems with no information about the fugitive, *Differentsial'nye Uravneniya* 18 (5) (1982) 1345–1352 (in Russian).
- [17] K. Sugihara, I. Suzuki, Optimal algorithms for a pursuit-evasion problem in grids, *SIAM J. Discrete Math.* 2 (1) (1989) 126–143.