

# Strong-mixed Searching and Pathwidth

Boting Yang<sup>1</sup>

Department of Computer Science  
University of Regina

**Abstract.** In this paper, we propose a new search model, called strong-mixed search, which is a generalization of the mixed search. We show that the strong-mixed search number of a graph equals the pathwidth of the graph. We also describe relationships between the strong-mixed search number and other search numbers.

**Keywords:** node search, edge search, mixed search, cops-and-robber game, pursuit-evasion problem.

## 1 Introduction

Let  $G$  be a connected undirected finite graph. Parsons [10] introduced and studied the pursuit-evasion problem on graphs. Megiddo et al. [9] discretized this problem to an edge search model. They proved that determining whether the edge search number of  $G$  ( $es(G)$ ) is bounded by a given integer is NP-complete. Kirousis and Papadimitriou [7] introduced the node search model and proved that  $ns(G) - 1 \leq es(G) \leq ns(G) + 1$ , where  $ns(G)$  denotes the node search number of  $G$ . They also proved that  $ns(G)$  equals the vertex separation number of  $G$  plus one. Kinnersley [6] showed that the vertex separation number of  $G$  equals the pathwidth of  $G$  ( $pw(G)$ ). This implies that  $ns(G) = pw(G) + 1$ . The mixed search model is a combination of the edge search and node search [3, 12]. Takahashi et al. [12] showed that mixed search number ( $ms(G)$ ) equals the proper-path-width of  $G$  ( $ppw(G)$ ). In summary, we have the following results due to [3, 6, 7, 12].

**1.1 Lemma.** [3, 6, 7, 12] *If  $G$  is a connected graph, then*

- (i)  $ns(G) - 1 \leq es(G) \leq ns(G) + 1$ ;
- (ii)  $ms(G) \leq es(G) \leq ms(G) + 1$ ;
- (iii)  $ms(G) \leq ns(G) \leq ms(G) + 1$ ;
- (iv)  $ns(G) = pw(G) + 1$ ; and
- (v)  $ms(G) = ppw(G)$ .

---

<sup>1</sup>Email: boting@cs.uregina.ca. Research was supported in part by NSERC and MITACS.

Graph search problems serve as models for important applied problems (see [1, 2, 4, 5]). In this paper, we propose and study a new search model, called *strong-mixed search*, which is a generalization of the mixed search. We proved that recontamination does not help in the strong-mixed search. We show that the strong-mixed search number of a graph equals the pathwidth of the graph. We also relate the strong-mixed search with other search models.

This paper is organized as follows. In Section 2, we give definitions and notations. In Section 3, we present properties of the strong-mixed search. In Section 4, we establish the relationship between the strong-mixed search number and the pathwidth by relating the strong-mixed search with the node search. In Section 5, we describe relationships between the strong-mixed search and other searches.

## 2 Preliminaries

Throughout this paper, let  $G$  be a connected undirected finite graph embedded in  $\mathbb{E}^3$  such that no pair of edges intersect at a point that is not a common end vertex. A point in  $G$  is either a vertex or an interior point of an edge in  $G$ .

We first define the strong-mixed search model. Initially,  $G$  contains one intruder who is located at a point in  $G$ , and  $G$  does not contain any searchers. Each searcher has no information of the whereabouts of the intruder, but the intruder has complete knowledge of the location of all searchers. The intruder always chooses the best strategy so that he evades capture. Suppose we start at time  $t_0$  and the intruder is captured at time  $t_N$ , and the search time is divided into  $N$  intervals  $(t_0, t_1]$ ,  $(t_1, t_2]$ ,  $\dots$ ,  $(t_{N-1}, t_N]$  such that in each interval  $(t_i, t_{i+1}]$ , exactly one searcher performs one of the following three actions:

- A1.** One searcher is placed on a vertex  $v$ , denoted as *place*( $v$ ).
- A2.** One searcher is removed from a vertex  $v$ , denoted as *remove*( $v$ ).
- A3.** One searcher slides along an edge  $uv$  from  $u$  to  $v$ , denoted as *slide*( $u, v$ ).

The above moment sequence  $(t_0, t_1, \dots, t_N)$  corresponds to the *search steps*. The intruder can move from a point  $x$  to a point  $y$  in  $G$  at any time in the interval  $(t_0, t_N)$  if there exists a path between  $x$  and  $y$  which contains no searcher. As usual, the intruder always takes the best strategy so that

he avoids being captured. Let  $N(v) = \{u : u \text{ is adjacent to } v\}$  and  $N[v] = \{v\} \cup N(v)$ . A *capture* takes place when one of the following events occurs:

- C1.** A searcher and the intruder occupy the same point on  $G$ .
- C2.** The intruder is on an edge whose end vertices are both occupied by searchers.
- C3.** The intruder is in the subgraph induced by  $N[v]$  and all vertices in  $N(v)$  are occupied by searchers.

The goal of searchers is to capture the intruder, and the goal of the intruder is to avoid being captured. We refer to this search model as the *strong-mixed searching*. The *strong-mixed search number* of  $G$ , denoted by  $ss(G)$ , is the smallest positive integer  $k$  such that  $k$  searchers can capture the intruder using the strong-mixed search model. A *strong-mixed search strategy* of  $G$  is a finite sequence of searcher actions that results in the intruder being captured. Note that the strong-mixed search is different from the edge search, node search or mixed search only in the definitions of actions and captures. The edge search uses actions A1, A2 and A3 but only uses capture C1; the node search uses actions A1 and A2 and captures C1 and C2; and the mixed search uses actions A1, A2 and A3 and captures C1 and C2. Similar to the strong-mixed search, we can define the edge search number of  $G$  ( $es(G)$ ), node search number of  $G$  ( $ns(G)$ ) and mixed search number of  $G$  ( $ms(G)$ ), and similarly define their search strategies. We say that a search strategy is *optimal* if the intruder can be captured using the minimum number of searchers.

Before the intruder is captured, there always exists a region of  $G$  in which the intruder hides. This region is called the *intruder's territory*. The vertices and edges in the intruder's territory are called *contaminated*, or *dirty*. For a strong-mixed search, all vertices and edges of  $G$  are contaminated at  $t_0$ . A vertex  $v$  can be *cleared* in two ways: (1)  $v$  is occupied by a searcher; or (2) every vertex in  $N(v)$  is occupied by a searcher. We say that  $v$  is cleared by *sight* in case (2). An edge  $uv$  can be *cleared* in three ways: (1) both  $u$  and  $v$  are occupied by searchers; (2) all vertices of  $N(u)$  or  $N(v)$  are occupied by searchers; or (3) if vertex  $u$  is occupied by a searcher and every edge incident with  $u$ , other than  $uv$ , is already cleared, then the searcher slides along  $uv$  from  $u$  to  $v$ . For cases (1) and (2), we say that  $uv$  is cleared by *sight*; for case (3) we say that  $uv$  is cleared by *sliding*. If the path connecting a cleared edge  $uv$  and a dirty edge contains no searcher at some moment, then  $uv$  becomes a portion of the intruder's territory and must be cleared again. In this case we say  $uv$  becomes *recontaminated*.

At moment  $t_i$ , if a vertex  $v$  has both cleared and contaminated edges incident on it, then  $v$  must have a searcher, say  $\gamma$ , to protect cleared edges incident on it. If there are two or more dirty edges incident on  $v$ , then  $\gamma$  must stay on  $v$  in  $(t_i, t_{i+1}]$  if we do not allow the cleared edges incident on  $v$  to become recontaminated. The searcher  $\gamma$  is called a *guard* at  $t_i$ . If there is exactly one dirty edge incident on  $v$ , then  $\gamma$  may slide along this dirty edge in  $(t_i, t_{i+1}]$  without any recontamination, but we cannot remove  $\gamma$  from  $v$  in  $(t_i, t_{i+1}]$  if we do not allow the cleared edges incident on  $v$  to become recontaminated. The searcher  $\gamma$  is said to be *semi-free* at  $t_i$ . If a searcher is neither a guard nor a semi-free searcher at  $t_i$ , then this searcher is said to be *free* at  $t_i$ .

Let  $E_i$  be the set of cleared edges at  $t_i$ . A search strategy for which  $E_i \subseteq E_{i+1}$  for all  $t_i$  is said to be *monotonic* [3, 8]. The fewest number of searchers required to capture the intruder in  $G$  under the monotonic strong-mixed search model is called the *monotonic strong-mixed search number*, denoted  $\text{mss}(G)$ . Similarly, we can define the monotonic node search and the monotonic node search number ( $\text{mns}(G)$ ).

Robertson and Seymour [11] introduced the concept of the pathwidth of a graph. For graph  $G(V, E)$ , a sequence  $(W_1, W_2, \dots, W_r)$  of subsets of  $V$  is a *path decomposition* of  $G$  if each of the following conditions are satisfied:

- (i)  $\bigcup_{i=1}^r W_i = V$ ;
- (ii) for each edge  $uv \in E$ , there exists a  $W_i$  such that  $u, v \in W_i$ ; and
- (iii) for all integers  $i, j, k$  with  $1 \leq i \leq j \leq k \leq r$ ,  $W_i \cap W_k \subseteq W_j$ .

The *width* of a path decomposition is defined as  $\max\{|W_i| - 1 : 1 \leq i \leq r\}$ . The *pathwidth* of  $G$ , denoted by  $\text{pw}(G)$ , is the minimum width over all possible path decompositions of  $G$ .

### 3 Properties of strong-mixed searching

We first consider the subgraph cleared at each step in the strong-mixed search.

**3.1 Lemma.** *Let  $X = (X_0, X_1, \dots, X_{N-1})$  be a strong-mixed search strategy for a connected graph  $G$  such that  $X_i$  is the action in interval  $(t_i, t_{i+1}]$ . Let  $E_i, 0 \leq i \leq N$ , be the set of cleared edges at  $t_i$  and  $G_{i+1}$  be the graph formed by the edges of  $E_{i+1} \setminus E_i$ . If  $E_{i+1} \setminus E_i \neq \emptyset$ , then there exists a vertex  $v$  in  $G_{i+1}$  such that the graph  $G_{i+1} - N[v]$  consists of only isolated vertices.*

PROOF. Note that the strong-mixed search has three kinds of actions. For an action  $X_i$  in  $(t_i, t_{i+1}]$ , if  $X_i$  is a removing action, then  $E_{i+1} \setminus E_i = \emptyset$ . Thus, we only need to consider placing and sliding actions. In either one of

these two actions, let  $v$  be the vertex that receives a searcher in  $(t_i, t_{i+1}]$ . If  $v$  is cleared at  $t_i$ , then  $E_{i+1} \setminus E_i = \emptyset$ . Thus, we can assume that  $v$  is dirty at  $t_i$ . Hence, at least one adjacent vertex of  $v$  is not occupied by a searcher at  $t_i$ . Let  $u$  be a vertex adjacent to  $v$ . If a searcher is on  $u$  at  $t_{i+1}$ , then  $uv \in E_{i+1} \setminus E_i$ . We now consider the case that no searcher is on  $u$  at  $t_{i+1}$ . If each vertex in  $N(u)$  is occupied by a searcher at  $t_{i+1}$ , then all edges incident on  $u$  (which form a star) belong to  $E_{i+1} \setminus E_i$ ; otherwise, there is a vertex  $w \in N(u) - \{v\}$  that is not occupied by any searcher at  $t_{i+1}$ . Then every edge incident on  $u$  is not in  $E_{i+1} \setminus E_i$ . Let  $u_1, u_2, \dots, u_m \in N(v)$  be such that each  $u_i$  is occupied by a searcher at  $t_i$  or  $u_i$  is not occupied by a searcher, but each vertex in  $N(u_i)$  is occupied by a searcher at  $t_{i+1}$ . It is easy to see that  $G_{i+1} - (\{v\} \cup \{u_i : 1 \leq i \leq m\})$  consists of only isolated vertices. ■

The graph  $G_{i+1}$  in Lemma 3.1 is called a *1-action-clearing graph* of a strong-mixed search and vertex  $v$  is called the *search center* of  $G_{i+1}$  (see Figure 1). Because  $G_{i+1}$  is the graph formed by the edges of  $E_{i+1} \setminus E_i$ , the vertex set of  $G_{i+1}$  consists of all end vertices of the edges in  $E_{i+1} \setminus E_i$ . The graph  $G_{i+1}$  can be decomposed into a set of stars (a star of  $k$  vertices is a tree with  $k - 1$  leaves and one center vertex). Each star has a center in  $N[v]$ , where  $v$  is the search center of  $G_{i+1}$ . It is easy to see that the radius of  $G_{i+1}$  is at most 2. From Lemma 3.1, we have the following result.

**3.2 Theorem.** *Let  $u$  be a vertex and  $vw$  be an edge in a given connected graph  $G$  such that  $u \notin \{v, w\}$ . For a strong-mixed search strategy of  $G$ , if  $u$  is not occupied by a searcher and  $vw$  is contaminated at  $t_i$ , and  $u$  is occupied by a searcher and  $vw$  becomes cleared at  $t_{i+1}$ , then exactly one end vertex of  $vw$  is occupied by a searcher at  $t_i$  and the other end vertex is adjacent to  $u$ .*

PROOF. From Lemma 3.1, we know that  $u$  is the search center of the graph  $G_{i+1}$  formed by the edges of  $E_{i+1} \setminus E_i$  and  $vw$  is an edge in  $G_{i+1}$ . At  $t_i$ , if both  $v$  or  $w$  are occupied by searchers, then  $vw$  is cleared. This contradicts the condition that  $vw \in E_{i+1} \setminus E_i$ . Suppose that no searcher is on  $v$  or  $w$  at  $t_i$ . If there is an intruder on the middle of edge  $vw$ , then the intruder cannot be captured by sight in  $(t_i, t_{i+1}]$ . Thus,  $vw$  cannot be cleared at  $t_{i+1}$ . This is a contradiction. Hence, at least one of the two end vertices of  $vw$  must be occupied by a searcher in  $(t_i, t_{i+1}]$ . Without loss of generality, we can assume that  $w$  is occupied by a searcher in  $(t_i, t_{i+1}]$ . We now show that  $v$  must be adjacent to  $u$ . If each adjacent vertex of  $v$  is occupied by a searcher at  $t_i$ , then  $v$  and all its incident edges are cleared. This is a contradiction. Let  $x \in N(v)$  be such that there is no searcher on

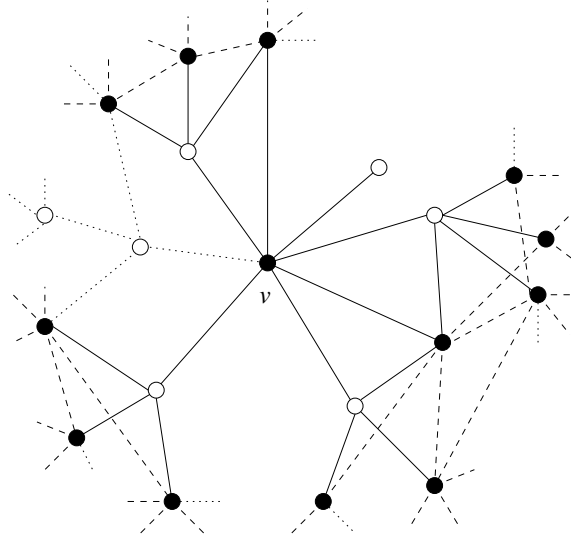


Figure 1: The solid-circle vertices are occupied by searchers. A searcher is placed on  $v$  at the current step. The solid edges form the 1-action-clearing graph with the search center  $v$ . The dashed edges are cleared in the previous steps and the dotted edges are contaminated edges.

$x$  at  $t_i$ . Then  $vx$  is contaminated at  $t_i$ . At  $t_{i+1}$ , if  $x$  has not been occupied by a searcher, then  $vx$  cannot be cleared by sight. Since  $vx$  is contaminated at  $t_{i+1}$ ,  $vw$  is still contaminated at  $t_{i+1}$ . This is a contradiction. Therefore,  $x$  must have a searcher at  $t_{i+1}$ . That means  $x$  is identical with  $u$ . ■

## 4 Strong-mixed search and pathwidth

In this section, we will prove that  $ss(G) = pw(G)$  by characterizing the relationship between the strong-mixed search and node search. We first show the following relation between  $ss(G)$  and  $ns(G)$ .

**4.1 Lemma.** *If  $G$  is a connected graph, then  $ns(G) \leq ss(G) + 1$ .*

PROOF. Let  $ss(G) = k$  and  $X = (X_0, X_1, \dots, X_{N-1})$  be an optimal strong-mixed search strategy such that  $X_i$  is the action in  $(t_i, t_{i+1}]$ . During  $(t_i, t_{i+1}]$ ,  $X_i$  is one of the three actions: *place*, *remove* and *slide*. There is no searcher on  $G$  at  $t_0$  and  $X_0$  is a placing action. Let  $E_i(X)$ ,  $0 \leq i \leq N$ , be the set of cleared edges at moment  $t_i$ .

By simulating the strong-mixed search strategy  $X$ , we will construct a node search strategy  $Y$  that uses  $k + 1$  searchers. For each action  $X_i$  in  $(t_i, t_{i+1}]$ ,  $0 \leq i \leq N - 1$ , we use a sequence of actions, denoted as  $y(X_i)$ , to simulate the action  $X_i$  within the same interval  $(t_i, t_{i+1}]$ . Thus,  $Y$  is the concatenation of  $y(X_i)$  and can be expressed as  $(y(X_0), y(X_1), \dots, y(X_{N-1}))$ . Let  $E_i(Y)$  be the set of all cleared edges by strategy  $Y$  at the moment  $t_i$ . Note that  $Y$  has only two actions, *place* and *remove*, and it may have several actions in  $(t_i, t_{i+1}]$ ,  $0 \leq i \leq N - 1$ . We now construct  $Y$  from  $X$  inductively such that  $E_i(X) = E_i(Y)$  for each  $i$  satisfying  $0 \leq i \leq N$ . It is easy to see that  $E_0(X) = E_0(Y) = \emptyset$ .

Initially, if the action  $X_0$  is  $place_X(u)$ , then the first action in  $y(X_0)$  is  $place_Y(u)$ . If  $E_1(X) = \emptyset$ , then  $y(X_0) = (place_Y(u))$  and  $E_1(Y) = \emptyset$ . If  $E_1(X) \neq \emptyset$ , then, for each edge in  $E_1(X)$ , one end vertex must be  $u$  and the other must be a leaf. Let  $E_1(X) = \{uv_1, uv_2, \dots, uv_m\}$ . Since the node search has  $k + 1$  searchers, we can construct  $y(X_0) = (place_Y(u), place_Y(v_1), remove_Y(v_1), \dots, place_Y(v_m), remove_Y(v_m))$ . Thus,  $E_1(Y) = E_1(X)$  and only vertex  $u$  is occupied by one searcher for both strategies at  $t_1$ .

Suppose that  $E_{j-1}(Y) = E_{j-1}(X)$  and both strategies have the same set of vertices that contain searchers at  $t_{j-1}$ . We now consider  $E_j(X)$  and  $E_j(Y)$ . There are three cases regarding the action of  $X_j$ .

CASE 1.  $X_j = place_X(a)$ . If  $E_j(X) \setminus E_{j-1}(X) = \emptyset$ , then no edge is cleared by  $X_j$ , and no recontamination happens. Thus, we set  $y(X_j) = (place_Y(a))$ . It is easy to see that  $E_j(Y) = E_{j-1}(Y) = E_{j-1}(X) = E_j(X)$ . If  $E_j(X) \setminus E_{j-1}(X) \neq \emptyset$ , it follows from Lemma 3.1 that the graph  $G_j$  formed by the edges of  $E_j(X) \setminus E_{j-1}(X)$  is a 1-action-clearing graph with search center  $a$ . It is easy to see that each vertex of  $G_j - N[a]$  is occupied by a searcher at  $t_{j-1}$ . Let  $V(G_j) \cap N(a) = \{u_1, u_2, \dots, u_m\}$ . We can construct  $y(X_j) = (place_Y(a), place_Y(u_1), remove_Y(u_1), \dots, place_Y(u_m), remove_Y(u_m))$  which can clear  $G_j$  using  $k + 1$  searchers by  $Y$ . Thus,  $E_j(X) \setminus E_{j-1}(X) = E_j(Y) \setminus E_{j-1}(Y)$ . It follows from the inductive hypothesis that  $E_j(X) = E_j(Y)$  and both strategies have the same set of vertices that are occupied by searchers at  $t_j$ .

CASE 2.  $X_j = remove_X(a)$ . We can simply set  $y(X_j) = (remove_Y(a))$ . Since no edge can be cleared by removing actions in both  $X$  and  $Y$ , we have  $E_j(X) \setminus E_{j-1}(X) = E_j(Y) \setminus E_{j-1}(Y) = \emptyset$ . If each edge incident on  $a$  is cleared at  $t_{j-1}$ , then no recontamination happens by  $remove_X(a)$  and  $remove_Y(a)$ . Thus,  $E_j(X) = E_j(Y)$ . If there exists a dirty edge incident on  $a$  at  $t_{j-1}$ , consider any cleared edge  $xy$  at  $t_{j-1}$ . If there is a path that contains  $a$  and  $xy$  and has only one searcher stationed on  $a$  at  $t_{j-1}$ , then  $xy$  becomes dirty at  $t_j$  for both  $X$  and  $Y$  strategies. Thus,  $E_{j-1}(X) \setminus E_j(X) = E_{j-1}(Y) \setminus E_j(Y)$ .

From the inductive hypothesis, we know that  $E_j(X) = E_j(Y)$  and both strategies have the same set of vertices that are occupied by searchers at  $t_j$ .

CASE 3.  $X_j = \text{slide}_X(a, b)$ . If  $E_j(X) \setminus E_{j-1}(X) = \emptyset$ , then we set  $y(X_j) = (\text{place}_Y(b), \text{remove}_Y(a))$ . It is easy to see that  $E_j(X) \setminus E_{j-1}(X) = E_j(Y) \setminus E_{j-1}(Y)$ . If  $E_j(X) \setminus E_{j-1}(X) \neq \emptyset$ , it follows from Lemma 3.1 that the graph  $G_j$  formed by the edges of  $E_j(X) \setminus E_{j-1}(X)$  is a 1-action-clearing graph with search center  $b$ . Similar to CASE 1, let  $V(G_j) \cap N(b) = \{u_1, u_2, \dots, u_m\}$ . We can construct  $y(X_j) = (\text{place}_Y(b), \text{remove}_Y(a), \text{place}_Y(u_1), \text{remove}_Y(u_1), \dots, \text{place}_Y(u_m), \text{remove}_Y(u_m))$  which can clear  $G_j$  using  $k + 1$  searchers by  $Y$ . Thus,  $E_j(X) \setminus E_{j-1}(X) = E_j(Y) \setminus E_{j-1}(Y)$ . On the other hand, if  $E_{j-1}(X) \setminus E_j(X) = \emptyset$ , then  $E_j(Y) = E_j(X)$ ; if  $E_{j-1}(X) \setminus E_j(X) \neq \emptyset$ , similar to CASE 2, we have that  $E_{j-1}(Y) \setminus E_j(Y) = E_{j-1}(X) \setminus E_j(X)$  since both  $X$  and  $Y$  have the same set of vertices that are occupied by searchers at  $t_{j-1}$  and  $t_j$ . Therefore,  $E_j(X) = E_j(Y)$  and both strategies have the same set of vertices are occupied by searchers at  $t_j$ . ■

Kirousis and Papadimitriou [7] proved the following monotonicity result for the node search.

**4.2 Lemma.** [7] *If  $G$  is a connected graph, then  $\text{ns}(G) = \text{mns}(G)$ , and furthermore, there always exists an optimal node search strategy in which no vertex is visited twice by a searcher, and in which every searcher is removed immediately after all the edges incident on it have been cleared (ties are broken arbitrarily).*

Similar to the strong-mixed search, suppose that the node search starts at time  $t_0$  and the intruder is captured at time  $t_N$ , and the searching time is divided into  $N$  intervals  $(t_0, t_1]$ ,  $(t_1, t_2]$ ,  $\dots$ ,  $(t_{N-1}, t_N]$  such that in each interval  $(t_i, t_{i+1}]$ , exactly one searcher is either placed on a vertex or removed from a vertex.

**4.3 Lemma.** *If  $G$  is a connected graph with at least one edge, then  $\text{mss}(G) + 1 \leq \text{mns}(G)$ .*

PROOF. Let  $\text{mns}(G) = k$  and  $Y = (Y_0, Y_1, \dots, Y_{N-1})$  be an optimal node search strategy described in Lemma 4.2 such that  $Y_i$  is the action in interval  $(t_i, t_{i+1}]$ . Since no vertex is visited twice,  $Y$  is monotonic. Because there is no searcher on  $G$  at  $t_0$ , the action at  $(t_0, t_1]$  must be *place*. Let  $E_i(Y)$ ,  $0 \leq i \leq N$  be the set of cleared edges at  $t_i$ . Thus,  $E_0(Y) = \emptyset$ .

By simulating  $Y$ , we construct a monotonic strong-mixed search strategy  $X$  that uses at most  $k - 1$  searchers. Let  $X = (X_0, X_1, \dots, X_{N-1})$ , where



$X_i, 0 \leq i \leq N-1$ , is an action corresponding to the action  $Y_i$ . Since an action  $slide_X(u, v)$  in  $X$  corresponds to two actions,  $remove_Y(u)$  and  $place_Y(v)$ , in  $Y$ , we introduce an *empty* action in  $X$  such that all searchers on the graph stay still during the interval when the empty action is executed. Notice that each sliding action in  $X$  is followed immediately by an empty action. Thus, the number of sliding actions is equal to that of empty actions in  $X$ . Let  $E_i(X)$  be the set of all cleared edges of strategy  $X$  at  $t_i$ .

We now construct  $X$  such that  $E_i(Y) \subseteq E_i(X)$  for each  $i$  satisfying  $0 \leq i \leq N$ . Since  $G$  has at least one edge, we know that  $k \geq 2$ . Consider an arbitrary interval  $(t_i, t_{i+1}]$ ,  $0 \leq i \leq N-1$ . If  $Y$  uses at most  $k-2$  searchers in  $G$  at moment  $t_i$ , then we set the action  $X_i$  to be the same action as  $Y_i$ . If  $Y$  uses  $k-1$  searchers in  $G$  at  $t_i$  and  $Y_i$  is a removing action, then we can set  $X_i$  to the same action. If  $Y$  uses  $k-1$  searchers in  $G$  at  $t_i$  and  $Y_i$  is a placing action, then  $Y_{i+1}$  must be a removing action. Let  $Y_i = place_Y(u)$  and  $Y_{i+1} = remove_Y(v)$ . For strategy  $Y$ , it follows from Lemma 4.2 that every searcher is removed immediately after all the edges incident on it have been cleared. Thus, there must exist at least one dirty edge incident on  $v$  at  $t_i$  and all the edges incident on  $v$  are cleared at  $t_{i+1}$ . Hence,  $uv$  is the only dirty edge incident on  $v$  at  $t_i$ . Therefore, we can set  $X_i = slide_X(v, u)$  and  $X_{i+1} = empty$ .

We now show that  $E_i(Y) \subseteq E_i(X)$  for each  $i$  by induction. For the basis, we know that  $Y_0 = place_Y(u)$  and  $X_0 = place_X(u)$ . Thus,  $E_1(Y) \subseteq E_1(X)$ . Suppose that the inductive hypothesis holds for  $i$ . We now prove it is true for  $i+1$ . Consider action  $Y_i, 1 \leq i \leq N-1$ , in  $(t_i, t_{i+1}]$ . If  $Y$  uses at most  $k-2$  searchers in  $G$  at  $t_i$  or  $Y$  uses  $k-1$  searchers in  $G$  at  $t_i$  and  $Y_{i+1}$  is a removing action, then  $X_{i+1} = Y_{i+1}$  and  $E_{i+1}(Y) \setminus E_i(Y) \subseteq E_{i+1}(X) \setminus E_i(X)$ . Thus,  $E_{i+1}(Y) \subseteq E_{i+1}(X)$ . If  $Y$  uses  $k-1$  searchers in  $G$  at  $t_i$  and  $Y_i = place_Y(u)$  and  $Y_{i+1} = remove_Y(v)$ , then  $X_i = slide_X(v, u)$  and  $X_{i+1} = empty$ . Since  $uv$  is the only dirty edge incident on  $v$  at  $t_i$ ,  $slide_X(v, u)$  can clear  $uv$  in  $(t_i, t_{i+1}]$ . By the hypothesis, we have  $E_{i+1}(Y) \subseteq E_{i+1}(X)$ .

We have constructed a monotonic strong-mixed search strategy  $X$  that uses at most  $k-1$  searchers. Therefore,  $mss(G) + 1 \leq mns(G)$ . ■

From Lemma 4.3 and Lemma 1.1(iv), we know that  $mss(G) \leq pw(G)$ . We can also prove this inequality by using a path decomposition directly. The basic idea is to establish a relation between an occupied vertex set at each step and the bag in a path decomposition. Note that each vertex is visited exactly once by a searcher because we are considering the monotonic strong-mixed search model. Thus, we can put all occupied vertices at a step into one bag and we may add one more vertex  $v$  into the same bag if all vertices in  $N(v)$  are occupied. In order to show that  $mss(G) \leq pw(G)$ , we

need to assume that  $|(W_i \setminus W_{i+1}) \cup (W_{i+1} \setminus W_i)| = 1$  and  $|W_1| = |W_N| = 1$  in a given path decomposition  $(W_1, W_2, \dots, W_N)$  of width  $k$  ( $= \text{pw}(G)$ ). We can then design a monotonic strong-mixed search strategy to clear all vertices in every bag one by one in the order of  $W_1, W_2, \dots, W_N$  by using  $k$  searchers.

Since  $\text{ss}(G) \leq \text{mss}(G)$ , from Lemmas 4.1, 4.2 and 4.3, we have the following theorem.

**4.4 Theorem.** *If  $G$  is a connected graph with at least one edge, then  $\text{ss}(G) = \text{mss}(G) = \text{ns}(G) - 1$ .*

From Theorem 4.4 and Lemma 1.1 (iv), we have the following main result of this section.

**4.5 Theorem.** *If  $G$  is a connected graph with at least one edge, then  $\text{ss}(G) = \text{pw}(G)$ .*

Since computing  $\text{ns}(G)$  is an NP-complete problem [7], from Theorem 4.4, we have the following result.

**4.6 Corollary.** *Given a graph  $G$  and an integer  $K$ , the problem of determining whether  $\text{ss}(G) \leq K$  is NP-complete.*

## 5 Relations between search models

In Theorem 4.4, we show that  $\text{ss}(G) = \text{mss}(G) = \text{ns}(G) - 1$ . In this section, we relate the strong-mixed search with other search models. From Lemma 1.1 and Theorem 4.4, it is easy to show the following lemma.

**5.1 Lemma.** *If  $G$  is a connected graph, then  $\text{ss}(G) \leq \text{es}(G) \leq \text{ss}(G) + 2$  and  $\text{ss}(G) \leq \text{ms}(G) \leq \text{ss}(G) + 1$ .*

**5.2 Theorem.** *If  $\text{ns}(G) < \text{es}(G)$ , then  $\text{ss}(G) = \text{es}(G) - 2 = \text{ms}(G) - 1 = \text{ns}(G) - 1$ .*

PROOF. Since  $\text{ns}(G) < \text{es}(G)$ , it follows from Lemma 1.1 (i), we know that  $\text{ns}(G) = \text{es}(G) - 1$ . Suppose that  $\text{ms}(G) < \text{ns}(G)$ . From Lemma 1.1 (iii), we have  $\text{ms}(G) = \text{ns}(G) - 1$ . Thus,  $\text{ms}(G) = \text{es}(G) - 2$ . This contradicts Lemma 1.1 (ii). Hence,  $\text{ms}(G) = \text{ns}(G) = \text{es}(G) - 1$ . From Theorem 4.4, we have  $\text{ss}(G) = \text{es}(G) - 2$ . ■

For the complete bipartite graph  $K_{3,3}$ , it is easy to see that  $\text{ns}(K_{3,3}) = 4$  and  $\text{es}(K_{3,3}) = 5$ . It follows from Theorem 5.2 that  $\text{ms}(K_{3,3}) = 4$  and  $\text{ss}(K_{3,3}) = 3$ .

**5.3 Theorem.** *If  $\text{ns}(G) > \text{es}(G)$ , then  $\text{ss}(G) = \text{es}(G) = \text{ms}(G) = \text{ns}(G) - 1$ .*

PROOF. From Lemmas 1.1 and 5.1, we have

$$\text{ns}(G) - 1 = \text{ss}(G) \leq \text{ms}(G) \leq \min\{\text{es}(G), \text{ns}(G)\}.$$

Since  $\text{ns}(G) > \text{es}(G)$ , we know that  $\min\{\text{es}(G), \text{ns}(G)\} = \text{es}(G) = \text{ns}(G) - 1$ . Therefore,  $\text{ss}(G) = \text{ms}(G) = \text{es}(G) = \text{ns}(G) - 1$ . ■

Let  $P$  be a path. It is easy to see that  $\text{ns}(P) = 2$  and  $\text{ss}(P) = \text{ms}(P) = \text{es}(P) = 1$ .

If  $\text{ns}(G) = \text{es}(G)$ , then  $\text{ss}(G)$  may be equal to  $\text{ms}(G)$  or  $\text{ms}(G) + 1$ . For example, let  $K_4$  be a complete graph of order 4,  $K'_4$  be a graph obtained by replacing each edge of  $K_4$  by a path of length 2, and  $K''_4$  be a graph obtained by replacing each edge of  $K_4$  by a path of length 3. It is easy to verify that  $\text{ss}(K_4) = \text{ms}(K_4) < \text{es}(K_4) = \text{ns}(K_4)$ ;  $\text{ss}(K'_4) < \text{ms}(K'_4) = \text{es}(K'_4) = \text{ns}(K'_4)$ ; and  $\text{ss}(K''_4) = \text{ms}(K''_4) = \text{es}(K''_4) < \text{ns}(K''_4)$ .

**5.4 Corollary.** *If  $\text{ms}(G) = \text{es}(G)$ , then  $\text{ss}(G) \geq \text{es}(G) - 1$ .*

PROOF. Since  $\text{ms}(G) \leq \text{ns}(G)$ , we have  $\text{es}(G) \leq \text{ns}(G)$ . If  $\text{es}(G) = \text{ns}(G)$ , then  $\text{ss}(G) = \text{es}(G) - 1$ . If  $\text{es}(G) = \text{ns}(G) - 1$ , then  $\text{ss}(G) = \text{es}(G)$ . Thus,  $\text{ss}(G) \geq \text{es}(G) - 1$ . ■

**5.5 Corollary.** *If  $\text{ms}(G) < \text{ns}(G)$ , then  $\text{ss}(G) = \text{ms}(G)$ .*

PROOF. From Lemma 1.1 (iii), we have  $\text{ms}(G) = \text{ns}(G) - 1$ . It follows from Theorem 4.4 that  $\text{ss}(G) = \text{ms}(G)$ . ■

Given graph  $G$ , we call a path  $u_1u_2 \dots u_t$  *suspended* when the vertices  $u_2, u_3, \dots, u_{t-1}$  all have degree 2 in  $G$  and both  $u_1$  and  $u_t$  have degree 3 or more in  $G$ . We call a path  $u_1u_2 \dots u_t$  *pendant* when the vertices  $u_2, u_3, \dots, u_{t-1}$  all have degree 2 in  $G$ , and either  $u_1$  or  $u_t$  have degree 1 in  $G$  and neither of them has degree 2 in  $G$ .

Theorem 4.4 shows a relationship between  $\text{ss}(G)$  and  $\text{ns}(G)$ . The following four theorems describe the relation between  $\text{ss}(G)$  and  $\text{es}(G)$  from different aspects.

**5.6 Theorem.** *For a connected graph  $G$ , if  $\text{es}(G) = \text{ss}(G)$ , then for any optimal monotonic edge search strategy, there exists a moment  $t_i$  such that this strategy has  $\text{es}(G) - 1$  guards at  $t_i$  and the remaining searcher clears a suspended path of length at least 3 or clears a pendant path of length at least 2 just after  $t_i$ .*

PROOF. Let  $\text{es}(G) = k$  and  $Z = (Z_0, Z_1, \dots, Z_{N-1})$  be an optimal monotonic edge search strategy such that  $Z_i$  is the action in  $(t_i, t_{i+1}]$ . Let  $E_i(Z), 0 \leq i \leq N$ , be the set of cleared edges at  $t_i$ . Thus,  $E_0(Z) = \emptyset$ .

Suppose that there is no moment  $t_i$  such that  $Z$  has  $k - 1$  guards at  $t_i$  and the remaining searcher clears a suspended path of length at least 3 or clears a pendant path of length at least 2 just after  $t_i$ . By simulating  $Z$ , we construct a strong-mixed search strategy  $X$  that uses at most  $k - 1$  searchers. Let  $X = (X_0, X_1, \dots, X_{N-1})$ , where  $X_i, 0 \leq i \leq N - 1$ , is an action in  $(t_i, t_{i+1}]$  which corresponds to the action  $Z_i$ . Let  $E_i(X)$  be the set of all cleared edges of strategy  $X$  at  $t_i$ .

We now show that  $E_i(Z) \subseteq E_i(X)$  for each  $i$  satisfying  $0 \leq i \leq N - 1$ . If  $Z$  uses at most  $k - 2$  guards at  $t_i$ , then we set  $X_i = Z_i$ . If  $Z$  uses  $k - 1$  guards and the free or semi-free searcher clears a suspended path of length at most 2 or clears a pendant edge, then this suspended path of length at most 2 or pendant edge is already cleared at  $t_i$  under the strong-mixed search model. Thus,  $X_i$  can take an empty action. In both cases, it is easy to see that  $E_{i+1}(Z) \subseteq E_{i+1}(X)$ . Therefore, if there is no moment  $t_i$  satisfies the condition of the theorem, then  $\text{ss}(G) \leq \text{es}(G) - 1$ . This is a contradiction. ■

**5.7 Theorem.** *If there exists an optimal strong-mixed search strategy such that for each moment  $t_i$  when this strategy has at least  $\text{ss}(G) - 2$  guards, no vertex or edge must be cleared by sight in  $(t_i, t_{i+1}]$ , then  $\text{es}(G) = \text{ss}(G)$ .*

PROOF. Let  $\text{ss}(G) = k$  and  $X = (X_0, X_1, \dots, X_{N-1})$  be an optimal strong-mixed search strategy such that  $X_i$  is the action in interval  $(t_i, t_{i+1}]$ .

We construct an edge search strategy  $Z$  that uses  $k$  searchers to simulate  $X$ . Let  $Z = (z(X_0), z(X_1), \dots, z(X_{N-1}))$ , where  $z(X_i), 0 \leq i \leq N - 1$ , is a sequence of actions in  $(t_i, t_{i+1}]$  that clear the same set of edges as the action  $X_i$  does. Let  $E_i(X)$  and  $E_i(Z)$  be the set of all cleared edges by strategy  $X$  and  $Z$  at  $t_i$ , respectively.

We now show that  $E_i(X) = E_i(Z)$  for each  $i$  satisfying  $0 \leq i \leq N - 1$ . If  $X$  uses at most  $k - 3$  guards at  $t_i$ , then we set the first action in  $z(X_i)$  as  $X_i$ . Notice that while  $E_{i+1}(X) \setminus E_i(X)$  is cleared by one searcher in  $X$ ,  $E_{i+1}(X) \setminus E_i(X)$  is cleared by at most three searchers in  $Z$ . Thus, we can set the remaining actions in  $z(X_i)$  to clear edges  $E_{i+1}(X) \setminus E_i(X)$  such that  $E_{i+1}(Z) = E_{i+1}(X)$ . If  $X$  uses at least  $k - 2$  guards, since no vertex or edge must be cleared by sight in  $(t_i, t_{i+1}]$ , we can use the sliding actions in  $z(X_i)$  to clear edges such that  $E_{i+1}(Z) = E_{i+1}(X)$ . Thus,  $G$  can be cleared by the edge search strategy  $Z$  using  $k$  searchers. Hence,  $\text{es}(G) \leq k$ . Since  $\text{ss}(G) \leq \text{es}(G)$ , we have  $\text{es}(G) = \text{ss}(G)$ . ■

**5.8 Theorem.** *Let  $G$  be a connected graph. For any optimal strong-mixed search strategy  $X$ , for any moment  $t_i$  when  $X$  has  $\text{ss}(G) - 1$  guards, if no vertex of degree at least 3 is cleared by sight in  $(t_i, t_{i+1}]$ , then  $\text{es}(G) \leq \text{ss}(G) + 1$ .*

PROOF. Let  $\text{ss}(G) = k$  and  $X = (X_0, X_1, \dots, X_{N-1})$ , where  $X_i$  is the action in  $(t_i, t_{i+1}]$ . We now construct an edge search strategy  $Z = (z(X_0), z(X_1), \dots, z(X_{N-1}))$  such that  $G$  can be cleared by  $k + 1$  searchers, where  $z(X_i)$  is a sequence of actions that simulate the action  $X_i$  in  $(t_i, t_{i+1}]$ . Let  $E_i(X)$  be the set of cleared edges by the strategy  $X$  at  $t_i$ , and  $E_i(Z)$  be the set of cleared edges by the strategy  $Z$  at  $t_i$ . Both  $X$  and  $Z$  have three actions, *place*, *slide* and *remove*. Since a searcher in  $Z$  is not as powerful as that in  $X$ ,  $z(X_i)$  may have several actions in  $(t_i, t_{i+1}]$  to clear the same edges as  $X_i$  does. We now construct  $Z$  from  $X$  inductively such that  $E_i(X) = E_i(Z)$  for each  $i$  satisfying  $0 \leq i \leq N$ . It is easy to see that  $E_0(X) = E_0(Z) = \emptyset$ . Suppose  $E_i(X) = E_i(Z)$  for any  $i < N - 1$ . We now consider the interval  $(t_i, t_{i+1}]$ . For any edge cleared by  $X$  in  $(t_i, t_{i+1}]$ , if this edge is cleared by sliding a searcher along an edge, then this edge can be also cleared by  $Z$  in the same way in  $(t_i, t_{i+1}]$ ; if this edge is cleared by sight in  $X$ , from the assumption that no vertex of degree at least 3 is cleared by sight when  $X$  uses  $k - 1$  guards, we know that this edge can be cleared by a free searcher or two free searchers of  $Z$  in  $(t_i, t_{i+1}]$ . Hence,  $E_{i+1}(X) = E_{i+1}(Z)$  and  $G$  can be cleared by  $k + 1$  searchers of  $Z$ . Therefore,  $\text{es}(G) \leq \text{ss}(G) + 1$ . ■

From Theorem 5.8, we have the following corollary.

**5.9 Corollary.** *For a connected graph  $G$ , if  $\text{es}(G) = \text{ss}(G) + 2$ , then for any optimal strong-mixed search strategy, there exists a moment  $t_i$  such that  $G$  has  $\text{ss}(G) - 1$  guards at  $t_i$  and a vertex of degree at least 3 is cleared by sight in  $(t_i, t_{i+1}]$ .*

Theorem 5.8, which describes a relation between  $\text{ss}(G)$  and  $\text{es}(G)$ , depends on search strategies. The following theorem reveals the same relation from the structure of  $G$ .

**5.10 Theorem.** *For a connected graph  $G$ , let  $(W_1, W_2, \dots, W_r)$  be its path decomposition of width  $\text{pw}(G)$ . If  $|W_{i-1} \cap W_{i+1}| < \text{pw}(G)$  for each  $2 \leq i \leq r - 1$ , then  $\text{es}(G) \leq \text{ss}(G) + 1$ .*

PROOF. Let  $\text{pw}(G) = k$ . By Theorem 4.5, we only need to show that  $G$  can be cleared by an edge search strategy using at most  $k + 1$  searchers.

For each vertex set  $W_i$  in the path decomposition  $(W_1, W_2, \dots, W_r)$ , if it is a subset of another vertex set  $W_j$ , we can delete  $W_i$ . By this operation, we get a new vertex set sequence  $(W_{i_1}, W_{i_2}, \dots, W_{i_m})$  such that any element in this sequence is not a subset of another element. It is easy to see that  $(W_{i_1}, W_{i_2}, \dots, W_{i_m})$  is still a path decomposition of width  $\text{pw}(G)$ . For any  $2 \leq j \leq m - 1$ , if  $i_{j+1} = i_{j-1} + 2$ , by the condition of the theorem, we have  $|W_{i_{j-1}} \cap W_{i_{j+1}}| < \text{pw}(G)$ ; if  $i_{j+1} \geq i_{j-1} + 3$ , by the condition (iii) in the definition of the path decomposition, we have  $W_{i_{j-1}} \cap W_{i_{j+1}} \subseteq W_{i_{j-1}+2}$ . Thus,  $|W_{i_{j-1}} \cap W_{i_{j+1}}| \leq |W_{i_{j-1}} \cap W_{i_{j-1}+2}| < \text{pw}(G)$ . Hence,  $(W_{i_1}, W_{i_2}, \dots, W_{i_m})$  also satisfies the condition of the theorem. For simplicity, we still use  $(W_1, W_2, \dots, W_r)$  to denote  $(W_{i_1}, W_{i_2}, \dots, W_{i_m})$ . We will use induction to show that each subgraph induced by  $W_i, 1 \leq i \leq r$ , can be cleared by an edge search strategy using at most  $k + 1$  searchers.

We first clear all edges in  $W_1$  using at most  $k + 1$  searchers. Since  $W_1 \not\subseteq W_2$ , there exists a vertex  $a \in W_1 \setminus W_2$ . Since  $|W_1| \leq k + 1$ , we know that  $\deg(a) \leq k$ . Hence, we can place a searcher on  $a$  and slide  $\deg(a)$  searchers from  $a$  to its neighbors to clear all edges incident on  $a$ . Remove the searcher on  $a$  and then place searchers to each vertex that is not a neighbor of  $a$ . We can use one free searcher to clear all remaining dirty edges in  $W_1$  and then remove some searchers such that each vertex in  $W_1 \cap W_2$  is occupied by one searcher and each vertex in  $W_1 \setminus W_2$  is not occupied by a searcher. Thus, we cleared all edges in  $W_1$  using at most  $k + 1$  searchers. Suppose we have cleared all edges in  $W_1, W_2, \dots, W_i$  one by one using at most  $k + 1$  searchers and immediately after all edges in  $W_i$  are cleared, each vertex in  $W_i \cap W_{i+1}$  is occupied by one searcher and each vertex in  $W_i \setminus W_{i+1}$  is not occupied by a searcher. We now consider clearing all edges in  $W_{i+1}$  using at most  $k + 1$  searchers. If  $i = r - 1$ , then  $W_{i+1}$  is the last bag in the path decomposition. Select a vertex  $u \in W_{r-1} \cap W_r$ . Since  $u$  has at most  $k$  neighbors in  $W_r$ , we can clear all dirty edges incident on  $u$  by sliding searchers from  $u$  to its neighbors. Remove the searcher on  $u$  and then place searchers on each vertex that is not a neighbor of  $u$ . We can use one free searcher to clear all remaining dirty edges in  $W_r$ . If  $i < r - 1$ , then we have two cases regarding  $W_i \cap W_{i+1}$ .

CASE 1.  $W_i \cap W_{i+1} \not\subseteq W_{i+2}$ . In this case, there must exist a vertex  $u \in (W_i \cap W_{i+1}) \setminus W_{i+2}$ . Since  $u$  is occupied by a searcher and it has at most  $k$  neighbors in  $W_{i+1}$ , we can clear all dirty edges incident on  $u$  by sliding searchers from  $u$  to its neighbors. Then we can remove the searcher on  $u$  and then place searchers to each vertex that is not a neighbor of  $u$  in  $W_{i+1}$ . We can use one free searcher to clear all remaining dirty edges in  $W_{i+1}$ .

CASE 2.  $W_i \cap W_{i+1} \subseteq W_{i+2}$ . Since  $W_{i+1} \not\subseteq W_{i+2}$ , there must exist a vertex  $u \in W_{i+1} \setminus W_{i+2}$ . Since  $W_i \cap W_{i+1} \subseteq W_{i+2}$ , we have  $u \notin W_i$ . From the condition (iii) in the definition of the path decomposition, we know that  $u \notin W_j$  for any  $j \neq i + 1$ . Since  $|W_i \cap W_{i+1}| = |W_i \cap W_{i+1} \cap W_{i+2}| \leq |W_i \cap W_{i+2}| < k$ , we have  $k + 1 - |W_i \cap W_{i+1}| \geq 2$ . Thus, immediately after all edges in  $W_i$  are cleared, each vertex in  $W_i \cap W_{i+1}$  is occupied by one searcher and we have at least two free searchers. Hence, we can place one free searcher on  $u$  and then use one free searcher to clear all edges between  $u$  and every vertex in  $W_i \cap W_{i+1}$ . We then clear all dirty edges incident on  $u$  by sliding searchers from  $u$  to its neighbors in  $W_{i+1} \setminus W_i$ . Remove the searcher on  $u$  and then place searchers to each vertex that is not a neighbor of  $u$  in  $W_{i+1}$ . We can use one free searcher to clear all remaining dirty edges in  $W_{i+1}$ .

After all edges in  $W_{i+1}$  are cleared using at most  $k + 1$  searchers, we remove some searchers such that each vertex in  $W_{i+1} \cap W_{i+2}$  is occupied by one searcher and each vertex in  $W_{i+1} \setminus W_{i+2}$  is not occupied by a searcher. By induction, we know that  $\text{es}(G) \leq \text{ss}(G) + 1$ . ■

**5.11 Corollary.** *For a connect graph  $G$ , if  $\text{es}(G) = \text{ss}(G) + 2$ , then for any path decomposition  $(W_1, W_2, \dots, W_r)$  of  $G$  with width  $\text{pw}(G)$ , there exists a subscript  $i$  such that  $|W_{i-1} \cap W_{i+1}| = \text{ss}(G)$ .*

## 6 Conclusion

In this paper, we proposed and studied the strong-mixed search, which is a generalization of the mixed search. We proved that the strong-mixed search is monotonic. We also proved that the strong-mixed search number of a graph equals the pathwidth of the graph. Finally, we described relationships between different search models.

This paper gives an approach to study the pathwidth by investigating the strong-mixed search. For convenience, we only considered connected finite graphs. In fact, all lemmas and theorems in Sections 3 and 4 hold for any finite graph with loops and multiple edges.

The strong-mixed search can be generalized to two new search problems. Let  $G$  be a connected finite graph embedded in  $\mathbb{E}^3$ . For any two vertices  $u, v \in V(G)$ , the distance of  $u$  and  $v$ , denoted by  $\text{dist}_G(u, v)$ , is the length of the shortest path between them. For an edge  $ab \in E(G)$ , the distance between  $v$  and  $ab$ , denoted by  $\text{dist}_G(v, ab)$ , is defined as  $\text{dist}_G(v, ab) =$

$\min\{\text{dist}_G(v, a), \text{dist}_G(v, b)\} + 1$ . We can first generalize the strong-mixed search from the visibility of searchers. A searcher on vertex  $v \in V(G)$  has *k-visibility* if he can see whether the intruder is hidden in the subgraph formed by the edge set  $E_v = \{ab \in E(G) : \text{dist}_G(v, ab) \leq k\}$ . In the strong-mixed search, if we grant each searcher *k-visibility*, then we may define a new search problem, called *k-visible search*. It is easy to see that 1-visible search is identical with the strong-mixed search.

We can also generalize the strong-mixed search from the capture ability of searchers. Let  $H \subseteq G$  be an induced connected subgraph with diameter at most  $k$ , and  $\partial(H)$  be the set of vertices in  $H$  with a neighbor in  $G - H$ . If each vertex of  $\partial(H)$  is occupied by at least one searcher, then all dirty edges and vertices in  $H$  become cleared. Combining this with the searchers' actions A1, A2 and A3, we may define a new search problem, called *k-strong-mixed search*. It is easy to see that 1-strong-mixed search is identical with the node search and 2-strong-mixed search is identical with the strong-mixed search.

For these two search problems, their properties and the relations with other search problems are totally unknown.

### Acknowledgments:

The author would like to extend special thanks to Yi Cao and Runtao Zhang for the discussion on pathwidth problems. The author would also like to thank Chris Worman and the referees for comments and suggestions.

### References

- [1] B. Alspach, Searching and sweeping graphs: a brief survey, *Le Matematiche*, 34pp (to appear).
- [2] L. Barrière, P. Flocchini, P. Fraigniaud, and N. Santoro, Capture of an intruder by mobile agents, *Proc. of the 14th ACM Symposium on Parallel Algorithms and Architectures (SPAA 2002)*, pp. 200–209, 2002.
- [3] D. Bienstock and P. Seymour, Monotonicity in graph searching, *Journal of Algorithms*, Vol.12, pp.239–245, 1991.
- [4] M. Fellows and M. Langston, On search, decision and the efficiency of polynomial time algorithm. *21st ACM Symp. on Theory of Computing (STOC 89)*, (1989) pp. 501–512.



- [5] M. Frankling, Z. Galil, and M. Yung. Eavesdropping games: A graph-theoretic approach to privacy in distributed systems, *Journal of ACM*, Vol.47, pp.225–243, 2000.
- [6] N.G. Kinnersley, The vertex separation number of a graph equals its path-width, *Information Processing Letters*, Vol.42, pp.345–350, 1992.
- [7] L.M. Kirousis and C.H. Papadimitriou, Searching and pebbling, *Theoretical Computer Science*, Vol.47, pp.205–216,, 1986.
- [8] A. S. LaPaugh, Recontamination does not help to search a graph. *Journal of ACM*, Vol.40, pp.224–245, 1993.
- [9] N. Megiddo, S. L. Hakimi, M. Garey, D. Johnson and C. H. Papadimitriou, The complexity of searching a graph, *Journal of ACM* Vol.35, pp.18–44, 1988.
- [10] T. D. Parsons, Pursuit-evasion in graphs, *Theory and Applications of Graphs*, Y. Alavi and D. R. Lick, eds., Springer, Berlin, pp.426–441, 1976.
- [11] N. Robertson and P. Seymour, Graph minors I: excluding a forest, *Journal of Combinatorial Theory Series B*, Vol.35, pp.39–61, 1983.
- [12] A. Takahashi, S. Ueno and Y. Kajitani, Mixed searching and proper-Path-Width, *Theoretical Computer Science*, Vol.137, pp.253–268, 1995.