SIAM J. DISCRETE MATH.
Vol. 13, No. 4, pp. 454–464


# GRAPH SEARCHING AND INTERVAL COMPLETION*

FEDOR V. FOMIN† AND PETR A. GOLOVACH‡

**Abstract.** In the early studies on graph searching a graph was considered as a system of tunnels in which a fast and clever fugitive is hidden. The "classical" search problem is to find a search plan using the minimal number of searchers. In this paper, we consider a new criterion of optimization, namely, the search cost. First, we prove monotone properties of searching with the smallest cost. Then, making use of monotone properties, we prove that for any graph $G$ the search cost of $G$ is equal to the smallest number of edges of all interval supergraphs of $G$. Finally, we show how to compute the search cost of a cograph and the corresponding search strategy in linear time.

**Key words.** graph searching, search cost, vertex separation, interval graph completion, linear layout, profile, cograph

**AMS subject classifications.** 05C78, 05C50, 68R10

**PII.** S0895480199350477

**1. Introduction.** Search problems on graphs were introduced by Parsons [29] and Petrov [30, 31] (see also [27]). These problems attract the attention of researchers from different fields of mathematics and computer science for a variety of reasons. In the first place, this is the resemblance of graph searching to certain pebble games [21] which model sequential computation. The second motivation of the interest to graph searching arises from VLSI theory. The use of game-theoretic approaches to some important parameters of graph layouts such as cutwidth [26], topological bandwidth [25], and vertex separation number [13] is very useful for constructions of efficient algorithms. Yet another reason is connections between graph searching, pathwidth, and treewidth. These parameters play a very important role in the theory of graph minors developed by Robertson and Seymour (see [1, 12, 33]). Also, graph searching has applications in motion coordinations of multiple robots [34] and in problems of privacy in distributed environments with mobile eavesdroppers ("bugs") [15]. One can find more information on graph searching and related problems in the surveys [1, 4, 14, 28, 32].

In the "classical" node-search version of searching (see, e.g., [21]), at every move of searching a searcher is placed at a vertex or is removed from a vertex. Initially all edges are contaminated (uncleared). A contaminated edge is cleared once both its endpoints are occupied by searchers. A clear edge $e$ is recontaminated if there is a path without searchers leading from $e$ to a contaminated edge. The "classical" search problem is to find the search program such that the maximal number of searchers used at any move is minimized. In this paper we introduce an alternative criterion of optimization. We are looking for node-search programs with the minimal sum of numbers of searchers (the sum is taken over all moves of the search program). We call this criterion the search cost of a graph. Loosely speaking, the cost of a search

*Received by the editors December 30, 1998; accepted for publication (in revised form) July 5, 2000; published electronically October 6, 2000.

http://www.siam.org/journals/sidma/13-4/35047.html

†Faculty of Mathematics and Mechanics, St. Petersburg State University, Bibliotechnaya sq.2, St. Petersburg, 198904, Russia (fomin@gamma.math.spbu.ru). The research of this author was partially supported by the RFBR grant N98-01-00934.

‡Department of Applied Mathematics, Faculty of Mathematics, Syktyvkar State University, Oktyabrsky pr., 55, Syktyvkar, 167001, Russia (golovach@ssu.komi.com). The research of this author was partially supported by the RFBR grant N96-00-00285.

program is the total number of "man-steps" used in this program, and the search cost is the cost of an optimal program. The reader is referred to section 2 for formal definitions of searching and its cost.

One of the most important issues concerning searching is that of recontamination. In some search problems (see [2, 23]) recontamination does not help to search a graph, i.e., if searchers can clear the graph, then they can do it without recontamination of previously cleared edges. We establish the monotonicity of search programs of the smallest cost. To prove the monotonicity we use special constructions called clews. Clews are closely related to crusades used by Bienstock and Seymour in [2] and the notion of measure of clew is related to the notion of linear width introduced by Thomas in [35].

This paper is organized as follows. In section 2 we give necessary definitions. In section 3 we introduce clews and prove the monotonicity of graph searching. In section 4 it is proved that for any graph $G$ the search cost of $G$ is equal to the smallest number of edges of an interval supergraph of $G$. In section 5 it is shown that for any graph $G$ the search cost of $G$ is equal to the vertex separation sum and profile of $G$. In section 6 we show how to compute the search cost of the product of graphs. In section 7 we prove that the search cost of a cograph and the corresponding search program can be obtained in linear time.

**2. Statement of the problem.** We use the standard graph-theoretic terminology compatible with [7], to which we refer the reader for basic definitions. Unless otherwise specified, $G$ is an undirected, simple (without loops and multiple edges), and finite graph with the vertex set $V(G)$ and the edge set $E(G)$; $n$ denotes the order of $G$, i.e., $|V(G)| = n$.

A *search program* $\Pi$ on a graph $G$ is the sequence of pairs

$$(A_0^1, Z_0^1), (A_0^2, Z_0^2), (A_1^1, Z_1^1), (A_1^2, Z_1^2), \ldots, (A_m^1, Z_m^1), (A_m^2, Z_m^2)$$

such that the following hold.

I. For $i \in \{0, \ldots, m\}$, $j \in \{1, 2\}$, $A_i^j \subseteq E(G)$, and $Z_i^j \subseteq V(G)$.

II. For $i \in \{0, \ldots, m\}$, $j \in \{1, 2\}$, any vertex incident to an edge in $A_i^j$ and to an edge in $E(G) - A_i^j$ is in $Z_i^j$.

III. For $j \in \{1, 2\}$, $A_0^j = \emptyset$, and $A_m^j = E(G)$.

IV (*placing new searchers and clearing edges*). For $i \in \{1, \ldots, m\}$, there is $v \in V(G) \setminus Z_{i-1}^2$ 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 incident to $v$ edges having one end in $Z_{i-1}^2$.

V (*removing searchers and possible recontamination*). For $i \in \{1, \ldots, m\}$, $Z_i^2 \subseteq Z_i^1$ and $A_i^2$ is the set of all edges $e \in A_i^1$ such that every path containing $e$ and an edge of $E(G) - A_i^1$ has an internal vertex in $Z_i^2$.

We call these the *search axioms*. It is useful to treat $Z_i^1$ as the set of vertices occupied by searchers immediately after placing a new searcher at the $i$th step, $Z_i^2$ as the set of vertices occupied by searchers immediately before making the $(i+1)$th step, and $A_i^1$, $A_i^2$ as the sets of cleared edges.

The well-known node search problem [21] is to find $\Pi$ with the smallest $\max_{i \in \{0,\ldots,m\}} |Z_i^1|$. (This maximum can be treated as the maximum number of searchers used in one step.) Let us suggest an alternative measure of search. We define the *cost* of $\Pi$ to be $\sum_{i=0}^{m} |Z_i^2|$. One can interpret the cost of a search program as the total number of "man-steps" used for the search or as the total sum that searchers earn for doing their job. The *search cost* of a graph $G$, denoted by $\gamma(G)$, is the minimum cost of a search program where the minimum is taken over all search programs on $G$.

A search program $(A_0^1, Z_0^1), (A_0^2, Z_0^2), \ldots, (A_m^1, Z_m^1), (A_m^2, Z_m^2)$ is *monotone* if, for each $i \in \{0, \ldots, m\}$, $A_i^1 = A_i^2$. (Recontamination does not occur when searchers are removed at the $i$th step.) The *monotone search cost* of $G$, $\gamma_m(G)$, is the cost of the minimal (over all monotone search programs) search program on $G$.

Notice that search programs can be defined not only for simple graphs but for graphs with loops and multiple edges as well. Adding loops and multiple edges does not change the search cost. (We suppose that a loop edge is cleared once a searcher is placed on its endpoint.)

**3. Monotone programs and clews.** Let $G$ be a graph. For $X \subseteq E(G)$ we define $V(X)$ to be the set of vertices which are endpoints of $X$, and we let $\delta(X) = V(X) \cap V(E(G) - X)$.

We consider only clews in graphs of a special structure. Let $G^0$ be obtained by adding a loop at each vertex of a graph $G$. A *clew* in $G^0$ is the sequence $(X_0, X_1, \ldots, X_m)$ of subsets of $E(G^0)$ such that the following hold.

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

The *measure of the clew* is $\sum_{i=0}^{m} |\delta(X_i)|$. A clew $(X_0, X_1, \ldots, X_m)$ is *progressive* if $X_0 \subseteq X_1 \subseteq \cdots \subseteq X_m$ and for any $i \in \{1, \ldots, m\}$, $|V(X_i) - V(X_{i-1})| = 1$. Notice that if a clew $(X_0, X_1, \ldots, X_m)$ is progressive, then $m = n$.

THEOREM 3.1. *For any graph $G$ and $k \geq 0$ the following assertions are equivalent.*

(i) $\gamma(G) \leq k$.

(ii) *Let $G^0$ be obtained by adding a loop at each vertex of $G$. There is a clew in $G^0$ of measure $\leq k$.*

(iii) *Let $G^0$ be obtained by adding a loop at each vertex of $G$. There is a progressive clew in $G^0$ of measure $\leq k$.*

(iv) $\gamma_m(G) \leq k$.

*Proof.* (i) $\Rightarrow$ (ii). As mentioned above, $\gamma(G) = \gamma(G^0)$. Let

$$(A_0^1, Z_0^1), (A_0^2, Z_0^2), (A_1^1, Z_1^1), (A_1^2, Z_1^2), \ldots, (A_m^1, Z_m^1), (A_m^2, Z_m^2)$$

be a search program on $G^0$ with the cost $\leq k$. We prove that $A_0^2, A_1^2, \ldots, A_m^2$ is the clew of measure $\leq k$ in $G^0$. The third search axiom implies $A_0^2 = \emptyset$, $A_m^2 = E(G)$. The second search axiom says that for every $i \in \{0, \ldots, m\}$, $\delta(A_i^2) \subseteq Z_i^2$, and hence $\sum_{i=1}^{m} |\delta(A_i^2)| \leq k$. Thus $A_0^2, A_1^2, \ldots, A_m^2$ is the clew if $|V(A_i^2) - V(A_{i-1}^2)| \leq 1$ for every $i \in \{1, \ldots, m\}$. Suppose that for some $i \in \{1, \ldots, m\}$ this inequality does not hold. Then there are vertices $u \neq v$ of $V(A_i^2) - V(A_{i-1}^2)$. Notice that the loops $e_u, e_v$ at $u$ and $v$ belong to $A_i^2 - A_{i-1}^2$. From the fifth search axiom $A_i^1 \supseteq A_i^2$ it follows that $e_u, e_v \in A_i^1 - A_{i-1}^2$. The latter is in contradiction with the fourth axiom.

(ii)$\Rightarrow$ (iii). The proof of this implication is closely related to the crusades monotonicity proof by Bienstock and Seymour [2]. Let us choose a clew $(X_0, X_1, \ldots, X_m)$ in $G^0$ such that

$$\sum_{i=0}^{m} |\delta(X_i)| \text{ is minimum} \tag{3.1}$$

and, subject to (3.1),

$$\sum_{i=0}^{m} (|X_i| + 1) \text{ is minimum.} \tag{3.2}$$

First we prove that $X_{i-1} \subseteq X_i$ for $i \in \{1, \ldots, m\}$.

Because $V(X_{i-1} \cup X_i) - V(X_{i-1}) = V(X_i) - V(X_{i-1})$ and $V(X_{i+1}) - V(X_{i-1} \cup X_i) \subseteq V(X_{i+1}) - V(X_i)$, it follows that $(X_0, X_1, \ldots, X_{i-1}, X_{i-1} \cup X_i, X_{i+1}, \ldots, X_m)$ is the clew. Using (3.1), we obtain

$$|\delta(X_{i-1} \cup X_i)| \geq |\delta(X_i)|. \tag{3.3}$$

It is easy to check that $|\delta|$ satisfies the submodular inequality

$$|\delta(X_{i-1} \cup X_i)| + |\delta(X_{i-1} \cap X_i)| \leq |\delta(X_{i-1})| + |\delta(X_i)|. \tag{3.4}$$

Combining (3.3) and (3.4), we obtain

$$|\delta(X_{i-1} \cap X_i)| \leq |\delta(X_{i-1})|. \tag{3.5}$$

If $v \in V(X_{i-1}) \cap V(X_i)$, then the loop at $v$ belongs to $X_{i-1} \cap X_i$ by the third search axiom; therefore, $v \in V(X_{i-1} \cap X_i)$ and, consequently, $V(X_i) - V(X_{i-1} \cap X_i) \subseteq V(X_i) - V(X_{i-1})$. Thus, $V(X_{i-1} \cap X_i) - V(X_{i-2}) \subseteq V(X_{i-1}) - V(X_{i-2})$, and $(X_0, X_1, \ldots, X_{i-2}, X_{i-1} \cap X_i, X_i, X_{i+1}, \ldots, X_m)$ is a clew. Taking into account (3.5), (3.1), and (3.2), we get $|X_{i-1} \cap X_i| \geq |X_{i-1}|$. Thus we have $X_{i-1} \subseteq X_i$.

If $|V(X_i) - V(X_{i-1})| = 0$, then $(X_0, X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_m)$ is the clew contradicting (3.2). Hence, $(X_0, X_1, \ldots, X_m)$ is progressive.

(iii)⇒ (iv). Let $(X_0, X_1, \ldots, X_n)$ be a progressive clew of measure $\leq k$ in $G^0$. We define the search program on $G^0$ setting $Z_0^1 = Z_0^2 = \emptyset$ and $Z_i^1 = \delta(X_i) \cup \{V(X_i) - V(X_{i-1})\}$, $Z_i^2 = \delta(X_i)$ for $i \in \{1, \ldots, n\}$. Suppose that at the $i$th step searchers are placed at vertices of $Z_i^1$ and that all edges of $X_i$ are cleaned. Obviously, no recontamination occurs by removing all searchers from vertices of $Z_i^1 - Z_i^2$.

Define $v = V(X_{i+1}) - V(X_i)$. Every edge of $X_{i+1} - X_i$ either is the loop at $v$, or is incident to $v$ and to a vertex of $\delta(X_i) = Z_i^2$. Then at the $(i+1)$th step the searcher placed at $v$ cleans all edges of $X_{i+1} - X_i$. Finally, $X_0 = \emptyset$, $X_n = E(G)$ with the result that $\gamma_m(G) = \gamma_m(G^0) \leq k$.

(iv)⇒ (i). This part of the proof is obvious. □

**4. Interval graphs.** A graph $G$ is an *interval graph* if and only if one can associate with each vertex $v \in V(G)$ an open interval $I_v = (l_v, r_v)$ on the real line, such that for all $v, w \in V(G)$, $v \neq w$: $(v, w) \in E(G)$ if and only if $I_v \cap I_w \neq \emptyset$. The set of intervals $\mathcal{I} = \{I_v\}_{v \in V}$ is called an (interval) *representation* of $G$.

It is easy to check that every interval graph has an interval representation in which the left endpoints are distinct integers $1, 2, \ldots, n$. Such a representation is said to be *canonical*.

A graph $G$ is a *supergraph* of the graph $G'$ if $V(G') = V(G)$ and $E(G') \subseteq E(G)$. Let $G$ be an interval graph, and let $\mathcal{I} = \{I_v\}_{v \in V(G)}$ be a canonical representation of $G$. The length of $G$ with respect to $\mathcal{I}$, denoted by $l(G, \mathcal{I})$, is

$$\sum_{v \in V} \lfloor r_v - l_v \rfloor.$$

We define the *length* $l(G)$ of an interval graph $G$ as the minimum length over all canonical representations of $G$. For any graph $G$ we define the *interval length* of $G$, denoted by $il(G)$, as the smallest length over all interval supergraphs of $G$.

In the proof of Theorem 4.2, we shall use the following property of canonical representation.

LEMMA 4.1. *Let $I$ be an interval graph of $n$ vertices and $\mathcal{I} = \{I_v = (l_v, r_v)\}_{v \in V(G)}$, $l_v < r_v$, be its canonical representation such that*

$$\sum_{v \in V} \lfloor r_v - l_v \rfloor \text{ is minimum.} \tag{4.1}$$

*For $i \in \{1, \ldots, n\}$ let $P(i)$ be the set of intervals $I_v$, $v \in V(I)$, containing $i$. Then*

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

*Proof.* (4.1) implies that every interval $I_v = (l_v, r_v)$, $v \in V(I)$, contains $\lfloor r_v - l_v \rfloor$ integers. Every $i \in \{1, \ldots, n\}$ belongs to $|P(i)|$ intervals of $\mathcal{I}$; therefore,

$$\sum_{i=1}^{n} |P(i)| = \sum_{v \in V} \lfloor r_v - l_v \rfloor. \tag{4.2}$$

Let $deg(v)$ be the degree of a vertex $v$ in $I$. Then for every $v \in V(I)$

$$deg(v) = |P(l_v)| + \lfloor r_v - l_v \rfloor \tag{4.3}$$

(the number of intervals $I_u = (l_u, r_u)$ such that $l_u < l_v < r_u$ plus the number of intervals $I_w = (l_w, r_w)$ such that $l_v < l_w < r_v$). By (4.3),

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

which, combined with (4.2), proves Lemma 4.1. □

THEOREM 4.2. *For any graph $G$ and $k > 0$ the following assertions are equivalent.*

(i) $\gamma(G) \leq k$.

(ii) $il(G) \leq k$.

(iii) *There is an interval supergraph $I$ of $G$ such that $|E(I)| \leq k$.*

*Proof.* (i)⇒ (ii). If $\gamma(G) \leq k$, then by Theorem 3.1 there is a monotone search program

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

on $G$ with cost $\leq k$. We choose $\varepsilon < 1$ and assign to each vertex $v$ of $G$ the interval $(l_v, r_v + \varepsilon)$, where a searcher is placed on $v$ at the $l_v$th step and this searcher is removed from $v$ at the $r_v$th step, i.e., $l_v = \min\{i \in \{1, \ldots, n\} | v \in Z_i^1\}$ and $r_v = \max\{i \in \{1, \ldots, n\} | v \in Z_i^1\}$. After the $n$th step of the search program all edges of $G$ are cleared, and hence for every edge $e$ of $G$ there is a step such that both ends of $e$ are occupied by searchers. Therefore, the interval graph $I$ with the canonical representation $\mathcal{I} = \{I_v = (l_v, r_v + \varepsilon)\}_{v \in V(G)}$ is the supergraph of $G$. Because for sufficiently small $\varepsilon$

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

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

(ii)⇒ (iii). This part of the proof follows immediately from Lemma 4.1.

(iii)⇒ (i). Let $\mathcal{I} = \{I_v = (l_v, r_v)\}_{v \in V(G)}$, $l_v < r_v$, be a canonical representation of the minimal length of an interval supergraph $I$ of $G$. It is clear that $r_v < n + 1$, $v \in V(G)$.

Let us describe the following search program on $G$.

- $Z_0^1 = \emptyset$.
- For $i \in \{1, \dots, n\}$, we put $Z_i^1 = Z_{i-1}^2 \cup \{v\}$, where $v$ is the vertex assigned to the interval with the left endpoint $i$.
- For $i \in \{0, \dots, n-1\}$, $Z_i^2 = P(i+1)$.
- $Z_n^2 = \emptyset$.

Such actions of searchers do not imply recontamination because each path in $I$ (and hence in $G$) from a vertex $w$, $l_w > i + 1$ to a vertex $u$, $l_u < i + 1$ contains a vertex of $P(i + 1)$. For each edge $e$ of $I$ (and hence for each edge of $G$) there is $i \in \{1, \dots, n\}$ such that both ends of $e$ belong to $Z_i^1$. Then at the $n$th step all edges are cleared.

The cost of the program is

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

By Lemma 4.1 $\gamma(G) \le |E(I)|$. □

Notice that by Theorem 4.2 for any graph $G$ on $n$ vertices and $e$ edges

$$e \le \gamma(G) \le \frac{1}{2} n(n-1).$$

Also, $\gamma(G) = e$ if and only if $G$ is an interval graph, and $\gamma(G) = \frac{1}{2}n(n-1)$ if and only if $G$ is a complete graph.

**5. Linear layouts.** A *linear layout* of a graph $G$ is a one-to-one mapping $f: V(G) \to \{1, \dots, n\}$. There are various interesting parameters associated with linear layouts like Bandwidth, Sum Bandwidth, Cutwidth, etc. (see [10] and [26] for further references).

Define the *profile* [10] of a symmetric $n \times n$ matrix $A = (a_{ij})$ as the minimum value of the sum

$$\sum_{i=1}^{n} (i - \min\{j: a_{ij} \neq 0\})$$

taken over all symmetric permutations of $A$, it being assumed that $a_{ii} = 1$, $i \in \{1, \dots, n\}$. Profile reduction is relevant to the speedup of matrix computations; see [10] for further references on profile. The profile may be redefined as a graph invariant $p(G)$ by finding a linear layout $f$ of $G$ which minimizes the sum

$$\sum_{u \in V(G)} (f(u) - \min\{f(v): v \in V(G), v \cong u\}),$$

where $\cong$ stands for "is adjacent or equal to."

For $U \subseteq V(G)$ we define

$$\partial U = \{u: u \in U \text{ and there exists } v \in V(G) \setminus U \text{ such that } (u, v) \in E(G)\}.$$

Ellis, Sudborough, and Turner [13] (see also [24]) introduced the following graph parameter. Let $f$ be a linear layout of a graph $G$. Denote by $S_i(G, f)$, $i \in \{1, \dots, n\}$,

the set of vertices $\{v: v \in V(G), f(v) \leq i\}$. The *vertex separation with respect to layout* $f$ is

$$vs(G,f) = \max_{i \in \{1,\ldots,n-1\}} |\partial S_i(G,f)|,$$

and the *vertex separation* $vs(G)$ of a graph $G$ is the minimum vertex separation over all linear layouts of $G$. Let us define an alternative "average norm" ("cost version") of vertex separation. The *vertex separation sum with respect to layout* $f$ is

$$vs_{sum}(G,f) = \sum_{i=1}^{n-1} |\partial S_i(G,f)|,$$

and the *vertex separation sum* $vs_{sum}(G)$ of a graph $G$ [18] is the minimum vertex separation sum over all linear layouts of $G$.

The following result is due to Billionnet [3].

THEOREM 5.1 (Billionnet). *For any graph $G$ the profile of $G$ is equal to the smallest number of edges, where minimum is taken over all interval supergraphs of $G$.*

The next theorem is related to the theorem of Kirousis and Papadimitriou [20] about node search number and interval width.

THEOREM 5.2. *For any graph $G$ and $k > 0$, the following assertions are equivalent.*

(i) $\gamma(G) \leq k$.
(ii) $vs_{sum}(G) \leq k$.
(iii) $p(G) \leq k$.

*Proof.* Because the application of Theorems 4.2 and 5.1 yields (i)$\Leftrightarrow$ (iii), it remains to prove (i)$\Leftrightarrow$ (ii).

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

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

be a monotone search program on $G$ with cost $\leq k$. Define a layout $f: V(G) \to \{1,\ldots,n\}$ so that $f(u) < f(v)$ if and only if $u$ accepts a searcher before $v$ does. By the second search axiom $|Z_i^2| \leq |\partial S_i(G,f)|$ for each $i \in \{1,\ldots,n\}$, and we conclude that $vs_{sum}(G,f) \leq k$.

(ii)$\Rightarrow$ (i). Let $f: V(G) \to \{1,\ldots,n\}$ be a linear layout such that $vs_{sum}(G,f) \leq k$. We define the following subsets of $V(G)$.

- $Z_0^1 = Z_0^2 = \emptyset$.
- For $i \in \{1,\ldots,n\}$, we put $Z_i^1 = Z_{i-1}^2 \cup \{v\}$, where $v = f^{-1}(i)$.
- For $i \in \{1,\ldots,n\}$, $Z_i^2 = \partial S_i(G,f)$.

For $i \in \{0,\ldots,n\}$ and $j = 1,2$, let $A_i^j$ be the set of edges induced by $\cup_{k=0}^{i} Z_k^1$. We observe that for each edge $(u,v) \in E(G)$ there is $i \in \{1,\ldots,n-1\}$ such that $u, v \in Z_i^1$. (If $f(u) < f(v)$, then $i = f^{-1}(v)$.) On this basis it is straightforward to prove (as in Theorem 4.2) that the sequence

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

is the (monotone) search program of cost $\leq k$ on $G$. ∎

**5.1. Complexity remark.** The problem of INTERVAL GRAPH COMPLETION is as follows.

Instance: A graph $G$ and an integer $k$.

Question: Is there an interval supergraph $I$ of $G$ such that $|E(I) - E(G)| \leq k$?

This problem is NP-complete even when $G$ is stipulated to be an edge graph (see [16, Problem GT35]). INTERVAL GRAPH COMPLETION arises in computational biology (see, e.g., [5]) and is known to be *fixed parameter tractable (FPT)* [9, 19].

From Theorem 4.2 it follows immediately that the problem of SEARCH COST, deciding given a graph $G$ and an integer $k$ whether $\gamma(G) \leq k$ or not, is NP-complete even for edge graphs and that finding the search cost is $FPT$ for a fixed $k$.

An $O(n^{1.722})$ time algorithm was given in [22] for the profile problem for the case that $G$ is a tree with $n$ vertices.

**6. Product of graphs.** Let $G$ and $H$ be disjoint graphs, i.e., $V(G) \cap V(H) = \emptyset$. The disjoint union of disjoint graphs $G$ and $H$ is the graph $G \dot{\cup} H$ with the vertex set $V(G) \cup V(H)$ and the edge set $E(G) \cup E(H)$.

We use $G \times H$ to denote the following type of "product" of disjoint graphs $G$ and $H$: $G \times H$ is the graph with the vertex set $V(G) \cup V(G)$ and the edge set $E(G) \cup E(H) \cup \{(u, w) : v \in V(G), w \in V(H)\})$.

The following theorem is similar to results on edge-search and node-search numbers [17, 28] and on the pathwidth and the treewidth of a graph [6].

THEOREM 6.1. *Let $G_1, G_2$ be disjoint graphs, $|V(G_1)| = n_1$, $|V(G_2)| = n_2$. Then*

(i) $\gamma(G_1 \dot{\cup} G_2) = \gamma(G_1) + \gamma(G_2)$.

(ii) $\gamma(G_1 \times G_2) = \min\{\frac{1}{2}n_1(n_1 - 1) + \gamma(G_2), \frac{1}{2}n_2(n_2 - 1) + \gamma(G_1)\} + n_1 n_2$.

*Proof.*

(i) This part of the proof is trivial.

(ii) Let $f$ be an optimal layout of $G_1 \times G_2$, i.e.,

$$vs_{sum}(G_1 \times G_2) = \sum_{i=1}^{n_1+n_2-1} |\partial S_i(G_1 \times G_2, f)|.$$

Let $k$ be the smallest number ensuring that $V(G_1) \subseteq S_k(G_1 \times G_2, f)$ or $V(G_2) \subseteq S_k(G_1 \times G_2, f)$. For clarity's sake, without loss of generality we suppose that $V(G_1) \subseteq S_k(G_1 \times G_2, f)$. Obviously, $k \geq n_1$. Let $g : V(G_2) \to n_2$ be the "restriction" of $f$ to $V(G_2)$, i.e., for any $u, v \in V(G_2)$, $g(u) < g(v)$ if and only if $f(u) < f(v)$.

Putting $S_0(G_2, g) = \emptyset$, we see that $|\partial S_i(G_1 \times G_2, f)| = i$ for $i \in \{1, \ldots, k-1\}$ and $|\partial S_i(G_1 \times G_2, f)| = n_1 + |\partial S_{i-n_1}(G_2, g)|$ for $i \in \{k, \ldots, n_1 + n_2 - 1\}$. In addition, for any $i \in \{1, \ldots, n_2\}$, $|\partial S_i(G_2, g)| \leq i$. Consequently,

$$\begin{aligned}
vs_{sum}(G_1 \times G_2) &= \sum_{i=1}^{n_1+n_2-1} |\partial S_i(G_1 \times G_2, f)| \\
&= \sum_{i=1}^{k-1} i + \sum_{i=k}^{n_1+n_2-1} (n_1 + |\partial S_{i-n_1}(G_2, g)|) \\
&= \sum_{i=1}^{n_1} i + \sum_{i=1}^{k-n_1-1} i + \sum_{i=k}^{n_1+n_2-1} |\partial S_{i-n_1}(G_2, g)| + n_1(n_2 - 1) \\
&= \frac{1}{2}n_1(n_1 - 1) + \sum_{i=1}^{k-n_1-1} i + \sum_{i=k-n_1}^{n_2-1} |\partial S_i(G_2, g)| + n_1 n_2
\end{aligned}$$

$$\begin{aligned}&\geq \frac{1}{2}n_1(n_1-1)+\sum_{i=1}^{n_2-1}|\partial S_i(G_2,g)|+n_1n_2\\&\geq \frac{1}{2}n_1(n_1-1)+vs_{sum}(G_2)+n_1n_2.\end{aligned}$$

Finally,

$$vs_{sum}(G_1\times G_2)\geq \min\left\{\frac{1}{2}n_1(n_1-1)+vs_{sum}(G_2),\frac{1}{2}n_2(n_2-1)+vs_{sum}(G_1)\right\}+n_1n_2.$$

For the other direction, let $f$ be an arbitrary layout of $G_1$ and let $g$ be a layout of $G_2$ such that $vs_{sum}(G_2)=\sum_{i=1}^{n_2-1}|\partial S_i(G_2,g)|$. Define the layout $h$ of $G_1\times G_2$ by the rule

$$h(v)=\begin{cases}f(v), & \text{if } v\in V(G_1),\\ n_1+g(v), & \text{if } v\in V(G_2).\end{cases}$$

It follows easily that $|\partial S_i(G_1\times G_2,h)|=i$ whenever $i\in\{1,\ldots,n_1\}$ and that $|\partial S_i(G_1\times G_2,h)|=n_1+|\partial S_{i-n_1}(G_2,g)|$ if $i\in\{n_1+1,\ldots,n_1+n_2-1\}$. We conclude that

$$\begin{aligned}vs_{sum}(G_1\times G_2)&\leq \sum_{i=1}^{n_1+n_2-1}|\partial S_i(G_1\times G_2,h)|\\&=\sum_{i=1}^{n_1}i+\sum_{i=n_1+1}^{n_1+n_2-1}(n_1+|\partial S_{i-n_1}(G_2,g)|)\\&=\frac{1}{2}n_1(n_1+1)+n_1(n_2-1)+\sum_{i=1}^{n_2-1}|\partial S_i(G_2,g)|\\&=\frac{1}{2}n_1(n_1-1)+vs_{sum}(G_2)+n_1n_2.\end{aligned}$$

Similarly,

$$vs_{sum}(G_1\times G_2)\leq \frac{1}{2}n_2(n_2-1)+vs_{sum}(G_1)+n_1n_2.\qquad \square$$

**7. Cographs.** Theorem 6.1 can be used for obtaining linear time algorithms for the search cost and the corresponding search strategy on cographs. Recall that a graph $G$ is a cograph if and only if one of the following conditions is fulfilled:

1. $|V(G)|=1$.
2. There are cographs $G_1,\ldots,G_k$ and $G=G_1\dot\cup G_2\dot\cup\cdots\dot\cup G_k$.
3. There are cographs $G_1,\ldots,G_k$ and $G=G_1\times G_2\times\cdots\times G_k$.

(See [8] for references on cographs and different graph classes.)

Linear time algorithms computing the search cost and optimal search program on cographs follow rather straightforwardly from the theory developed in the preceding sections. A similar algorithm for the treewidth and the pathwidth of cographs was described in [6]. The main idea of the algorithms is in constructing a sequence of operations $\dot\cup$ and $\times$ producing the cograph $G$. With each cograph $G$ one can associate a binary labeled tree which is called the *cotree* $T_G$. $T_G$ has the following properties.

1. Each internal vertex $v$ of $T_G$ has label$(v)\in\{0,1\}$.
2. There is a bijection $\tau$ between the set of leaves of $T_G$ and $V(G)$.

3. To each vertex $v \in (T_G)$ we assign the subgraph $G_v$ of $G$ as follows.
   (a) If $v$ is a leaf, then $G_v = \tau(v)$.
   (b) If $v$ is an internal vertex and label$(v)= 0$, then $G_v = G_u \dot{\cup} G_w$, where $u, w$ are the sons of $v$.
   (c) If $v$ is an internal vertex and label$(v)= 1$, then $G_v = G_u \times G_w$, where $u, w$ are the sons of $v$.

Notice that if $r$ is the root of $T_G$, then $G_r = G$. Corneil, Perl, and Stewart [11] gave an $O(|V(G)| + |E(G)|)$ algorithm for determining whether a given graph $G$ is a cograph and, if so, for constructing the corresponding cotree.

We omit the detailed proofs of the following theorems.

THEOREM 7.1. *The search cost of a cograph given with a corresponding cotree can be computed in $O(n)$ time.*

THEOREM 7.2. *Let $G$ be a cograph of $n$ vertices and $e$ edges. The optimal search program on $G$ can be constructed in $O(n + e)$ time.*

**8. Concluding remarks.** In this paper, we have introduced a game-theoretic approach to the problem of interval completion with the smallest number of edges. There are similar approaches to the pathwidth and treewidth parameters. The interesting problem is whether there is a graph-searching "interpretation" of the fill-in problem (see also [36] for monotonicity proofs of another variant of graph searching).

REFERENCES


[1] D. BIENSTOCK, *Graph searching, path-width, tree-width and related problems (a survey)*, DIMACS Ser. Discrete Mathematics and Theoretical Computer Science 5, Amer. Math. Soc., Providence, RI, 1991, pp. 33–49.

[2] D. BIENSTOCK AND P. SEYMOUR, *Monotonicity in graph searching*, J. Algorithms, 12 (1991), pp. 239–245.

[3] A. BILLIONNET, *On interval graphs and matrice profiles*, RAIRO Rech. Opér., 20 (1986), pp. 245–256.

[4] H. L. BODLAENDER, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci., 209 (1998), pp. 1–45.

[5] H. L. BODLAENDER, R. G. DOWNEY, M. R. FELLOWS, M. T. HALLETT, AND H. T. WAREHAM, *Parameterized complexity analysis in computational biology*, Computer Applications in the Biosciences, 11 (1995), pp. 49–57.

[6] H. L. BODLAENDER AND R. H. MÖHRING, *The pathwidth and treewidth of cographs*, SIAM J. Discrete Math., 6 (1993), pp. 181–188.

[7] J. A. BONDY, *Basic graph theory: Paths and circuits*, in Handbook of Combinatorics, 1, R. L. Graham, M. Grötschel, and L. Lovász, eds., Elsevier, Amsterdam, 1995, pp. 3–110.

[8] A. BRANDSTÄDT, V. B. LE, and J. P. SPINRAD, *Graph classes: A survey*, SIAM Monogr. Discrete Math. Appl., SIAM, Philadelphia, PA, 1999.

[9] L. CAI, *Fixed-parameter tractability of graph modification problems for hereditary properties*, Technical report, Department of Computer Science, The Chinese University of Hong Kong, Shatin New Territories, Hong Kong, 1995.

[10] P. Z. CHINN, J. CHVÁTALOVÁ, A. K. DEWDNEY, AND N. E. GIBBS, *The bandwidth problem for graphs and matrices—a survey*, J. Graph Theory, 6 (1982), pp. 223–254.

[11] D. G. CORNEIL, Y. PERL, AND L. K. STEWART, *A linear recognition algorithm for cographs*, SIAM J. Comput., 14 (1985), pp. 926–934.

[12] N. D. DENDRIS, L. M. KIROUSIS, AND D. M. THILIKOS, *Fugitive-search games on graphs and related parameters*, Theoret. Comput. Sci., 172 (1997), pp. 233–254.

[13] J. A. ELLIS, I. H. SUDBOROUGH, AND J. TURNER, *The vertex separation and search number of a graph*, Inform. and Comput., 113 (1994), pp. 50–79.

[14] F. V. FOMIN AND N. N. PETROV, *Pursuit-evasion and search problems on graphs*, Congr. Numer., 122 (1996), pp. 47–58.

[15] M. FRANKLIN, Z. GALIL, AND M. YUNG, *Eavesdropping games: A graph-theoretic approach to privacy in distributed systems*, in Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, 1993, pp. 670–679.

[16] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.
[17] P. A. Golovach, *Node-search and search number of a combination of graphs*, Vestnik Leningrad. Univ. Math. Mekh. Astronom., 1990, OVYR vyp. 2, 90–91, 122 (in Russian); translation in Vestnik Leningrad. Univ. Math., 23 (1990), pp. 53–55.
[18] P. A. Golovach, *Vertex separation sum of a graph*, Diskret. Mat., 9 (1997), pp. 86–91 (in Russian).
[19] H. Kaplan, R. Shamir, and R. E. Tarjan, *Tractability of parameterized completion problems on chordal and interval graphs: Minimum fill-in and physical mapping*, in Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 1994, pp. 780–791.
[20] L. M. Kirousis and C. H. Papadimitriou, *Interval graphs and searching*, Discrete Math., 55 (1985), pp. 181–184.
[21] L. M. Kirousis and C. H. Papadimitriou, *Searching and pebbling*, Theoret. Comput. Sci., 47 (1986), pp. 205–218.
[22] D. Kuo and G. J. Chang, *The profile minimization problem in trees*, SIAM J. Comput., 23 (1994), pp. 71–81.
[23] A. S. LaPaugh, *Recontamination does not help to search a graph*, J. ACM, 40 (1993), pp. 224–245.
[24] T. Lengauer, *Black-white pebbles and graph separation*, Acta Inform., 16 (1981), pp. 465–475.
[25] F. S. Makedon, C. H. Papadimitriou, and I. H. Sudborough, *Topological bandwidth*, SIAM J. Algebraic Discrete Methods, 6 (1985), pp. 418–444.
[26] F. S. Makedon and I. H. Sudborough, *On minimizing width in linear layouts*, Discrete Appl. Math., 23 (1989), pp. 243–265.
[27] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou, *The complexity of searching a graph*, J. ACM, 35 (1988), pp. 18–44.
[28] R. H. Möhring, *Graph problems related to gate matrix layout and PLA folding*, in Computational Graph Theory, Computing Suppl. 7, E. Mayr, H. Noltemeier, and M. Sysło, eds., Springer-Verlag, Berlin, 1990, pp. 17–51.
[29] T. D. Parsons, *Pursuit-evasion in a graph*, in Theory and Application of Graphs, Y. Alavi and D. R. Lick, eds., Springer Verlag, Berlin, 1976, pp. 426–441.
[30] N. N. Petrov, *Some extremal search problems on graphs*, Differential Equations, 18 (1982), pp. 591–595. Translation from Differ. Uravn 18, (1982) pp. 821–827.
[31] N. N. Petrov, *Pursuit problems with no information concerning the evader*, Differential Equations, 18 (1983), pp. 944–948. Translation from Differ. Uravn., 18 (1982), pp. 1345–1352.
[32] B. Reed, *Treewidth and tangles: A new connectivity measure and some applications*, in Surveys in Combinatorics, R. A. Bailey, ed., Cambridge University Press, Cambridge, UK, 1997, pp. 87–162.
[33] N. Robertson and P. D. Seymour, *Graph minors—a survey*, in Surveys in Combinatorics, I. Anderson, ed., Cambridge University Press, Cambridge, UK, 1985, pp. 153–171.
[34] K. Sugihara and I. Suzuki, *Optimal algorithms for a pursuit-evasion problem in grids*, SIAM J. Discrete Math., 2 (1989), pp. 126–143.
[35] R. Thomas, *Tree decompositions of graphs*, Lecture notes, Georgia Institute of Technology, Atlanta, GA, 1996.
[36] Y. S. Stamatiou and D. M. Thilikos, *Monotonocity and Inert Fugitive Search Games*, Technical Report LSI-99-35-R, Departament de Llenguatges i Sistemes Informátics, Universitat Politécnica de Catalunya, Spain, 1999.