JOURNAL OF ALGORITHMS 12, 239–245 (1991)

Monotonicity in Graph Searching

D. BIENSTOCK* AND PAUL SEYMOUR

Bellcore, 445 South Street, Morristown, New Jersey 07960

Received May 1988; accepted November 1990

We give a new proof of the result, due to A. LaPaugh, that a graph may be optimally "searched" without clearing any edge twice. © 1991 Academic Press, Inc.

1. INTRODUCTION

Let us regard a graph as a system of tunnels containing a (lucky, invisible, fast) fugitive. We desire to capture this fugitive by "searching" all edges of the graph, in a sequence of discrete steps, while using the fewest possible "guards." This problem was introduced by Breisch [2] and Parsons [6]. In the version of graph searching considered in [5] (which we call *edge-searching*, using terminology from [3]) a search step consists of placing a guard at a vertex, or removing a guard from a vertex, or sliding a guard along an edge. Further, an edge $\{u, v\}$ is *cleared* by sliding a guard from u to v, while shielding u from contaminated (that is, uncleared) edges with appropriately placed guards (for example, by keeping another guard at u). If, at any point in time, there is a path from a contaminated edge e to a cleared edge e' that is not blocked by guards, e' becomes instantaneously recontaminated and must be cleared again. Our objective is to reach a state in which all edges are simultaneously cleared, so that the maximum number of guards used at any step is minimized. Any strategy that achieves this result is called *optimal*, and the optimal number of guards is the *edge-search number* of the graph.

LaPaugh [5] proved that there always exists an optimal strategy that is monotone (without recontamination). One implication of this important result is that there is an optimal strategy that terminates after a linear number of steps.

*Current address: Dept. of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027.

0196-6774/91 $3.00

Kirousis and Papadimitriou [3, 4] introduced a variant of searching called *node-searching*. In this version an edge may be declared cleared as soon as there is a guard at each endpoint. By reducing node-searching to edge-searching, they proved that for node-searching there also exists a monotone optimal strategy.

We introduce a new version of searching, called *mixed*, that combines features of both edge- and node-searching. In our version, an edge is cleared *either* by sliding or by placing guards at each end. We insist that at most one edge is cleared at any step, both for mixed- and node-searching (if placing a guard at a vertex simultaneously clears many edges, then we break up this process into several steps, each of which clears one of the edges, in arbitrary order). More precisely, a *mixed-search* in G is a sequence of pairs

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

(intuitively, Z_i is the set of vertices occupied by guards immediately before the $(i+1)$th step, and A_i is the set of clear edges) such that

(i) for $0 \leq 1 \leq n$, $A_i \subseteq E(G)$ and $Z_i \subseteq V(G)$

(ii) for $0 \leq i \leq n$, any vertex incident with an edge in A_i and with an edge in $E(G) - A_i$ belongs to Z_i

(iii) $A_0 = \emptyset$, $A_n = E(G)$

(iv) for $1 \leq i \leq n$, either

(a) (*placing new guards*) $Z_i \supseteq Z_{i-1}$ and $A_i = A_{i-1}$, or

(b) (*removing guards*) $Z_i \subseteq Z_{i-1}$ and A_i is the set of all edges e such that every path containing e and an edge of $E(G) - A_{i-1}$ has an internal vertex in Z_i, and in particular $A_i \subseteq A_{i-1}$, or

(c) (*node searching* e) $Z_i = Z_{i-1}$, and $A_i = A_{i-1} \cup \{e\}$ for some edge $e \in E(G) - A_{i-1}$ with both ends in Z_{i-1}, or

(d) (*edge-searching* e) $Z_i = (Z_{i-1} - \{u\}) \cup \{v\}$ for some $u \in Z_{i-1}$ and $v \in V(G) - Z_{i-1}$, and there is an edge $e \in E(G) - A_{i-1}$ with ends u, v, and every other edge incident with u belongs to A_{i-1}, and $A_i = A_{i-1} \cup \{e\}$.

A node-search is defined similarly, except that (d) is deleted. To formalize an edge-search in this notation, we must permit Z_i to be a multiset instead of a set, because it may be important that two guards occupy the same vertex (for perhaps one is about to slide along some edge, and the other will remain to prevent recontamination). With this modification (in particular in (d), allowing other edges not in A_{i-1} to be incident with u, provided that u is currently occupied by at least two guards, one of whom is about to slide to v) an edge-search is defined by deleting (c).

We denote by es(G), ns(G), and ms(G) respectively, the edge-, node-, and mixed-search numbers of a graph G defined in the natural way. It follows easily that $\text{ms}(G) \leq \text{ns}(G), \text{es}(G)$.

Now given a graph G, let us construct graphs G^{e} and G^{n} by replacing each edge of G by two edges in series, or two edges in parallel, respectively. Then it is not difficult to show that $\text{es}(G) = \text{ms}(G^{\text{e}})$, and $\text{ns}(G) = \text{ms}(G^{\text{n}})$ (for completeness we prove the first equality in Section 3). Further, a monotone optimal mixed-search strategy on G^{e} and G^{n} can be converted to a monotone optimal edge- or node-search, respectively, in G. Hence our monotonicity result for mixed-searching will imply monotonicity for both edge- and node-searching, and hence implies the theorems of LaPaugh [5] and Kirousis and Papadimitriou [3, 4]. We remark that a similar proof can be constructed directly for, say, edge-searching, but working with mixed-searching is not only more general but also easier.

2. Crusades

Let G be a graph. If $X \subseteq E(G)$ we let $\delta(X)$ be the set of vertices which are endpoints of an edge in X and also of an edge in $E(G) - X$. Notice that $\delta(X) = \delta(E(G) - X)$. Also, $|\delta|$ satisfies the submodular inequality

$$|\delta(X \cap Y)| + |\delta(X \cup Y)| \leq |\delta(X)| + |\delta(Y)|$$

for any X, Y, because each vertex counted in the left-hand side is also counted at least as many times in the right-hand side.

A *crusade* in G is a sequence $(X_0, X_1, \ldots, X_n)$ of subsets of $E(G)$, such that $X_0 = \emptyset$, $X_n = E(G)$, and $|X_i - X_{i-1}| \leq 1$ for $1 \leq i \leq n$. The crusade *uses* $\leq k$ *guards* if $|\delta(X_i)| \leq k$ for $0 \leq i \leq n$.

(2.1) *If* $\text{ms}(G) \leq k$ *then there is a crusade in G using* $\leq k$ *guards*.

Proof. Let $(A_0, Z_0), \ldots, (A_n, Z_n)$ be a mixed-search in G with each $|Z_i| \leq k$. Then each $\delta(A_i) \subseteq Z_i$ and hence each $|\delta(A_i)| \leq k$, and so $(A_0, A_1, \ldots, A_n)$ is a crusade in G using $\leq k$ guards. □

The converse of (2.1) is true (with some exceptions) but not obvious, and is a corollary of our main result. The reason for using crusades is that it is very easy to prove a monotonicity result for them, as follows. A crusade $(X_0, X_1, \ldots, X_n)$ is *progressive* if $X_0 \subseteq X_1 \subseteq \cdots \subseteq X_n$ and $|X_i - X_{i-1}| = 1$ for $1 \leq i \leq n$.

(2.2) *Suppose that there is a crusade in G using $\leq k$ guards. Then there is a progressive crusade in G using $\leq k$ guards.*

Proof. Choose a crusade $(X_0, X_1, \ldots, X_n)$ using $\leq k$ guards, such that

(1) $\Sigma_{0 \leq i \leq n}(|\delta(X_i)| + 1)$ *is minimum*

and, subject to (1),

(2) $\Sigma_{0 \leq i \leq n}|X_i|$ *is minimum.*

We shall show that $(X_0, X_1, \ldots, X_n)$ is progressive. For choose j with $1 \leq j \leq n$.

(3) $|X_j - X_{j-1}| = 1$.

For $|X_j - X_{j-1}| \leq 1$, and if $X_j \subseteq X_{j-1}$ then $(X_0, X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_n)$ is a crusade contradicting (1).

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

For otherwise $(X_0, X_1, \ldots, X_{j-1}, X_{j-1} \cup X_j, X_{j+1}, \ldots, X_n)$ is a crusade contradicting (1).

(5) $X_{j-1} \subseteq X_j$.

For from the submodularity of $|\delta|$, we have that

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

From (4), it follows that $|\delta(X_{j-1} \cap X_j)| \leq |\delta(X_{j-1})|$. Hence

$$(X_0, X_1, \ldots, X_{j-2}, X_{j-1} \cap X_j, X_j, X_{j+1}, \ldots, X_n)$$

is a crusade using $\leq k$ guards. From (2), $|X_{j-1} \cap X_j| \geq |X_j|$, and the claim follows.

From (3) and (5) we deduce that $(X_0, X_1, \ldots, X_n)$ is progressive. □

Actually (2.2) is a corollary of Theorem (3.2) of [7], but we have given the proof in full for completeness.

(2.3) *Let G be such that every vertex is incident with at least two edges. Let $(X_0, X_1, \ldots, X_n)$ be a progressive crusade in G using $\leq k$ guards, and for $1 \leq i \leq n$ let $X_i - X_{i-1} = \{e_i\}$. Then there is a monotone mixed-search of G using $\leq k$ guards, such that the edges of G are searched in the order $e_1, e_2, \ldots, e_n$.*

Proof. We construct the mixed-search inductively. Suppose then that $1 \leq j \leq n$, and we have succeeded in clearing edges $e_1, \ldots, e_{j-1}$ in order, in such a way that no other edges have been cleared yet. Let A be the set of all vertices $v \in V(G)$ such that every edge incident with v is in X_{j-1}. Certainly each vertex in $\delta(X_{j-1})$ is currently occupied by a guard, because it is incident both with a clear and with a contaminated edge. Remove all other guards (no recontamination occurs). Since $e_j \notin X_{j-1}$, it has no end in A. Let N be the set of ends of e_j. If $|N \cup \delta(X_{j-1})| \leq k$, we may place new guards on the ends of e_j and declare it cleared. We assume then that $|N \cup \delta(X_{j-1})| > k$. Since $|\delta(X_{j-1})| \leq k$ it follows that $N \not\subseteq \delta(X_{j-1})$. Choose $v \in N - \delta(X_{j-1})$. Moreover, $N - \delta(X_{j-1}) \subseteq \delta(X_j)$, because every vertex of G is incident with ≥ 2 edges; it follows that $\delta(X_{j-1}) \not\subseteq \delta(X_j)$. Choose $u \in \delta(X_{j-1}) - \delta(X_j)$. Then $u \in N$, and e_j has ends u, v and u is incident with no edge in $E(G) - X_{j-1}$ except e_j. Thus, we can clear e_j by sliding the guard at u along e_j to v. □

A proof can also be given in terms of our formal definition of a mixed search as a sequence $(A_0, Z_0), \ldots, (A_n, Z_n)$, but it is rather opaque, and we prefer the "informal" proof given.

We deduce:

(2.4) *If there is a mixed-search of G using $\leq k$ guards, then there is a monotone mixed-search using $\leq k$ guards.*

Proof. We may assume that G has no isolated vertices. Let G' be obtained from G by adding a loop at each vertex; then $\text{ms}(G') = \text{ms}(G) \leq k$. By (2.1) there is a crusade in G' using $\leq k$ guards. By (2.2) there is a progressive crusade in G' using $\leq k$ guards. By (2.3) there is a monotone mixed-search in G' using $\leq k$ guards. Hence there is such a mixed-search in G, as required. □

In summary, we have shown

(2.5) *Let G be such that every vertex is incident with at least two edges. For $k \geq 0$, the following are equivalent*:

(i) $\text{ms}(G) \leq k$
(ii) there is a crusade in G using $\leq k$ guards
(iii) there is a progressive crusade in G using $\leq k$ guards
(iv) there is a monotone mixed-search in G using $\leq k$ guards.

In particular, the converse of (2.1) holds if G has minimum degree ≥ 2. It does not always hold without this condition. For example, in general a caterpillar (a tree where the vertices of degree > 1 form a path) has mixed-search number 2 and yet has a crusade with only one guard.

3. Edge- and Node-Searching

We claimed in Section 1 that LaPaugh's theorem on edge-searching and the Kirousis–Papadimitriou theorem on node-searching both follow from

(2.4). We show the former as follows. Let G^e be obtained from G by replacing each edge by two edges in series. To show that there is a monotone edge-search of G using $\text{es}(G)$ guards, it suffices, in view of (2.4), to show that $\text{es}(G) \geq \text{ms}(G^e)$, and that a monotone mixed-search of G^e yields a monotone edge-search of G. We begin with

(3.1) $\text{es}(G^e) \leq \text{es}(G)$, *and a monotone edge-search of* G^e *can be converted to a monotone edge-search of G using the same number of guards.*

Proof. Clearly $\text{es}(G^e) \leq \text{es}(G)$, for any edge-search of G can be converted easily to an edge-search of G^e. (In fact equality holds, but we do not need that). Now G^e has two edges in series for each edge f of G; let these edges be f', f'' say. Let H be obtained from G^e by contracting all the edges f'' $(f \in E(G))$; then H is isomorphic to G. Each vertex v of H is obtained from G^e by identifying a subset C_v of vertices of G^e, under contraction. Given an edge-search of G^e, we convert it to an edge-search of H by the rule that: if at the ith step guard s is at vertex $u \in V(G^e)$, then at the ith step of the new search, guard s is at vertex $v \in V(H)$, where $u \in C_v$. This satisfies our requirements. □

We deduce from (3.1) that $\text{ms}(G^e) \leq \text{es}(G)$, for certainly $\text{ms}(G^e) \leq \text{es}(G^e)$. To prove that a monotone mixed-search of G^e can be converted to a monotone edge-search of G, it suffices in view of (3.1) to prove the following (for then we may apply (3.2) to G^e).

(3.2) *Let G be such that every edge has an end incident with precisely one other edge. Then a monotone mixed-search of G using* $\leq k$ *guards can be converted to a monotone edge-search of G using* $\leq k$ *guards.*

Proof. Let us choose a monotone mixed-search using $\leq k$ guards, such that as many edges as possible are cleared by sliding. Suppose, for a contradiction, that some edge f is not cleared by sliding. Let f be cleared in the ith step; thus, immediately before the ith step there are guards at both ends u, v of f. Let v be incident with a unique other edge g. We shall replace the ith step of the search by three other steps, as follows. If g is still contaminated immediately before the ith step, we shall clear f by: remove the guard from v and place him at u; and then slide him along f to v. If g is clear immediately before the ith step, we shall clear f by: slide the guard at v along f to u; then remove him from u and place him at v. In either case we search f by sliding, and can continue the mixed-search as before. This contradicts the choice of the mixed-search. It follows that every edge is cleared by sliding, as required. □

From (3.1), (3.2), and (2.4), we deduce LaPaugh's theorem. One can deduce the Kirousis–Papadimitriou theorem similarly. A direct proof of the latter is given in [1]. A monotonicity theorem for another searching problem (cornering a fugitive that we can see) is given in [8].

References

1. D. Bienstock, N. Robertson, P. D. Seymour, and R. Thomas, Quickly excluding a forest, *J. Combin. Theory Ser. B*, to appear.
2. R. L. Breisch, An intuitive approach to speleotopology, *Southwestern Cavers* (published by the Southwestern Region of the National Speleological Society) **6**, No. 5 (1967), 72–78.
3. L. M. Kirousis and C. H. Papadimitriou, Searching and pebbling, *J. Theoret. Comput. Sci.* **47** (1986), 205–218.
4. L. M. Kirousis, and C. H. Papadimitriou, Interval graphs and searching, *Discrete Math.* **55** (1985), 181–184.
5. A. LaPaugh, "Recontamination Does Not Help to Search a Graph," Tech. Report 335, Dept. of Elec. Eng. and Comput. Sci., Princeton University, 1982.
6. T. D. Parsons, Pursuit-evasion in a graph, *in* "Theory and Application of Graphs" (Y. Alavi and D. R. Lick, Eds.), New York/Berlin, pp. 426–441, Springer-Verlag, 1976.
7. N. Robertson and P. D. Seymour, Graph minors. X. Obstructions to tree-decomposition, *J. Combin. Theory Ser. B*, to appear.
8. P. D. Seymour and R. Thomas, Graph searching, and a minimax theorem for tree-width, submitted (manuscript 1988).