

Directed Searching Digraphs: Monotonicity and Complexity

Boting Yang¹ and Yi Cao²

¹ Department of Computer Science, University of Regina
boting@cs.uregina.ca

² Department of Computing Science, University of Alberta
cao1@cs.ualberta.ca

Abstract. In this paper, we introduce and study two new search models on digraphs: the directed searching and mixed directed searching. In these two search models, both searchers and intruders must follow the edge directions when they move along edges. We prove the monotonicity of both search models, and we show that both directed and mixed directed search problems are NP-complete.

1 Introduction

Many real-world problems can be naturally modeled by graph search problems. Examples include: capturing intruders in a building, clearing a complex system of interconnected pipes which is contaminated by some noxious gas, and killing a computer virus in a network system. The meaning of a cleared or contaminated edge varies with the problems. For example, in the problem of capturing intruders, a cleared edge means that there is no intruder hiding along this edge, while a contaminated edge means that there may be some intruders hiding along this edge.

In general, a *graph or digraph search problem* is to find the minimum number of searchers required to capture all the intruders hiding in a graph or digraph. In the edge search problem introduced in [9], there are three types of actions for searchers, i.e., placing, removing and sliding, and an edge is cleared only by a sliding action in a proper way. In the node search problem introduced in [7], there are only two types of actions for searchers, i.e., placing and removing, and an edge is cleared if both end vertices are occupied by searchers. Kirousis and Papadimitriou [7] showed that the node search number is equal to the pathwidth plus one. Bienstock and Seymour [4] introduced the mixed search problem that combines the edge search and node search problems. Thus, in the mixed search problem, an edge is cleared if both end vertices are occupied by searchers or cleared by a sliding action in a proper way. In these three graph search problems, intruders are invisible and they can move along a path that contains no searchers at a great speed at any time. Seymour and Thomas [13] introduced another variant of the graph search problem in which an intruder hiding in the graph is visible to searchers. They showed that the search number of this variant is equal to the treewidth plus one.

When studying search problems from a computational complexity viewpoint, we are interested in deciding the search number of a graph. Megiddo et al. [9] showed that the edge search problem is NP-hard. This problem belongs to the NP class follows from the monotonicity result of [8] in which LaPaugh showed that recontamination of edges cannot reduce the number of searchers needed to clear a graph. Monotonicity is a very important issue in graph search problems. Bienstock and Seymour [4] proposed a method that gives a succinct proof for the monotonicity of the mixed search problem, which implies the monotonicity of the edge search problem and the node search problem. Fomin and Thilikos [5] provided a general framework that can unify monotonicity results in a unique minmax theorem.

An undirected graph is not always sufficient in representing all the information of a real-world problem. For example, directed edges are required if the graph models one-way streets in a road system. Johnson et al. [6] generalized the concepts of tree-decomposition and treewidth to digraphs and introduced a digraph search problem accordingly. Reed [11] defined another treewidth on digraphs. Safari [12] introduced a new parameter of digraphs, d -width, which is related to the directed treewidth of a digraph. Barat [2] generalized the cops-and-robber game to digraphs. He proved that an optimal monotonic search strategy for a digraph needs at most one more searcher than the search number of the digraph and he conjectured that the monotonicity is held for the cops-and-robber game on digraphs. Yang and Cao [14] introduced the strong searching model in which the intruder must follow the edge directions but searchers need not when they move along edges. Yang and Cao [15] also introduced the weak searching model in which searchers must follow the edge directions but the intruder need not when they move along edges. In [16] Yang and Cao introduced the directed vertex separation and investigated the relations between different digraph searching models, directed vertex separation, and directed pathwidth. In the digraph searching models in [14,15], there are three types of actions for searchers: placing, removing and sliding. Alspach et al. [1] proposed four digraph search models in which searchers cannot be removed from digraphs.

Throughout this paper, we use D to denote a digraph, (u, v) to denote a directed edge with tail u and head v , and $u \rightsquigarrow v$ to denote a directed path from u to v . All graphs and digraphs in this paper contain at least one edge.

A natural generalization of the edge search model is the directed search model. In the *directed search model*, both searchers and intruders must move in the edge directions. Initially, all edges of D are *contaminated*. Each intruder can move from vertex u to vertex v along a directed path $u \rightsquigarrow v$ that contains no searchers at a great speed at any time. There are three types of actions for searchers: (1) placing a searcher on a vertex, (2) removing a searcher from a vertex, and (3) sliding a searcher along an edge from its tail to its head. A *directed search strategy* is a sequence of actions such that the final action leaves all edges of D *uncontaminated* (or *cleared*). A contaminated edge (u, v) can be cleared in one of two ways by a sliding action: (1) sliding a searcher from u to v along (u, v) while at least one searcher is located on u and (2) sliding a searcher from u to v along

(u, v) while all edges with head u are already cleared. The digraph D is *cleared* if all of the edges are cleared. The minimum number of searchers needed to clear D in the directed search model is the *directed search number* of D , denoted by $ds(D)$.

A generalization of the mixed search model is the mixed directed search model, which can be considered as a kind of combination of the directed search model and node search model. We will give a precise definition of this search model in the next section.

In the directed or mixed directed search model, a cleared edge will be recontaminated if there is a directed path from the head of a contaminated edge to the tail of this cleared edge such that there is no searcher stationing on any vertex of this path.

We say that a vertex in D is *occupied* at some moment if at least one searcher is located on this vertex at this moment. Any searcher that is not on D at some moment is said *free* at this moment.

In the directed or mixed directed search model, let S be a search strategy and let A_i be the set of cleared edges immediately after the i th action. S is *monotonic* if $A_i \subseteq A_{i+1}$ for each i . We say that this search model has the property of *monotonicity* (or is *monotonic*) if for any digraph D , there exists a monotonic search strategy that can clear D using k searchers, where k is the search number of D in this search model.

This paper is organized as follows. In Section 2, we prove the monotonicity of the mixed directed search model. In Section 3, we prove the monotonicity of the directed search model. In Section 4, we show the NP-completeness results for both directed and mixed directed search problems. Finally, we conclude this paper in Section 5.

2 Monotonicity of the Mixed Directed Search Model

We will show the monotonicity of the mixed directed search model in this section, which means that recontamination does not help to reduce the mixed directed search number of a digraph. We will extend the method proposed by Bienstock and Seymour [4]. We first give the definition of a critical vertex.

Definition 1. Let D be a digraph. For an edge set $X \subseteq E(D)$, a vertex in $V(D)$ is *critical* if it is the tail of an edge in X and the head of an edge in $E(D) - X$. The set of all critical vertices in $V(D)$ is denoted by $\delta(X)$.

We then define the campaign and the progressive campaign.

Definition 2. Let D be a digraph. A *campaign* in D is a sequence (X_0, X_1, \dots, X_n) of subsets of $E(D)$ such that $X_0 = \emptyset$, $X_n = E(D)$ and $|X_i - X_{i-1}| \leq 1$, for $1 \leq i \leq n$. The *width* of the campaign is defined as $\max_{0 \leq i \leq n} |\delta(X_i)|$. A campaign is *progressive* if $X_0 \subseteq X_1 \subseteq \dots \subseteq X_n$ and $|X_i - X_{i-1}| = 1$, for $1 \leq i \leq n$.

Similar to [2] and [4], we have the following lemma for the progressive campaign.

Lemma 1. *If there is a campaign in D of width at most k , then there is a progressive campaign in D of width at most k .*

In the remainder of this section, we will prove that the mixed directed search problem is monotonic. The *mixed directed search model* can be obtained by modifying the directed search model as follows. Recall that there are two ways to clear an edge by a sliding action in the directed search model. In the mixed directed search model, we replace the first way by the *node-search-clearing* rule: an edge can be cleared if both of its end vertices are occupied. Another modification is to disallow the recontamination caused by a sliding action, that is, the action of sliding a searcher from u to v along edge (u, v) changes the state of edge (u, v) from contaminated to clear, but it does not change the state of any other edge. Thus, in a mixed directed search strategy, each sliding along an edge must clear this edge if it is contaminated. More precisely, we define the four types of actions in the mixed directed search model as follows.

Definition 3. Let $S = (s_1, s_2, \dots, s_n)$ be a mixed directed search strategy for a digraph D . For $1 \leq i \leq n$, let A_i be the set of cleared edges and Z_i be the set of occupied vertices immediately after action s_i such that $\delta(A_i) \subseteq Z_i$. Let $A_0 = Z_0 = \emptyset$. Each action s_i , $1 \leq i \leq n$, is one of the following four types:

- (a) (*placing a searcher on v*) $Z_i = Z_{i-1} \cup \{v\}$ for some vertex $v \in V(D) - Z_{i-1}$ and $A_i = A_{i-1}$ (note that each vertex in Z_i has exactly one searcher);
- (b) (*removing the searcher from v*) $Z_i = Z_{i-1} - \{v\}$ for some vertex $v \in Z_{i-1}$ and $A_i = \{e \in A_{i-1} : \text{if there is a directed path } u \rightsquigarrow w \text{ containing } e \text{ and an edge } e' \in E(D) - A_{i-1} \text{ such that } w \text{ is the head of } e \text{ and } u \text{ is the tail of } e', \text{ then } u \rightsquigarrow w \text{ has an internal vertex in } Z_i\}$;
- (c) (*node-search-clearing e*) $Z_i = Z_{i-1}$ and $A_i = A_{i-1} \cup \{e\}$ for some edge $e = (u, v) \in E(D)$ with both ends u and v in Z_{i-1} ;
- (d) (*edge-search-clearing e*) $Z_i = (Z_{i-1} - \{u\}) \cup \{v\}$ and $A_i = A_{i-1} \cup \{e\}$ for some edge $e = (u, v) \in E(D)$ with $u \in Z_{i-1}$ and $v \in V(D) - Z_{i-1}$ and every (possibly 0) edge with head u belongs to A_{i-1} .

From Definition 3, we know that at most one edge can be cleared in one action and each vertex is occupied by at most one searcher at any time. Note that recontamination in the mixed directed search problem is caused only by removing actions. In (c) and (d), if $e \in A_{i-1}$, then we say this action is *superfluous*. Adding or deleting superfluous actions will not affect the number of searchers used in a search strategy, however, sometimes allowing superfluous actions may make arguments simple.

The *mixed directed search number* of a digraph D , denoted by $\text{mds}(D)$, is the minimum number of searchers needed to clear D in the mixed directed search model. The following lemma reveals the relationship between the mixed directed searching of D and a campaign in D .

Lemma 2. *Let D be a digraph without any multiple edges. If $\text{mds}(D) \leq k$, then there is a campaign in D of width at most $k - 1$.*

Proof. Let $S = (s_1, s_2, \dots, s_m)$ be a mixed directed search strategy of D using at most k searchers. For $1 \leq i \leq m$, let A_i be the set of cleared edges and Z_i be the set of occupied vertices immediately after s_i , and let $A_0 = Z_0 = \emptyset$. We first normalize S such that the normalized search strategy can also clear D using at most k searchers. The normalized search strategy may contain some superfluous actions, but this will not increase the number of searchers required to clear D .

The normalization is conducted by inserting some node-search-clearing actions after each placing action and edge-search-clearing action. Specifically, for each $1 \leq i \leq m$, if $Z_i - Z_{i-1}$ is empty, i.e., s_i is a removing or node-search-clearing action, then we leave s_i unchanged; otherwise, $Z_i - Z_{i-1}$ contains a vertex v , i.e., s_i is a placing or edge-search-clearing action such that v is occupied, then let $E_v^1 = \{(u, v) \in E(D) : u \in Z_{i-1}\}$, $E_v^2 = \{(v, u) \in E(D) : u \in Z_{i-1} \text{ and all edges with head } u \text{ except } (v, u) \text{ are in } A_{i-1}\}$, and $E_v^3 = \{(v, u) \in E(D) : u \in Z_{i-1} \text{ and } (v, u) \notin E_v^2\}$, and we then insert a subsequence of node-search-clearing actions between s_i and s_{i+1} , such that each edge in E_v^1 is cleared first, then each edge in E_v^2 , and finally each edge in E_v^3 (edges in the same set are cleared in an arbitrary order). After the normalization, we obtain a new sequence of actions that contains each old action and some new node-search-clearing actions. It is easy to see that this new sequence of actions, denoted by $(s'_1, s'_2, \dots, s'_n)$, is still a mixed directed search strategy of D using at most k searchers.

For $1 \leq i \leq n$, let A'_i be the set of cleared edges and Z'_i be the set of occupied vertices immediately after s'_i , and let $A'_0 = Z'_0 = \emptyset$. Since $\delta(A'_i) \subseteq Z'_i$, $|Z'_i| \leq k$, and $|A'_i - A'_{i-1}| \leq 1$, $1 \leq i \leq n$, we know that $(A'_0, A'_1, \dots, A'_n)$ is a campaign in D of width at most k .

We now show that the campaign $(A'_0, A'_1, \dots, A'_n)$ can be converted into a campaign (X_0, X_1, \dots, X_n) of width at most $k - 1$. For each i from 0 to n , if $|\delta(A'_i)| \leq k - 1$, then let $X_i = A'_i$. If $|\delta(A'_i)| = k$, then $\delta(A'_i) = Z'_i$. Let v be the last vertex occupied by a searcher in Z'_i . Recall that just after v receives a searcher in a placing or edge-search-clearing action, the following actions clear all edges in E_v^1, E_v^2 and E_v^3 by node-search-clearing. Note that at the step when an edge $(u, v) \in E_v^1$ is cleared, v is not a critical vertex at this step. When an edge $(v, u) \in E_v^2$ is cleared, since D has no multiple edges, all edges with head u are cleared and thus u is not a critical vertex. Hence, when $|\delta(A'_i)| = k$, s'_i must be a node-search-clearing action that clears an edge in E_v^3 . Therefore, each vertex in Z'_i has at least one contaminated edge with tail not in Z'_i . Let s'_j be the first removing action after s'_i . Such an action must exist; otherwise, D will not be cleared because each vertex in Z'_i has at least one contaminated edge with tail not in Z'_i . Let $R = A'_{j-1} - A'_j$ and $X_p = A'_p - R$ for $i \leq p \leq j$. Since $|A'_p - A'_{p-1}| \leq 1$, $i \leq p \leq j$, we know that $|X_p - X_{p-1}| \leq 1$ for $1 \leq p \leq j$. Suppose that s'_j removes the searcher on w . Then all edges with tail w must be contaminated immediately after s'_j , which means that A'_j contains no edges with tail w . Hence, X_p contains no edges with tail w for $i \leq p \leq j$. Thus $w \notin \delta(X_p)$ and $|\delta(X_p)| \leq k - 1$ for $i \leq p \leq j$. We then consider A'_{j+1} and construct X_{j+1} . We can continue this process and finally we obtain a campaign (X_0, X_1, \dots, X_n) in D of width at most $k - 1$.

Lemma 3. *For a digraph D , if (X_0, X_1, \dots, X_n) is a progressive campaign in D of width at most $k - 1$, then there is a monotonic mixed directed search strategy that clears D using at most k searchers such that the edges of D are cleared in the order e_1, e_2, \dots, e_n , where $e_i = X_i - X_{i-1}$, $1 \leq i \leq n$.*

Proof. We construct the monotonic mixed directed search strategy inductively. Suppose that we have cleared the edges e_1, \dots, e_{j-1} , $2 \leq j \leq n$, in order, and that no other edges have been cleared yet. Let $e_j = (u, v)$ and $C_{j-1} = \{p \in V(D) : p \text{ has no in-edge or all in-edges of } p \text{ belong to } X_{j-1}\}$. Before we clear (u, v) , we may remove searchers such that each vertex in $\delta(X_{j-1})$ is occupied by a searcher and all other searchers are free. If $|\{u, v\} \cup \delta(X_{j-1})| \leq k$, we may place free searchers on both ends of e_j and execute node-search-clearing. Assume that $|\{u, v\} \cup \delta(X_{j-1})| > k$. Since $|\delta(X_{j-1})| \leq k - 1$, it follows that $|\delta(X_{j-1})| = k - 1$ and $\{u, v\} \cap \delta(X_{j-1}) = \emptyset$. Thus, we have one free searcher. We now prove that $u \in C_{j-1}$. If $u \notin C_{j-1}$, then $u \in \delta(X_j)$ and $|\delta(X_j)| = k$, which contradicts the condition that (X_0, X_1, \dots, X_n) has width at most $k - 1$. Thus, u has no contaminated in-edges and we can place the free searcher on u and then slide the searcher from u to v along (u, v) to clear e_j by edge-search-clearing.

From Lemmas 1, 2 and 3, we have the following result.

Lemma 4. *Given a digraph D that has no multiple edges, the following are equivalent:*

- (i) $\text{mds}(D) \leq k$;
- (ii) *there is a campaign in D of width at most $k - 1$;*
- (iii) *there is a progressive campaign in D of width at most $k - 1$; and*
- (iv) *there is a monotonic mixed directed search strategy that clears D using at most k searchers.*

From Lemma 4, we can prove the monotonicity of the mixed directed search model.

Theorem 1. *Given a digraph D , if $\text{mds}(D) = k$, then there is a monotonic mixed directed search strategy that clears D using k searchers.*

3 Monotonicity of the Directed Search Model

In Section 2, we have proved that the mixed directed search problem is monotonic. In this section we will prove that the monotonicity of the mixed directed search problem implies the monotonicity of the directed search problem. The following lemma describes a relationship between the directed searching and the mixed directed searching.

Lemma 5. *If D is a digraph, then $\text{ds}(D) - 1 \leq \text{mds}(D) \leq \text{ds}(D)$.*

The two equalities in Lemma 5 can be achieved. From Lemma 5, we know that the difference between $\text{ds}(D)$ and $\text{mds}(D)$ is not a fixed constant. It is not easy to use this lemma to prove the monotonicity of the directed search model. However, we can transform D into another digraph D^* such that $\text{ds}(D) = \text{mds}(D^*)$. Then we can use this relation to prove the monotonicity of the directed search model.

Theorem 2. *For a digraph D , let D^* be a digraph obtained from D by replacing each edge $(u, v) \in E(D)$ by two directed paths (u, v', v) and (u, v'', v) . For $(u, v) \in E(D)$, let $f_{(u,v)}^1 = (u, v')$, $f_{(u,v)}^2 = (v', v)$, $f_{(u,v)}^3 = (u, v'')$ and $f_{(u,v)}^4 = (v'', v)$. The following are equivalent:*

- (i) $\text{ds}(D) \leq k$;
- (ii) $\text{mds}(D^*) \leq k$;
- (iii) *there is a progressive campaign (X_0, X_1, \dots, X_n) in D^* of width at most $k - 1$ such that for each $(u, v) \in E(D)$, $m_1 < m_2$ and $m_3 < m_4$, where m_i , $1 \leq i \leq 4$, is the subscript of X_{m_i} that is the first set containing $f_{(u,v)}^i$;*
- (iv) *there is a monotonic mixed directed search strategy that clears D^* using at most k searchers such that for each $(u, v) \in E(D)$, $f_{(u,v)}^1$ is cleared before $f_{(u,v)}^2$ and $f_{(u,v)}^3$ is cleared before $f_{(u,v)}^4$; and*
- (v) *there is a monotonic directed search strategy that clears D using at most k searchers.*

Proof. (i) \Rightarrow (ii). Let (s_1, s_2, \dots, s_n) be a directed search strategy of D using at most k searchers. We will inductively construct a mixed directed search strategy $(S'_1, S'_2, \dots, S'_n)$ of D^* using at most k searchers, where S'_i is a subsequence of actions corresponding to s_i . Since s_1 is a placing action, let S'_1 be the same placing action. Suppose that we have constructed $S'_1, S'_2, \dots, S'_{j-1}$ such that the following two conditions are satisfied: (1) the set of occupied vertices immediately after s_{j-1} is the same as the set of occupied vertices immediately after the last action in S'_{j-1} , and (2) if $(u, v) \in E(D)$ is cleared immediately after s_{j-1} , then the corresponding four edges $f_{(u,v)}^i \in E(D^*)$, $1 \leq i \leq 4$, are also cleared immediately after the last action in S'_{j-1} .

We now construct S'_j . If s_j is a placing action that places a searcher on an unoccupied vertex, S'_j will take the same action. If s_j is a placing action that places a searcher on an occupied vertex, S'_j will be empty. If s_j is a removing action that removes the only searcher from a vertex, S'_j will take the same action. If s_j is a removing action that removes a searcher from a vertex occupied by at least two searchers, S'_j will be empty. If s_j is a sliding action that slides a searcher from vertex u to vertex v along edge (u, v) to clear the contaminated edge (u, v) , we have two cases.

Case 1. All edges with head u are cleared in D immediately before s_j . By the hypothesis, the vertex $u \in V(D^*)$ is also occupied and all edges with head u in D^* are also cleared immediately after the last action in S'_{j-1} . If v is not occupied, then we can construct S'_j as follows: edge-search-clearing (u, v') , edge-search-clearing (v', v) , removing the searcher from v , placing the searcher on u , edge-search-clearing (u, v'') and edge-search-clearing (v'', v) . If v is occupied,

then we can construct S'_j as follows: edge-search-clearing (u, v') , node-search-clearing (v', v) , removing the searcher from v' , placing the searcher on u , edge-search-clearing (u, v'') , node-search-clearing (v'', v) , and removing the searcher from v'' .

Case 2. At least one edge with head u is contaminated in D immediately before s_j . We know that there is at least one searcher on u while performing s_j , which implies that u is occupied by at least two searchers immediately before s_j . By the hypothesis, the vertex $u \in V(D^*)$ is also occupied and we have at least one free searcher immediately after the last action in S'_{j-1} . If v is not occupied, then we can construct S'_j as follows: placing a searcher on v' , node-search-clearing (u, v') , edge-search-clearing (v', v) , removing the searcher from v , placing the searcher on v'' , node-search-clearing (u, v'') and edge-search-clearing (v'', v) . If v is occupied, then we can construct S'_j as follows: placing a searcher on v' , node-search-clearing (u, v') , node-search-clearing (v', v) , removing the searcher from v' , placing the searcher on v'' , node-search-clearing (u, v'') , node-search-clearing (v'', v) , and removing the searcher from v'' .

If s_j is a sliding action that slides a searcher from vertex u to vertex v along edge (u, v) but does not clear the contaminated edge (u, v) , we know that immediately before s_j , u is occupied by only one searcher and at least one edge with head u is contaminated. By the hypothesis, the vertex $u \in V(D^*)$ is also occupied immediately after the last action in S'_{j-1} . If v is occupied, then S'_j consists of only one action: removing the searcher from u . If v is not occupied, then S'_j consists of two actions: removing the searcher from u and placing it on v .

It is easy to see that $(S'_1, S'_2, \dots, S'_n)$ can clear D^* using at most k searchers.

(ii) \Rightarrow (iii). Since D^* has no multiple edges, by Lemma 4, there is a progressive campaign (X_0, X_1, \dots, X_n) in D^* of width at most $k - 1$. We can modify this campaign to satisfy the requirement of (iii). By Definition 2, we know that m_1, m_2, m_3 and m_4 have different values. We have four cases regarding the smallest value.

Case 1. $m_1 = \min\{m_1, m_2, m_3, m_4\}$. We already have $m_1 < m_2$. If $m_3 > m_4$, then, for each $m_1 + 1 \leq i \leq m_3$, replace X_i by $X'_i = X_{i-1} \cup \{f^3_{(u,v)}\}$. Since $X'_{m_3} = X_{m_3-1} \cup \{f^3_{(u,v)}\} = X_{m_3}$ and $v'' \notin \delta(X'_i)$, $m_1 + 1 \leq i \leq m_3$, it is easy to see that the updated campaign is still a progressive campaign in D^* of width at most $k - 1$. Let the updated campaign still be denoted by (X_0, X_1, \dots, X_n) and m_i ($1 \leq i \leq 4$) is the subscript of X_{m_i} that is the first set containing $f^i_{(u,v)}$ in the updated campaign. Thus, we have $m_1 < m_2$ and $m_3 = m_1 + 1 < m_4$ in the updated campaign.

Case 2. $m_2 = \min\{m_1, m_2, m_3, m_4\}$. For each $m_2 \leq j \leq m_1$, replace X_j by $X'_j = X_{j-1} \cup \{f^1_{(u,v)}\}$. After this modification, the first set containing $f^1_{(u,v)}$ is X'_{m_2} and the first set containing $f^2_{(u,v)}$ is X'_{m_2+1} . Since $f^2_{(u,v)} \in X_i$ and $f^1_{(u,v)} \notin X_i$ for $m_2 \leq i \leq m_1 - 1$, we know that $v' \in \delta(X_i)$, $m_2 \leq i \leq m_1 - 1$. Thus, for $m_2 \leq i \leq m_1 - 1$, we have $\delta(X'_i) \subseteq (\delta(X_i) - \{v'\}) \cup \{u\}$ and $|\delta(X'_i)| \leq |\delta(X_i)|$. We also know that $X'_{m_1} = X_{m_1-1} \cup \{f^1_{(u,v)}\} = X_{m_1}$. It follows that the updated campaign is still

a progressive campaign in D^* of width at most $k - 1$. Let the updated campaign still be denoted by (X_0, X_1, \dots, X_n) and m_i ($1 \leq i \leq 4$) is the subscript of X_{m_i} that is the first set containing $f_{(u,v)}^i$ in the updated campaign. Thus, we have $m_1 < m_2$. Then we can use the method described in Case 1 to achieve $m_3 < m_4$ by modifying the campaign if necessary.

Case 3. $m_3 = \min\{m_1, m_2, m_3, m_4\}$. We already have $m_3 < m_4$ and we can use the method described in Case 1 to modify the campaign such that $m_1 < m_2$ in the updated campaign.

Case 4. $m_4 = \min\{m_1, m_2, m_3, m_4\}$. We can use the method described in Case 2 to modify the campaign such that $m_3 < m_4$ and $m_1 < m_2$ in the updated campaign.

For each $(u, v) \in E(D)$, we can repeat the above procedure for the corresponding four edges $f_{(u,v)}^i \in E(D^*)$, $1 \leq i \leq 4$. Finally, we can obtain a campaign as required.

(iii) \Rightarrow (iv). Let (X_0, X_1, \dots, X_n) be the progressive campaign described in (iii). The monotonic mixed directed search strategy constructed in Lemma 3 can use at most k searchers to clear $f_{(u,v)}^1$ before $f_{(u,v)}^2$ and to clear $f_{(u,v)}^3$ before $f_{(u,v)}^4$ for each $(u, v) \in E(D)$.

(iv) \Rightarrow (v). Let $S = (s_1, s_2, \dots, s_n)$ be a monotonic mixed directed search strategy of D^* satisfying the condition of (iv). We will construct a monotonic directed search strategy S' of D using at most k searchers. For each edge $(u, v) \in E(D)$, without loss of generality, suppose that S clears $f_{(u,v)}^1$ before $f_{(u,v)}^3$. For each i from 1 to n , consider s_i . If s_i is a placing or removing action on a vertex that is also in $V(D)$, S' will take the same action. If s_i is a placing or removing action on a vertex in $V(D^*) - V(D)$, S' will do nothing. If s_i is an edge-search-clearing action that clears edge $f_{(u,v)}^1$, then in S' , we can clear $(u, v) \in E(D)$ in the same way as s_i does. If s_i is a node-search-clearing action that clears edge $f_{(u,v)}^1$ by the searcher α on u and the searcher β on v' , then in S' , we know that α is also on u and β is free. Thus, we can place β on u and then clear $(u, v) \in E(D)$ by sliding β from u to v along (u, v) . If s_i clears edge $f_{(u,v)}^2$, $f_{(u,v)}^3$ or $f_{(u,v)}^4$, we do nothing in S' . It is easy to see that S' can clear D using at most k searchers.

(v) \Rightarrow (i). It is trivial.

4 NP-Completeness Results

Kirousis and Papadimitriou [7] proved that the node search problem is NP-complete. In this section, we will establish a relationship between the mixed directed searching and node searching. Using this relation, we can prove the mixed directed search problem is NP-complete. We can then prove the directed search problem is NP-complete from Theorem 2.

For an undirected graph G , the minimum number of searchers needed to clear G in the node search model is the *node search number* of G , denoted by $ns(G)$.

Theorem 3. *Let G be an undirected graph. If \bar{G} is a digraph obtained from G by replacing each edge $uv \in E(G)$ with two directed edges (u, v) and (v, u) , then $\text{mds}(\bar{G}) = \text{ns}(G)$.*

Proof. In the mixed directed search model, there are four types of actions, placing, removing, node-search-clearing, and edge-search-clearing, and in the node search model, there are only two types of actions, placing and removing. Note that there is no “clearing” action in the node search model corresponding to the node-search-clearing or edge-search-clearing. A contaminated edge is cleared in the node search model if both end vertices are occupied, while a contaminated edge is cleared in the mixed directed search model only by a node-search-clearing or edge-search-clearing action.

We first show that $\text{mds}(\bar{G}) \leq \text{ns}(G)$. Let S_n be a monotonic node search strategy that clears G using k searchers. Notice that S_n is a sequence of placing and removing actions. We will construct a mixed directed search strategy S_m by inserting some node-search-clearing actions into S_n as follows. Initially, we set $S_m = S_n$. For each placing action s in S_n , let A_s be the set of cleared edges just after s and B_s be the set of cleared edges just before s . If $A_s - B_s \neq \emptyset$, then for each edge $uv \in A_s - B_s$, we insert two node-search-clearing actions into the current S_m such that they clear both (u, v) and (v, u) . The order of these clearing actions is arbitrary. Finally, we have a mixed directed search strategy S_m for \bar{G} . It is easy to see that S_m can clear \bar{G} using k searchers. Therefore, $\text{mds}(\bar{G}) \leq \text{ns}(G)$.

We now show that $\text{ns}(G) \leq \text{mds}(\bar{G})$. Let S_m be a monotonic mixed directed search strategy that clears \bar{G} using k searchers. We first prove that there is no edge-search-clearing action in S_m . Suppose that s' is an edge-search-clearing action in S_m , which clears edge (u, v) by sliding from u to v . From Definition 3, all in-edges of u are cleared. Since (v, u) is cleared but (u, v) is contaminated just before s' , the vertex v must contain a searcher to protect (v, u) from recontamination. From Definition 3, (u, v) must be cleared by a node-search-clearing action because both u and v are occupied just before s' . This is a contradiction. Thus, S_m consists of only three types of actions: placing, removing, and node-search-clearing. Let S_n be a sequence of actions obtained from S_m by replacing each node-search-clearing action with an empty action. We next prove that S_n is a node search strategy that clears G using k searchers.

When an edge (u, v) is cleared by a node-search-clearing action in S_m , the corresponding edge uv in G is also cleared just before the corresponding empty action in S_n because both u and v are occupied. Note that S_n may not be monotonic. For any edge $uv \in E(G)$, when both (u, v) and (v, u) are cleared just after a node-search-clearing action in S_m , uv is also cleared just before the corresponding empty action in S_n because both u and v are occupied. We now show that this uv will keep cleared to the end of the search process. Notice that only a removing action may cause recontamination in S_n . For the sake of contradiction, suppose that there is a removing action in S_n such that just after this action an edge, which is incident on the vertex from which a search is just removed, becomes recontaminated, but the corresponding two edges in

\bar{G} are still cleared just after the corresponding removing action in S_m . Let r be the first such a removing action which removes a search from vertex u , and let edge $uv \in E(G)$ becomes recontaminated just after r and both (u, v) and (v, u) in \bar{G} are cleared just after the corresponding removing action r' in S_m . Since uv becomes recontaminated just after r , there must exist a contaminated edge incident on u just before r . Let wu be such a contaminated edge just before r . If (w, u) is also contaminated just before r' , then (u, v) becomes recontaminated just after r' . This contradicts the monotonicity of the strategy S_m . If both (w, u) and (u, w) are cleared just before r' , then wu is also cleared because uv is the first edge which is recontaminated but (u, v) and (v, u) are cleared. This is a contradiction. Thus, just before r' , (w, u) is cleared and (u, w) is contaminated, and w must contain a searcher to protect (w, u) from recontamination. Hence, both w and u are occupied just before r' . It follows that wu is cleared just before r . This is a contradiction. Therefore, for each edge $uv \in E(G)$, if both (u, v) and (v, u) in \bar{G} are cleared, then uv is also cleared. It is easy to see that S_n can clear G using k searchers. Thus, $\text{ns}(G) \leq \text{mds}(\bar{G})$.

Given a digraph D , the problem of determining $\text{ds}(D)$ or $\text{mds}(D)$ is the optimization problem of finding the smallest k such that D can be cleared using k searchers. The corresponding decision problem for the mixed directed search problem is as follows.

Problem: Mixed Directed Searching

Instance: Digraph D , positive integer k .

Question: Can we use k searchers to clear D under the mixed directed search model?

The decision problem for the directed search problem is to determine whether we can clear D using k searchers.

Problem: Directed Searching

Instance: Digraph D , positive integer k .

Question: Can we use k searchers to clear D under the directed search model?

From Theorem 3, we have the following result.

Theorem 4. *The Mixed Directed Searching problem is NP-complete.*

Proof. We first show that the Mixed Directed Searching problem belongs to NP. Suppose we are given a digraph D and a positive integer k . From Theorem 1, a nondeterministic algorithm needs only to guess a monotonic mixed directed search strategy such that the number of actions in this strategy is $O(V(D) + E(D))$. It is easy to see that checking whether this strategy can clear D using k searchers can be accomplished in deterministic polynomial time. Thus, the Mixed Directed Searching problem is in NP. By Theorem 3, we know that the Mixed Directed Searching problem is NP-hard because the node search problem is NP-complete [7]. Therefore, the Mixed Directed Searching problem is NP-complete.

From Theorems 2 and 4, we can prove that the directed search problem is NP-hard. From Theorem 2, we can prove that the directed search problem belongs to NP. Therefore, we have the major result of this section.

Theorem 5. *The Directed Searching problem is NP-complete.*

5 Conclusion

In this paper, we investigated two new digraph searching models, directed searching and mixed directed searching, in which both searchers and intruders must move in the edge directions. Using the method proposed by Bienstock and Seymour [4], we first proved the monotonicity of the mixed directed search model. We then proved the monotonicity of the directed searching. We also give a relationship between the mixed directed searching and the node searching. From this relation and the monotonicity results, we showed that the mixed directed and directed search problems are NP-complete.

References

1. B. Alspach, D. Dyer, D. Hanson and B. Yang, Some basic results in arc searching, *Technical report CS-2006-10*, University of Regina, 2006.
2. J. Barat, Directed path-width and monotonicity in digraph searching, *Graphs and Combinatorics*, 22 (2006) 161–172.
3. D. Berwanger, A. Dawar, P. Hunter and S. Kreutzer, DAG-width and parity games, *Proceedings of STACS, Lecture Notes in Computer Science*, Vol.3884, 524–436, 2006.
4. D. Bienstock and P. Seymour, Monotonicity in graph searching, *Journal of Algorithms*, 12 (1991) 239–245.
5. F. Fomin and D. Thilikos, On the monotonicity of games generated by symmetric submodular functions, *Discrete Applied Mathematics*, 131 (2003) 323–335.
6. T. Johnson, N. Robertson, P. Seymour and R. Thomas, Directed tree-width, *Journal of Combinatorial Theory Series B*, 82 (2001) 138–154.
7. L. Kirousis and C. Papadimitriou, Searching and pebbling, *Theoret. Comput. Sci.*, 47 (1996) 205–218.
8. A. LaPaugh, Recontamination does not help to search a graph. *Journal of ACM*, 40 (1993) 224–245.
9. N. Megiddo, S. Hakimi, M. Garey, D. Johnson and C. Papadimitriou, The complexity of searching a graph, *Journal of ACM*, 35 (1998) 18–44.
10. T. Parsons, Pursuit-evasion in a graph. *Theory and Applications of Graphs*, Lecture Notes in Mathematics, Springer-Verlag, 426–441, 1976.
11. B. Reed, Introducing Directed Tree Width, 6th Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1999), 8 pp. (electronic), *Electron. Notes Discrete Math.*, 3, Elsevier, Amsterdam, 1999.
12. M. Safari, D-Width: A More Natural Measure for Directed Tree Width, *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 3618, pp. 745–756, 2005.
13. P. Seymour and R. Thomas, Graph searching and a min-max theorem for tree-width, *Journal of Combinatorial Theory Series B*, 58 (1993) 22–33.
14. B. Yang and Y. Cao, Monotonicity of strong searching on digraphs, submitted.
15. B. Yang and Y. Cao, On the monotonicity of weak searching on digraphs, submitted.
16. B. Yang and Y. Cao, Digraph searching, directed vertex separation and directed pathwidth, submitted.