GRAPH SEARCHING AND RELATED PROBLEMS

ANTHONY BONATO AND BOTING YANG

Abstract. Suppose that there is a robber hiding on vertices or along edges of a graph or digraph. Graph searching is concerned with finding the minimum number of searchers required to capture the robber. Major results of graph searching problems are surveyed, focusing on algorithmic, structural, and probabilistic aspects of the field.

1. Introduction

Graph searching is a hot topic in mathematics and computer science now, as it leads to a wealth of beautiful mathematics, and since it provides mathematical models for many real-world problems such as eliminating a computer virus in a network, computer games, or even counterterrorism. In all searching problems, there is a notion of *searchers* (or *cops*) trying to capture some *robber* (or *intruder*, or *fugitive*). A basic optimization question here is: What is the fewest number of searchers required to capture the robber? There are many graph searching problems motivated by applied problems or inspired by some theoretical issues in computer science and discrete mathematics. These problems are defined by (among other things) the class of graphs (for example, undirected graphs, digraphs, or hypergraphs), the actions of searchers and robbers, conditions of captures, speed, or visibility.

Graph searching has a variety of names in the literature, such as Cops and Robbers games, and pursuit-evasion problems. The reader is referred to survey papers [5, 23, 70, 72, 81] for an outline of the many graph searching models. A recent book by Bonato and Nowakowski [37] covers all aspects of Cops and Robbers games.

In this chapter, broad overview of graph searching is given, focusing on the most important results. The first part considers so-called *searching* games where the robber may occupy an edge or vertex, and the robber can usually move at high speeds at any time (but not always, depending on the model). The second part considers so-called *Cops*

The authors gratefully acknowledge support from NSERC, MITACS, Ryerson University, and the University of Regina.

and Robbers games where the cops and robber occupy only vertices and they move alternatively to their neighbors. Searching games are often related to various width-type parameters, while there is no such connection in Cops and Robbers. Owing to space constraints, proofs are omitted (however, references are given for all results). The results presented here span probabilistic, algorithmic, and structural results. As such, the chapter is not entirely self-contained; the reader can find most of the relevant background in the books [37, 55, 143]. Note that this chapter is the first reference to give a comprehensive overview of both searching and Cops and Robbers games (although searching is discussed briefly in [37]). The chapter finishes each part with a dozen problems and conjectures which are arguably the most important ones in the field of graph searching.

Let $G = (V, E)$ ($D = (V, E)$) denote a graph (or digraph) with vertex set V and edge set E. Use uv to denote an edge in a graph with two end vertices u and v and (u, v) to denote a directed edge in a digraph with tail u and head v. All graphs and digraphs considered in this chapter are finite and simple, unless otherwise stated.

Part 1. Searching Games

The moniker *searching* is used for the case when the robber can move at great speed at any time or at least when a searcher approaches him (not just move to a neighbor in his turn, as is the case in Cops and Robbers).

Searching has been linked in the literature to pathwidth and vertex separation of a graph [58, 97], to pebbling (and hence, to computer memory usage) [97], to assuring privacy when using bugged channels [64], to VLSI (that is, very large-scale integrated) circuit design [62], and to motion planning of multiple robots [136]. A large number of graph searching games have been introduced. We will focus here on some of the basic models.

Many graph searching games are closely related to graph width parameters, such as treewidth, pathwidth, and cutwidth. They provide some interpretation of width parameters, and often they can give a deeper insight into the graph structures and produce efficient algorithms. Examples include the proof of a min-max theorem on treewidth [131], the linear time algorithm for computing pathwidth of a tree [58, 95], the polynomial time algorithm for computing branchwidth of a planar graph [132], the linear time algorithm for computing cutwidth of a tree [106], and the computation of the topological bandwidth [46, 105].

Part 1 of the chapter surveys searching games on undirected graphs and digraphs. There are four sections in this part. Section 2 deals with the undirected graph searching games in which the robber can move at high speeds along any searcher-free path. Section 3 deals with the digraph searching games in which the robber usually moves at high speeds along a searcher-free directed path (depending on the game setting). Section 4 deals with more recently introduced searching games. As with the end of Part 2, Part 1 closes with a dozen of open problems in the area.

2. Searching undirected graphs

The first searching model was introduced by Parsons [117], after being approached by the author of [39] (a paper dealing with finding a spelunker lost in a system of caves). Let G be a connected, undirected graph embedded in $\mathbb{R}^3$ with the Euclidean distance metric such that no pair of edges intersect at a point that is not a common endpoint. Imagine there is a robber in G who can be located at any point of G; that is, anywhere along an edge or at a vertex. We want to capture the robber using the minimum number of searchers. For each positive integer k, let $\mathcal{C}_k(G)$ be the set of all families $F = \{f_1, f_2, \dots, f_k\}$ of continuous functions $f_i : [0, \infty) \to G$. A *continuous search strategy* for G is a family $F \in \mathcal{C}_k(G)$ such that for every continuous function $h : [0, \infty) \to G$ (corresponding to the robber), there is a $t_h \in [0, \infty)$ and $f_i \in F$ satisfying $h(t_h) = f_i(t_h)$. We say that f_i captures the robber at time t_h. The *continuous search number* of a graph G is the smallest k such that there exists a search strategy for G in $\mathcal{C}_k(G)$. There are some contexts for which Parsons searching is the model required. For example, in the case of searching for someone lost or a robber hiding in a cave system, the robber and the searchers move continuously. But the notion of Parsons searching presents certain difficulties because the model involves the action of continuous functions defined on the non-negative real numbers. In this survey, only discrete versions of Parsons searching are considered.

2.1. **Robber is invisible and active.**

When the robber is invisible and active, there are three basic searching games: edge searching, node searching, and mixed searching, which involves placing some restriction on searchers, but placing no restrictions on the robber. In these search games, the discrete time intervals (or time-steps) are introduced. Initially, G contains a robber who is located at a vertex in G, and G does not contain any searchers. Each searcher has no information on the whereabouts of the robber (that is, robber is *invisible*), but the robber has complete knowledge of the location of all searchers. The goal of the searchers is to capture the robber, and the goal of the robber is to avoid being captured. The robber always chooses the best strategy so that he evades capture. Suppose the game starts at time t_0 and the robber is captured at time t_N, and the search time is divided into N intervals $(t_0, t_1]$, $(t_1, t_2]$, ..., $(t_{N-1}, t_N]$ such that in each interval $(t_i, t_{i+1}]$ (also called *step*), exactly one searcher performs one action: placing, removing or sliding. The robber can move from a vertex x to a vertex 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 (that is, robber is *active*).

In the *edge search game* introduced by Megiddo et al. [108], there are three actions for searchers: placing a searcher on a vertex, removing a searcher from a vertex, and sliding a searcher along an edge from one end to the other. The robber is captured if a searcher and the robber occupy the same vertex on G.

In the *node search game* introduced by Kirousis and Papadimitriou [97], there are two actions for searchers: placing a searcher on a vertex and removing a searcher from a vertex. The robber is captured if a searcher and the robber occupy the same vertex of G or the robber is on an edge whose endpoints are both occupied by searchers.

In the *mixed search game* introduced by Bienstock and Seymour [24], searchers have the same actions as those in the edge search game. The robber is captured if a searcher and the robber occupy the same vertex on G or the robber is on an edge whose endpoints are both occupied by searchers.

The *edge search number* of G, denoted by $\mathrm{es}(G)$, is the smallest positive integer k such that k searchers can capture the robber. Analogously, define the *node search number* of G (written $\mathrm{ns}(G)$), and *mixed search number* of G (written $\mathrm{ms}(G)$). The following theorem demonstrates the relationships between search numbers.

Theorem 2.1 ([24, 97])**.** *If G is a connected graph, then the following inequalities hold.*

(1) $\mathrm{ns}(G) - 1 \leq \mathrm{es}(G) \leq \mathrm{ns}(G) + 1$.
(2) $\mathrm{ms}(G) \leq \mathrm{es}(G) \leq \mathrm{ms}(G) + 1$.
(3) $\mathrm{ms}(G) \leq \mathrm{ns}(G) \leq \mathrm{ms}(G) + 1$.

The following result shows that all three search problems are **NP**-complete.

Theorem 2.2 ([24, 97, 101, 108])**.** *Given a graph G and an integer k, the problem of determining whether $\mathrm{es}(G) \leq k$ ($\mathrm{ns}(G) \leq k$ or $\mathrm{ms}(G) \leq k$) is **NP**-complete.*

Megiddo et al. [108] only showed that the edge search problem is **NP**-hard. This problem belongs to **NP** owing to the monotonicity result of [101], in which LaPaugh showed that recontamination of edges cannot reduce the number of searchers needed to clear a graph. A search strategy is *monotonic* if the set of cleared edges before any step is always a subset of the set of cleared edges after the step. Monotonicity is an important issue in graph search problems. Bienstock and Seymour [24] 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 [71] provided a general framework that can unify monotonicity results in a unique minmax theorem.

Theorem 2.3 ([24, 101])**.** *The edge search, node search and mixed search problems are monotonic.*

Search numbers have close relationships with pathwidth and treewidth [126, 127]. Given a graph G, a *tree decomposition* of G is a pair (T, W) with a tree $T = (I, F)$, $I = \{1, 2, \ldots, m\}$, and a family of non-empty subsets $W = \{W_i \subseteq V : i = 1, 2, \ldots, m\}$, such that

(1) $\bigcup_{i=1}^{m} W_i = V$,
(2) for each edge $uv \in E$, there is an $i \in I$ with $\{u, v\} \subseteq W_i$, and
(3) for all $i, j, k \in I$, if j is on the path from i to k in T, then $W_i \bigcap W_k \subseteq W_j$.

The *width* of a tree decomposition (T, W) is

$$\max\{|W_i| - 1 : 1 \leq i \leq m\}.$$

The *treewidth* of G, denoted by $\mathrm{tw}(G)$, is the minimum width over all tree decompositions of G. A tree decomposition (T, W) is a *path decomposition* if T is a path; the *pathwidth* of a graph G, denoted by $\mathrm{pw}(G)$, is the minimum width over all path decompositions of G. One can find more information on treewidth and related problems in the survey papers [25, 124].

Another related graph parameter is the vertex separation introduced by Ellis et al. [58]. A *layout* of a connected graph G is a one to one mapping L: $V \to \{1, 2, \ldots, |V|\}$. Let $V_L(i) = \{x : x \in V$, and there exists $y \in V$ such that $xy \in E$, $L(x) \leq i$ and $L(y) > i\}$. The *vertex separation of* G *with respect to* L, denoted by $\mathrm{vs}_L(G)$, is defined as

$$\mathrm{vs}_L(G) = \max\{|V_L(i)| : 1 \leq i \leq |V|\}.$$

The *vertex separation* of G is defined as

$$\mathrm{vs}(G) = \min\{\mathrm{vs}_L(G) : L \text{ is a layout of } G\}.$$

Yang [144] introduced the strong-mixed search game, which is a generalization of the mixed search game. The strong-mixed search game has the same setting as the mixed search game except that searchers have an extra power to clear a neighborhood subgraph; that is, the subgraph induced by $N[v]$ (a set contains v and its neighbors) is cleared if all vertices in $N(v)$ (a set contains only the neighbors of v) are occupied by searchers. The *strong-mixed search number* of G, denoted by $\mathrm{sms}(G)$, is the smallest positive integer k such that k searchers can clear G. The following relationships were given in [95, 97, 144].

Theorem 2.4 ([95, 97, 144]). *If G is a connected graph, then*

$$\mathrm{sms}(G) = \mathrm{pw}(G) = \mathrm{vs}(G) = \mathrm{ns}(G) - 1.$$

For an edge search strategy, if the subgraph induced by cleared edges is connected in every step, then call it *connected search*. We denote by $\mathrm{cs}(G)$ and $\mathrm{mcs}(G)$ respectively, the connected and monotonic connected search numbers of a graph G defined in the natural way.

The following theorem from [149, 150] demonstrates that the connected search game is not monotonic.

Theorem 2.5 ([149, 150]). *For any positive integer k, there is a graph W_k such that* $\mathrm{cs}(W_k) = 280k + 1$ *and* $\mathrm{mcs}(W_k) = 290k$.

Fraigniaud and Nisse [63] investigated visible robber connected node search. They proved that recontamination does help as well to catch a visible robber in a connected way.

Barrière et al. [17] proved that $\mathrm{mcs}(T) \leq 2\mathrm{es}(T)$ for any tree T. Dereniowski [54] extended the result to graphs as follows.

Theorem 2.6 ([54]). *For any graph G,* $\mathrm{mcs}(G) \leq 2\mathrm{es}(G) + 3$.

Fomin et al. [69] introduced the domination search game, in which the robber hides only on vertices and searchers are placed or are removed from vertices of a graph. In the domination search game, searchers are more powerful than those in the node search game. A searcher "captures" the robber if she can "see" him; that is, a searcher on a vertex v can clear v and all its neighbors. The following theorem gives a relation between the *domination search number*, written $\mathrm{dsn}(G)$, node search number and the domination number, written $\gamma(G)$.

Theorem 2.7 ([69]). *For any graph G, let G' be a graph obtained from G by replacing every edge by a path of length three, and H be a graph obtained from G by connecting every two non-adjacent vertices by a path of length three. Then*

$$\mathrm{ns}(G) = \mathrm{dsn}(G') \text{ and } \gamma(G) \leq \mathrm{dsn}(H) \leq \gamma(G) + 1.$$

Kreutzer and Ordyniak [100] extended the domination game to distance d-domination games, in which each searcher can see a certain radius d around her position. That is, a searcher on vertex v can see any other vertex within distance d of v and if this vertex is occupied by the robber, then the searcher can see the robber and capture him. They showed that all questions concerning monotonicity and complexity about d-domination games can be reduced to the case of $d = 1$. They gave a class of graphs on which two searchers can win on any graph in this class but the number of searchers required for monotone winning strategies is unbounded. Due to the following result, the domination search problem is much harder than standard graph search problems. For background on **PSPACE** and other complexity classes, see Spiser [134].

Theorem 2.8 ([100]). *The domination search problem is* ***PSPACE****-complete. Moreover, the problem of deciding whether two searchers have a winning strategy on a graph is* ***PSPACE****-complete.*

2.2. **Robber is visible or lazy.** Dendris et al. [53] introduced the *lazy-robber game*, which has the same setting as the node searching game except that the robber stays only on vertices and moves just before a searcher is going to be placed on the vertex currently occupied by the robber (that is, robber is *lazy*). When a searcher is going to be placed on a vertex u currently occupied by the robber, the robber can move from u to another vertex v if there is a searcher-free path from u to v; otherwise, the robber is captured. The following result from Dendris et al. [53] shows that the lazy-robber game is monotonic and the search number of a graph in the lazy-robber game is equal to the treewidth of the graph plus one.

Theorem 2.9 ([53]). *Let G be a graph. For a lazy robber with unbounded speed, the monotonic search number of G is equal to the search number of G, and moreover, it is equal to* $\mathrm{tw}(G) + 1$.

Dendris et al. [53] also considered the speed-limited lazy robber. They showed that if the speed is 1, then the search number minus 1 is equal to a graph parameter, called width, which is polynomial-time computable for arbitrary graphs. Given a layout L of a connected graph G, the *width of a vertex* $v \in V$ with respect to the layout L is the number of vertices which are adjacent to v and precede v in the layout. The *width of the layout* L is the maximum width of a vertex of G. The *width of* G is the minimum width of a layout of G.

Theorem 2.10 ([53]). *Let G be a graph. For a lazy robber with speed 1, the monotonic search number of G is equal to the search number of G, and moreover, it is equal to the width of G plus 1.*

Seymour and Thomas [131] introduced another variant of graph searching, *visible-robber game*. Its setting differs from node search in that the robber stands only on vertices and is visible to searchers. They showed that the visible-robber game is monotonic and the search number of a graph in the visible-robber game is equal to the treewidth of the graph plus one. They also showed that if there is a non-losing strategy for the robber, then there is a "nice" non-losing strategy with a particularly simple form. In order to show these results, Seymour and Thomas introduced jump-search and haven.

For a graph G and $X \subseteq V$, let $G - X$ be the graph obtained from G by deleting X. The vertex set of a component of $G - X$ is called an *X-flap*. Let $[V]^{<k}$ be the set of all subsets of V of cardinality less than k. The following lemma describes a strategy for the robber.

Lemma 2.11 ([131]). *A graph G cannot be cleared by less than k searchers if and only if there is a function σ mapping each $X \in [V]^{<k}$ to a non-empty union $\sigma(X)$ of X-flaps, such that if $X \subseteq Y \in [V]^{<k}$, then $\sigma(X)$ is the union of all X-flaps which intersect $\sigma(Y)$.*

Two vertex sets $X, Y \subseteq V(G)$ *touch* if either $X \cap Y \neq \emptyset$ or some vertex in X has a neighbor in Y. In *jump-searching*, each searcher can jump from a vertex to another vertex. At the start of the ith step, searchers occupy $X_{i-1} \in [V]^{<k}$ and the robber is in the subgraph R_{i-1}, which is an X_{i-1}-flap. After some searchers jump, the searchers occupy $X_i \in [V]^{<k}$ and the robber chooses an X_i-flap R_i, which touches R_{i-1}. Havens correspond to particularly nice winning strategies for the robber. A *haven* of order k is a function β which assigns an X-flap $\beta(X)$ to each $X \in [V]^{<k}$, in such a way that $\beta(X)$ touches $\beta(Y)$ for all $X, Y \in [V]^{<k}$. A *screen* in G is a set of subsets of $V(G)$ such that each subset induces a connected subgraph of G and any two subsets touch each other. A screen S has thickness at least k if there is no $X \in [V]^{<k}$ such that $X \cap H \neq \emptyset$ for all $H \in S$. The following result from Seymour and Thomas [131] demonstrates the relations among screen, haven, visible-robber search, monotonic visible-robber search, and treewidth.

Theorem 2.12 ([131]). *For a graph G and an integer $k \geq 1$, the following are equivalent.*

(1) *G has a screen of thickness at least k.*
(2) *G has a haven of order at least k.*

(3) *Fewer than k searchers cannot jump-search clear G.*
(4) *Fewer than k searchers cannot clear G,*
(5) *Fewer than k searchers cannot monotonically clear G,*
(6) *G has treewidth at least $k - 1$.*

The problem of determining whether k searchers can capture the robber in the visible-robber game (or lazy-robber game) is **NP**-complete since it is monotonic and computing treewidth (equivalently, partial k-tree) is **NP**-complete [13].

Theorem 2.13 ([13])**.** *Given a graph G and a positive integer k, the problem of determining whether the treewidth of G is at most k is **NP**-complete.*

There are many variants of graph searching games, for example, time constrained searching [6], network security searching [16], weighted graphs searching [153], robber-and-marshals games [79], geometric environment searching [137, 138, 139], and more searching games on undirected graphs can be found in Section 4.

3. Searching digraphs

3.1. **Robber is visible or lazy.** Johnson et al. [90] generalized the concept of treewidth to digraphs. For a digraph D, let Z and S be two disjoint subsets of V. The set S is *Z-normal* if for every directed path in D with first and last vertices in S, all vertices of the path belong to $S \cup Z$. An *arborescence* is a directed tree T with edges oriented away from a unique vertex $r \in V(T)$ (called the root). We write $t > e$ for $t \in V(T)$ and $e \in E(T)$ if e occurs on the unique directed path from r to t, and $e \sim t$ if e is incident with t. An *arboreal decomposition* of a digraph D is a triple (T, X, W) where T is an arborescence, and $X = \{X_e \subseteq V(D) : e \in E(T)\}$ and $W = \{W_t \subseteq V(D) : t \in V(T)\}$ satisfy the following items.

(1) W is a partition of $V(D)$ into non-empty sets, and
(2) If $e \in E(T)$, then $\bigcup\{W_t : t \in V(T) \text{ and } t > e\}$ is X_e-normal.

The width of an arboreal decomposition (T, X, W) is the minimum k such that for all $t \in V(T)$,

$$\left| W_t \cup \bigcup_{e \sim t} X_e \right| \leq k + 1.$$

The *directed treewidth* of D is the least integer k such that D has an arboreal decomposition of width k.

Johnson et al. [90] introduced a generalization of the visible-robber game, called the *strong-directed visible-robber game*. In the strong-directed visible-robber game, the robber occupies only vertices and must obey the edge directions when he moves along edges. The searchers have two actions – placing on vertices and removing from vertices. The robber is visible, and the robber can move from vertex u to v if there is a searcher-free directed cycle containing u and v.

Theorem 3.1 ([90])**.** *Let D be a digraph and k be an integer. If D has directed treewidth less than k, then k searchers have a winning strategy in the strong-directed visible-robber game.*

Johnson et al. [90] gave an example to show that the winning strategy in Theorem 3.1 need not be searcher-monotone in the sense that searchers may have to revisit certain vertices. Adler [1] showed that it may not be robber-monotone either. She constructed a digraph where four searchers have a winning strategy but they have no robber-monotone winning strategy. This is very different from the undirected case, where a graph has treewidth less than k implies that k searchers have a winning strategy where no vertex is revisited once it has been vacated. The following theorem gives a relation between the treewidth and directed treewidth.

Theorem 3.2 ([90])**.** *Let G be a graph, and let D be the digraph obtained from G by replacing every edge with two directed edges directed in opposite directions. Then the directed treewidth of D is equal to the treewidth of G.*

Let $k \geq 1$ be an integer. A *haven* of order k in a digraph D is a function β assigning to every $Z \subseteq V(D)$ with $|Z| < k$ the vertex set of a strong component of $D - Z$ in such a way that if $Z' \subseteq Z \subseteq V(D)$ with $|Z| < k$, then $\beta(Z) \subseteq \beta(Z')$. In the strong-directed visible-robber game, if $k-1$ searchers have a winning strategy on the digraph D, then D has no haven of order k. If β is a haven of order k in D, then the robber wins against $k-1$ searchers by staying in $\beta(Z)$, where Z is the set of vertices occupied by the searchers. Johnson et al. [90] proved the following relation between the directed treewidth and haven.

Theorem 3.3 ([90])**.** *Let D be a digraph and k be a positive integer. If D has a haven of order k, then its directed treewidth is at least $k-1$.*

Adler [1] showed that the converse of Theorem 3.3 is not true by constructing a digraph whose directed treewidth is at least 4 but it has no haven of order 5. Johnson et al. [90] showed the following upper bound.

Theorem 3.4 ([90])**.** *Let D be a digraph and k be a positive integer. Then either D has directed treewidth at most $3k-2$ or it has a haven of order k.*

Corollary 3.5 ([90])**.** *Let D be a digraph and k be a positive integer. Then either D has directed treewidth at most $3k-1$ or k searchers do not have a winning strategy in the strong-directed visible-robber game on D.*

Berwanger et al. [20] and Obdržálek [116] introduced another width measure for digraphs, called *DAG-width*. Given a digraph D, a set $X \subseteq V$ guards a set $Y \subseteq V$ if $X \cap Y = \emptyset$ and whenever there is an edge $(u, v) \in E$ such that $u \in Y$ and $v \notin Y$, then $v \in X$. A DAG-decomposition of D is a pair (T, W) with T an acyclic digraph and $W = \{W_t \subseteq V : t \in V(T)\}$ a family of non-empty subsets, such that the following hold.

(1) $\bigcup_{t \in V(T)} W_t = V$.
(2) For all $t, t', t'' \in V(T)$, if t' is on a directed path from t to t'' in T, then $W_t \bigcap W_{t''} \subseteq W_{t'}$.
(3) For all edges $(t, t') \in E(T)$, $W_t \bigcap W_{t'}$ guards $W_{\geq t'} \setminus W_t$, where $W_{\geq t'} = \bigcup\{W_{t''} : t'' \in V(T)$ and there is a directed path from t' to t'' in $T\}$. For all roots $r \in V(T)$, $W_{\geq r}$ is guarded by $\emptyset$.

The *width of a DAG-decomposition* (T, W) is

$$\max\{|W_t| : t \in V(T)\}.$$

The *DAG-width* of D is the minimum width of any of its DAG-decompositions.

Berwanger et al. [20] and Obdržálek [116] introduced the *directed visible-robber game*, which has the same setting as the strong-directed visible-robber game except that the robber is more powerful. The robber is allowed to move from vertex u to v if there is a searcher-free directed path from u to v. They use this game to characterize the DAG-width.

The directed visible-robber game and the strong-directed visible-robber game have some different properties owing to the different behavior of robbers. One difference is that the search number in the strong-directed visible-robber game is invariant under edge reversal; that is, it does not change if the directions of all edges of the digraph are reversed. But this is not the case for the directed visible-robber game. Another difference is that for the strong-directed visible-robber game with k searchers, there are digraphs that have a robber-monotone winning strategy, but no searcher-monotone winning strategy. However, Berwanger et al. [20] proved the following.

Theorem 3.6 ([20]). *For the directed visible-robber game with k searchers, if the searchers have a searcher-monotone or robber-monotone winning strategy, then they also have a winning strategy that is both searcher- and robber-monotone.*

A *monotone strategy* is a strategy that is both searcher-monotone and robber-monotone. Berwanger et al. [20] and Obdržálek [116] established the following relation between the search number and the DAG-width.

Theorem 3.7 ([20, 116]). *Let D be a digraph and k be a positive integer. Then D has a DAG-decomposition of width k if and only if k searchers have a monotone winning strategy in the directed visible-robber game on D.*

Similar to Theorem 3.2, Berwanger et al. proved the following.

Theorem 3.8 ([20]). *Let G be a graph, and let D be the digraph obtained from G by replacing every edge with two directed edges directed in opposite directions. Then the DAG-width of D is equal to the treewidth of G minus one.*

Berwanger et al. [20] also proved that if a digraph has DAG-width k, its directed treewidth is at most $3k + 1$. However, there is a family of digraphs with directed treewidth 1 and arbitrarily large DAG-width. Kreutzer and Ordyniak [99] proved that the directed visible-robber game is non-monotone.

Theorem 3.9 ([99]). *For any positive integer $k \geq 2$, there is a digraph D_k such that in the directed visible-robber game on D_k, the monotonic search number of D_k is $4k - 2$ and the search number of D_k is $3k - 1$.*

Hunter and Kreutzer [88] extended the lazy-robber game to the *directed lazy-robber game* on digraphs, which has the same setting as the lazy-robber game on graphs except that the robber must obey the edge directions when he moves along edges. They use this game to characterize the digraph width measure, called *Kelly-width*.

Given a digraph D, a *Kelly-decomposition* of D is a triple (T, B, W) with T an acyclic digraph, $B = \{B_t \subseteq V : t \in V(T)\}$ and $W = \{W_t \subseteq V : t \in V(T)\}$, such that the following properties hold.

1. The set B is a partition of V into non-empty sets.
2. For all $t \in V(T)$, W_t guards $B_{\geq t}$, where $B_{\geq t} = \bigcup\{B_{t'} : t' \in V(T)$ and there is a directed path from t to t' in $T\}$.

(3) For all $s \in V(T)$, there is a linear order on its children $t_1, \ldots, t_p$ such that for all $1 \le i \le p$,

$$W_{t_i} \subseteq B_s \cup W_s \cup \bigcup_{j<i} B_{\ge t_j}.$$

Similarly, there is a linear order on roots $r_1, \ldots, r_q$ of T such that for all $1 \le i \le q$,

$$W_{r_i} \subseteq \bigcup_{j<i} B_{\ge r_j}.$$

The *width of a Kelly-decomposition* (T, B, W) is

$$\max\{|B_t \cup W_t| : t \in V(T)\}.$$

The *Kelly-width* of D is the minimum width of any of its Kelly-decompositions.

The following result from Hunter and Kreutzer [88] shows that k searchers have a monotone winning strategy to capture a lazy robber in a digraph D if and only if D has Kelly-width k.

Theorem 3.10 ([88]). *Let D be a digraph and k be a positive integer. Then D has a Kelly-decomposition of width k if and only if k searchers have a monotone winning strategy in the directed lazy-robber game on D.*

Hunter and Kreutzer [88] gave the following relation between the directed lazy-robber game and the directed visible-robber game.

Theorem 3.11 ([88]). *Let D be a digraph and k be a positive integer. If k searchers have a robber-monotone winning strategy in the directed lazy-robber game on D, then $2k-1$ searchers have a winning strategy in the directed visible-robber game on D.*

Theorem 3.12 ([88]). *Let D be a digraph and k be a positive integer. If k searchers have a monotone winning strategy in the directed visible-robber game on D, then k searchers have a winning strategy in the directed lazy-robber game on D.*

Kreutzer and Ordyniak [99] proved that the directed lazy-robber game is non-monotone.

Theorem 3.13 ([99]). *For any positive integer $k \ge 2$, there is a digraph D_k such that in the directed lazy-robber game on D_k, the monotonic search number of D_k is $7k$ and the search number of D_k is $6k$.*

The problem of determining whether k searchers can capture the robber in the directed visible-robber game (resp. directed lazy-robber game, or strong-directed visible-robber game) is **NP**-hard.

Besides the above width measures, several new digraph measures are introduced recently, which include K-width [76], DAG-depth [76], entanglement [21], D-width [128], directed pathwidth [15], directed vertex separation [148], and bi-rank-width [92]. See also the thesis of Hunter for more discussions [87].

3.2. **Robber is invisible and active.** Reed, Seymour and Thomas introduced the directed pathwidth. Given a digraph D, a directed path decomposition of D is a sequence of subsets of vertices $W_1, W_2, \ldots, W_m$ such that the following hold.

(1) $\bigcup_{i=1}^{m} W_i = V$.
(2) For $1 \leq i < j < k \leq m$, $W_i \bigcap W_k \subseteq W_j$.
(3) For each edge in D, it either has both endpoints in the same W_i or has its tail in W_i and head in W_j, where $i < j$.

The *width of a directed path decomposition* of D is

$$\max_{1 \leq i \leq m} \{|W_i| - 1\}.$$

The *directed pathwidth* of D, denoted by $\mathrm{dpw}(D)$, is the minimum width over all directed path decompositions of D.

Barát [15] introduced a generalization of the node searching, called the *directed node search game*, to characterize the directed pathwidth. The directed node search game has the same setting as the node search game on graphs except that the robber occupies only vertices and must obey the edge directions when he moves along edges. 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 also proved that the directed pathwidth and the directed node search number differ by at most one. Specifically, he proved the following theorem.

Theorem 3.14 ([15])**.** *For a digraph D and an integer $k \geq 1$, the following downward implications apply.*

(1) *There is a monotone capture of the robber in D with at most k searchers.*
(2) *The directed pathwidth of D is at most $k - 1$.*
(3) *There is a capture of the robber in D with at most k searchers.*
(4) *There is a monotone capture of the robber in D with at most $k + 1$ searchers.*

Yang and Cao [145, 146, 147] generalized the edge searching problem to digraphs by introducing three digraph search games: directed search, strong search and weak search. In all three games, the robber is invisible and searchers have three types of actions: placing, removing, and sliding. These games differ in the abilities of the searchers and robber depending on whether or not they must obey the edge directions. In the *directed search* game, both searchers and robber must move in the edge directions; in the *strong search* game, the robber must move in the edge directions but searchers need not; and in the *weak search* game, searchers must move in the edge directions but the robber need not. In particular, in the directed and strong search games, the robber can move from vertex u to vertex v along a searcher-free directed path from u to v at a great speed at any time; and in the weak search game, the robber can move from vertex u to vertex v along a searcher-free undirected path between u and v at a great speed at any time.

For a digraph D, let $\mathrm{ds}(D)$, $\mathrm{ss}(D)$, and $\mathrm{ws}(D)$ denote the *directed*, *strong*, and *weak search number*, respectively. Let $\mathrm{mds}(D)$, $\mathrm{mss}(D)$, and $\mathrm{mws}(D)$ denote the monotonic directed, monotonic strong, and monotonic weak search number respectively. The following result shows that the directed, strong, and weak search problems are all monotonic and **NP**-complete.

Theorem 3.15 ([145, 146, 147])**.** *Let D be a digraph and k be a positive integer. Then* $\mathrm{mds}(D) = \mathrm{ds}(D)$, $\mathrm{mss}(D) = \mathrm{ss}(D)$ *and* $\mathrm{mws}(D) = \mathrm{ws}(D)$. *Moveover, the problem of deciding whether* $\mathrm{ds}(D) \leq k$, $\mathrm{ss}(D) \leq k$, *or* $\mathrm{ws}(D) \leq k$ *is* ***NP****-complete.*

Let $\mathrm{es}(D)$ be the edge search number of the underlying undirected graph of D. Yang and Cao gave the following relationships between search numbers.

Theorem 3.16 ([145, 146, 147])**.** *Let D be a digraph. Then the following inequalities hold.*

(1) $\mathrm{ss}(D) \leq \mathrm{ds}(D) \leq \mathrm{ws}(D)$.
(2) $\mathrm{ss}(D) \leq \mathrm{es}(D) \leq \mathrm{ws}(D)$.
(3) $\mathrm{ds}(D) - 1 \leq \mathrm{ss}(D) \leq \mathrm{ds}(D)$.
(4) $\mathrm{ds}(D) \leq \mathrm{es}(D) + 1$.
(5) $\mathrm{ws}(D) \leq \mathrm{es}(D) + 2$.

Nowakowski [114] and Alspach et al. [7] introduced another three digraph search games: internal directed search, internal strong search and internal weak search, which have the same setting as the directed search, strong search and weak search, respectively, except that in the internal games, searchers have only two types of actions: placing and sliding. Yang and Cao [145] proved that the internal directed search problem is monotonic, however, the internal strong and internal weak search problems are not monotonic. Although the internal strong search game is not monotonic, they still proved it is in **NP**. But the problem of whether the internal weak search game is in **NP** is still open.

Theorem 3.17 ([145])**.** *Given a digraph D and an integer k, the problem of determining whether k searchers can capture the robber in the internal directed (or strong) search game is **NP**-complete, and the problem of determining whether k searchers can capture the robber in the internal weak search game is **NP**-hard.*

Yang and Cao [148] extended the vertex separation to digraphs. Given a digraph D and a linear layout $L : V \rightarrow \{1, 2, ..., |V|\}$, let $DV_L(i) = \{x \in V$: there exists $y \in V$ such that edge $(y, x) \in E$ and $L(x) \leq i$ and $L(y) > i\}$. The directed vertex separation of D with respect to L, denoted by $\mathrm{dvs}_L(D)$, is defined as

$$\mathrm{dvs}_L(D) = \max\{|DV_L(i)| : 1 \leq i \leq |V|\}.$$

The *directed vertex separation of* D is defined as $\mathrm{dvs}(D) = \min\{\mathrm{dvs}_L(D) : L$ is a linear layout of $D\}$. The following relationships were given in [148].

Theorem 3.18 ([148])**.** *Let D be a digraph. Then the following inequalities hold.*

(1) $\mathrm{dvs}(D) = \mathrm{dpw}(D)$.
(2) $\mathrm{dvs}(D) + 1 \leq \mathrm{ds}(D) \leq \mathrm{dvs}(D) + 2$.
(3) $\mathrm{dvs}(D) \leq \mathrm{ss}(D) \leq \mathrm{dvs}(D) + 2$.

4. Other searching games

4.1. **Mixed searching with multiple robbers.** As mentioned in Section 2.1, Bienstock and Seymour [24] introduced the mixed search game, in which searchers have three actions: placing, removing and sliding. The mixed search game can be used to characterize the proper pathwidth introduced by Takahashi et al. [140].

A graph G is a *minor* of graph H if G can be obtained from a subgraph of H by edge contractions, where a contraction of edge uv is the deletion of uv, followed by identifying u and v such that all vertices adjacent u or v is adjacent to the new vertex. The *proper pathwidth* of a graph G, denoted as $\mathrm{ppw}(G)$, is the least integer k such that G is a minor of the graph $K_k \Box P$ for some path P, where K_k is a clique of order k and $\Box$ is the cartesian product operator. Similarly, the *proper treewidth* of a graph G, denoted as $\mathrm{ptw}(G)$, is the least integer k such that G is a minor of the graph $K_k \Box T$ for some tree T.

The following result from Takahashi et al. [141] shows the relation between the mixed search number and the proper pathwidth.

Theorem 4.1 ([141]). *For any graph G,* $\mathrm{ms}(G) = \mathrm{ppw}(G)$.

Richerby and Thilikos [125] studied the mixed search game with multiple robbers who stay only on vertices. A robber is captured if a searcher and the robber occupy the same vertex of G. A set of searchers win the game if they can capture all robbers, otherwise, robbers win.

For a graph G with $\mathrm{ms}(G) = k$, if monotonicity is ignored, then k searchers can capture any number of visible active robbers, one at a time by repeating the strategy to capture a single robber. So, monotonicity is crucial in the mixed search game with multiple robbers. The robbers' territory F_i at step i is the union of each robber's territory. A search strategy is *monotonic* if $F_{i+1} \subseteq F_i$ for all i. Let $\mathrm{mvams}(G, r)$ denote the *monotonic mixed search number for r visible active robbers* in G, $\mathrm{vams}(G, r)$ denote the *mixed search number for r visible active robbers* in G, $\mathrm{milms}(G, r)$ denote the *monotonic mixed search number for r invisible lazy robbers* in G, and $\mathrm{ilms}(G, r)$ denote the *mixed search number for r invisible lazy robbers* in G. Richerby and Thilikos showed the following relations.

Theorem 4.2 ([125]). *For any graph G with n vertices,*

(1) $\mathrm{mvams}(G, n) = \mathrm{ms}(G) = \mathrm{ppw}(G)$.
(2) $\mathrm{mvams}(G, 1) = \mathrm{milms}(G, 1)$.
(3) $\mathrm{ilms}(G, r) = \mathrm{ilms}(G, 1)$.
(4) $\mathrm{milms}(G, r) = \mathrm{milms}(G, 1) = \mathrm{ptw}(G)$.
(5) $\mathrm{mvams}(G, r) \le \min\{\mathrm{ppw}(G), \mathrm{ptw}(G)(\lfloor \log r \rfloor + 1)\}$.

Theorem 4.3 ([125]). *For any tree T with r robbers,*

$$\mathrm{mvams}(T, r) = \min\{\mathrm{ppw}(T), \lfloor \log r \rfloor + 1\}.$$

4.2. **Non-deterministic graph searching.** Fomin et al. [65] introduced non-deterministic graph searching, which can be used in algorithm design and combinatorial analysis applying to both pathwidth and treewidth. In the non-deterministic graph searching, the searchers and robber stay only on vertices. There are three actions for searchers: placing a searcher on a vertex, removing a searcher from a vertex, and query the oracle that returns the connected component containing the robber. The robber has the same behavior as the node search game. For a non-negative integer q, a q-limited non-deterministic searching is a non-deterministic searching that performs at most q query actions. Therefore, k searchers win a q-limited non-deterministic search game against a robber if they can capture the robber by querying the oracle at most q times. The q-limited non-deterministic search number of a graph G is the minimum number of searchers required to win the game.

Fomin et al. [65] extended the tree decomposition and treewidth to q-branched tree decomposition and q-branched treewidth. They used the non-deterministic graph search game to characterize the q-branched treewidth. A *rooted tree decomposition* of a graph G is a tree decomposition (T, W) of G in which T is a rooted tree and W is a family of non-empty subsets of $V(G)$ satisfying the three conditions of a tree decomposition. A *branching node* of a rooted tree decomposition is a node with at least 2 children. For a non-negative integer q, a *q-branched tree decomposition* of a graph G is a rooted tree decomposition (T, W) of G such that every path in T from the root to a leaf contains at most q branching nodes. The *q-branched treewidth* of G, is the minimum width of any q-branched tree decomposition of G. The following theorem from Fomin et al.[65] shows the relation between the non-deterministic graph search game and the q-branched treewidth.

Theorem 4.4 ([65]). *For a graph G, a non-negative integer q and a positive integer k, at most k searchers win the q-limited non-deterministic search game in a monotonic way if and only if the q-branched treewidth is less than k.*

Fomin et al. used the q-limited non-deterministic graph searching to design an exact exponential-time algorithm for computing the q-branched treewidth of a graph.

Theorem 4.5 ([65]). *For a graph G with n vertices, there exists an algorithm that computes the q-branched treewidth and its corresponding optimal q-branched tree decomposition of G in time $O(2^n n \log n)$.*

Mazoit and Nisse [107] showed that the q-limited non-deterministic graph searching is monotone for any non-negative integer q.

Theorem 4.6 ([107]). *For a graph G, a non-negative integer q and an integer $k \geq 2$, k searchers win the q-limited non-deterministic search game if and only if they can do it in a monotonic way.*

4.3. **Fast searching.** In the edge search game, the goal is to find the minimum number of searchers to clear a given graph. Yang [151] introduced a new game called the *fast edge-searching*, which has the same setting as the edge search game except the goal. In the fast edge-search game, the goal is to find the minimum number of steps (or equivalently, actions) to clear a given graph. The fast edge-search game has a strong connection with the fast search game, which was first introduced by Dyer et al. [57]. The fast search game has the same setting as the edge search game except that every edge is traversed exactly once by a searcher and searchers cannot be removed.

The motivation to consider the fast edge-searching and fast searching is that, in some real-life scenarios, the cost of a searcher may be relatively low in comparison to the cost of allowing a fugitive to be free for a long period of time. For example, if a dangerous fugitive hiding along streets in an area, policemen always want to capture the fugitive as soon as possible.

In the fast edge-searching game, the minimum number of steps required to clear G is the *fast edge-search time* of G, denoted by $\mathrm{fet}(G)$, and the minimum number of searchers required so that G can be cleared in $\mathrm{fet}(G)$ steps is the *fast edge-search number* of G, denoted by $\mathrm{fen}(G)$. Similarly, the minimum number of searchers required to clear G in the fast search game is the *fast search number* of G, denoted by $\mathrm{fsn}(G)$. A difference between the two games is that the family of graphs $\{G : \mathrm{fet}(G) \leq k\}$ is minor-closed for the fast edge-searching, but the family of graphs $\{G : \mathrm{fsn}(G) \leq k\}$ is not minor-closed for the fast searching. The difference between $\mathrm{fen}(G)$ and $\mathrm{fsn}(G)$ can be large. In fact, there exists a class of graphs H such that the ratio $\mathrm{fsn}(H)/\mathrm{fen}(H)$ is arbitrarily large. Yang gave the following relationship between the fast edge-searching game and the fast searching game.

Theorem 4.7 ([151]). *For any graph G,*

$$\mathrm{fet}(G) = \mathrm{fsn}(G) + |E|.$$

For a graph G, let G' be a graph obtained from G by adding a vertex a and connecting it to each vertex of G. Let A'_G be a multigraph obtained from G' by replacing each edge with four parallel edges. Let A_G be a graph obtained from A'_G by replacing each edge of A'_G with a path of length 2. Then the following relation between the node search number of G and the fast search number of A_G was given in [151].

Theorem 4.8 ([151]). *For a graph G and its corresponding graph A_G,*

$$\mathrm{ns}(G) = \mathrm{fsn}(A_G) - 2 = \mathrm{fen}(A_G) - 2.$$

Although the node search number and the fast search number have the above relation, Stanley and Yang [135] presented a linear time algorithm for computing the fast search number of cubic graphs, while it is **NP**-complete to find the node search number of cubic graphs [105].

Corollary 4.9. *The fast search problem and the fast edge-search problem are **NP**-complete. They remain **NP**-complete for Eulerian graphs.*

Yang showed the following bounds for the fast search number.

Theorem 4.10 ([151]). *For a connected graph G with minimum degree at least* 3,

$$\max\left\{\delta(G)+1, \left\lceil\frac{\delta(G)+|V_{\mathrm{odd}}(G)|-1}{2}\right\rceil\right\} \leq \mathrm{fsn}(G) \leq |E|,$$

where $\delta(G)$ is the minimum degree of G and $V_{\mathrm{odd}}(G)$ is the set of all odd vertices in G.

The fast searching game has a close relation with the graph brushing problem [109, 3, 78] and the balanced vertex-ordering problem [22, 93]. For all graphs, the brush number is equal to the total imbalance of an optimal vertex-ordering. For some graphs, such as trees, the fast search number is equal to the brush number.

4.4. **Graph guarding.** Fomin et al. [67] introduced the graph guarding games, in which a set of cops want to guard a region in a given graph against a robber. The robber and cops stay only on vertices of the graph, and both sides take alternative turns to play. All participants have complete information on the location of all other participants. Initially, the robber is placed on a vertex outside the protected region, and then all cops are placed on vertices inside the protected region. In robber's turn, he can move to any adjacent vertex. The robber wins if he can move to an unguarded vertex in the region. In the cops' turn, each cop can move to an adjacent vertex in the region or stay put. The cops win if they can forever prevent the robber to win. The guarding problem is to find the minimum number of cops needed to win the game.

Fomin et al. [67] showed that the guarding problem is polynomial time solvable if the robber's region is a path. Nagamochi [113] showed that the guarding problem is polynomial time solvable if the robber's region is a cycle.

Theorem 4.11 ([67]). *For a graph $G = (V, E)$, let C be the protected region (vertex set). If $V \setminus C$ induces a path in G, then the guarding problem can be solved in $O(n_1 n_2 m)$ time, where $n_1 = |C|$, $n_2 = |V \setminus C|$ and $m = |E|$.*

Fomin et al. showed that the guarding problem is **NP**-hard even if the robber's region is a star.

Theorem 4.12 ([67]). *Given a graph $G = (V, E)$, a protected region C (vertex set), and an integer k, it is **NP**-hard to decide whether k cops can win the guarding game. It remains **NP**-hard even if $V \setminus C$ induces a star. Moreover, the parameterized version of the problem with k (the number of cops) being a parameter is **W[2]**-hard.*

The guarding game can also play on digraphs, in which the robber and cops always follow edge directions when they move to neighbors.

Theorem 4.13 ([67])**.** *Given a digraph $D = (V, E)$, a protected region C (vertex set), and an integer k, if $V \setminus C$ induces an acyclic digraph, it is **PSPACE**-complete to decide whether k cops can win the guarding game.*

In the guarding game, the robber moves first. In [68], Fomin et al. also studied a variant of the guarding game, in which cops move first. They showed that the new guarding problem is **PSPACE**-hard on undirected graphs.

4.5. **Minimum cost searching.** Almost all graph search problems are to find search strategies such that the maximum number of searchers used at each step is minimized. Fomin and Golovach [66] proposed a different optimization criterion. They introduced a cost function in the node searching game, which is the sum of the number of searchers in every step of the node search process.

One can interpret the cost of a search as the total sum that searchers earn for doing their job. The *search cost* of a graph G, denoted by $\sigma(G)$, is the minimum cost of a search where the minimum is taken over all searches on G. For monotonic node searching, the *monotonic search cost* of G, denoted by $\sigma_m(G)$, can be defined similarly. Fomin and Golovach [66] proved their minimum cost searching is monotonic.

Theorem 4.14 ([66])**.** *For any graph G, $\sigma(G) = \sigma_m(G)$.*

Kirousis and Papadimitriou [96] found a relation between node searching and interval graphs. An interval graph is one that has an interval model, which is a set of intervals of the real line, one for each vertex, such that two intervals intersect if and only if the corresponding vertices are adjacent. Every graph is a subgraph of an interval graph in a trivial way. The *interval thickness* of a graph G, denoted by $\theta(G)$, is the smallest max-clique over all interval supergraphs of G.

Theorem 4.15 ([96])**.** *For any graph G, $\mathrm{ns}(G) = \theta(G)$.*

Fomin and Golovach proved the following relation between minimum cost searching and interval graphs.

Theorem 4.16 ([66])**.** *For any graph G and positive integer k, $\sigma(G) \leq k$ if and only if there is an interval supergraph I of G such that $E(I) \leq k$.*

Dyer et al. [57] introduced the following cost function in the edge searching game on a graph G:

$$C_G(s,t) = \alpha s + \beta st + \gamma t,$$

where $\mathrm{es}(G) \leq s \leq \mathrm{fsn}(G)$ and $t = t(s) \geq |E|$. The goal is to minimize the cost function C_G, instead of trying to minimize s, which corresponds to edge searching, or to minimize t, which corresponds to fast searching.

4.6. **Pebbling.** Pebble games on graphs and digraphs have been studied by both mathematicians and computer scientists [18, 47, 52, 85, 97, 110, 111]. Pebbling is a method of analyzing computational situations, especially situations in which notions such as time (number of operations) and space (number of memory locations) are of interest [51, 86, 91, 94, 98, 102]. We focus on the relationship between graph searching and pebbling [97].

Let D be an acyclic digraph. For an edge $(u, v) \in E(D)$, v is called a *successor* of u, and u is called a *predecessor* of v. We consider two versions of pebbling: one is black pebbling and the other is black and white pebbling. The rules for *black pebbling* are as follows.

(1) All vertices of D start pebble-free.
(2) All vertices of D end pebble-free.
(3) Each vertex receives and loses a pebble at least once.
(4) A pebble can be placed on a vertex v only if all the predecessors of v have a pebble. Vertices with zero in-degree can be pebbled at any time.
(5) A pebble can be removed any time.

The rules for *black and white pebbling* are as follows.

(1) All vertices of D start pebble-free.
(2) All vertices of D end pebble-free.
(3) Each vertex receives and loses a pebble at least once.
(4) A white pebble can be placed on a vertex at any time.
(5) A white pebble on vertex v turns black at the moment all the predecessors of v are pebbled.
(6) A black pebble can be removed at any time, but no white pebble can be removed.

The principle application of pebbling is to model computations. We say that a vertex has been pebbled if it had a pebble placed on it at some point. A pebbling scheme is a sequence of placing and removing pebbles so that every vertex has been pebbled. The size of a pebbling scheme is the maximum number of pebbles deployed at any step. The *black pebble number* of an acyclic digraph D, denoted $\mathrm{bp}(D)$, is the smallest size of a black pebbling scheme. Similarly, the *black and white* *pebble number* of an acyclic digraph D, denoted $\mathrm{bwp}(D)$, is the smallest size of a black and white pebbling scheme. A black pebbling scheme, or black and white pebbling scheme, is said to be *monotonic* if each vertex is pebbled only one time. The *monotonic black pebble number*, or *monotonic black and white pebble number*, for an acyclic digraph D is denoted by $\mathrm{mbp}(D)$, or $\mathrm{mbwp}(D)$, respectively. The following relation between the node search number and monotonic black pebble number, or monotonic black and white pebble number, was proved in [97].

Theorem 4.17 ([97])**.** *Given a graph G, let $\Omega(G)$ denote the set of acyclic digraphs that are orientations of G. Let $\mathrm{mmbp}(G)$ be the minimum monotonic black pebble number over all digraphs in $\Omega(G)$, and $\mathrm{mmbwp}(X)$ be the minimum monotonic black and white pebble number over all digraphs in $\Omega(G)$. Then the following relationships hold.*

(1) $\mathrm{mmbp}(G) = \mathrm{ns}(G) = \mathrm{mmbwp(G)}$.
(2) *For all $D \in \Omega(G)$ with in-degree at most k,*

$$\mathrm{bwp}(D) \leq \mathrm{mbwp}(D) \leq (k+1)\mathrm{ns}(G).$$

5. Open problems on searching games

The area of graph searching games is relatively new, there are several open problems. We present a dozen of them, and note that more problems can be found in the references.

(1) The problem of determining the edge search number is **NP**-complete [108, 101]. This problem remains **NP**-complete for graphs with a maximum vertex degree of 3 [105]. However, whether the problem remains **NP**-complete for planar graphs is still unknown. In fact, the complexity of determining the search number of planar graphs in all search games mentioned in Section 2 is unknown.
(2) The problems of designing efficient polynomial-time approximation algorithms for computing the search number of all search games mentioned in Part 1 are wide open. There are only few results for special classes of graphs. For example, Bodlaender and Fomin [26] gave an $O(n \log n)$ time 2-approximation algorithm for computing the pathwidth (or node search number) of outerplanar graphs.
(3) Finding good lower bounds for search numbers is a challenge for all search games mentioned in this chapter. There are few results for lower bounds. For example, it is not clear how to improve the following lower bound given by Alspach et al. [8]

$$\mathrm{es}(G) \geq \max\{\delta(G) + g(G) - 2, \chi(G) - 1\},$$

where the minimum vertex degree $\delta(G) \geq 3$, the girth $g(G) \geq 3$, and the chromatic number $\chi(G) \geq 4$.

(4) Megiddo et al. [108] showed that the edge search problem is **NP**-hard. This problem belongs to **NP** by the monotonicity result of [101], in which LaPaugh showed that recontamination does not help to clear a graph. Using a reduction from the edge search problem, it can be easily shown that the connected edge search problem is also **NP**-hard. However, Yang et al. [149, 150] showed that the connected search is not monotonic. An interesting problem left open is whether the connected edge search problem belongs to **NP**.

(5) In [90], Johnson et al. gave an min-max theorem between directed treewidth and the number of searchers required to capture a robber in the strong-directed visible-robber game. However, the directed treewidth and the search number in this theorem are given by a constant factor between them. Whether there is a modified version of the directed treewidth that has an exact min-max theorem with the search number in the associated game remains an open problem.

(6) In [99], Kreutzer and Ordyniak gave examples showing that neither the directed visible-robber game associated with DAG-width nor the directed lazy-robber game associated with Kelly-width are monotone. It is not clear if the ratio of the monotonic search number to the search number is bounded by a constant in both games. That is, it is unknown if there is a constant c such that whenever k searchers have a winning strategy in one of the games then ck searchers have a monotone winning strategy.

(7) In [88], Hunter and Kreutzer conjectured that for a digraph, the search number in the directed visible-robber game and that in the directed lazy-robber game lie within constant factors of one another.

(8) Yang and Cao [145] proved that the internal strong and internal weak search problems are not monotonic. Although the internal strong search game is not monotonic, they still proved it is in **NP**. However, the problem of whether the internal weak search game is in **NP** is still open.

(9) In [145], Yang and Cao showed by examples that the ratio of the monotonic internal strong search number to the internal strong search number may be as large as $\Omega(\log n)$, where n is the number of vertices in the digraph. They conjecture that $O(\log n)$ is an upper bound of the ratio for all digraphs. Another open

problem is whether there is a constant c such that whenever k searchers have a winning strategy in the internal weak search game then ck searchers have a monotone winning strategy. We conjecture that this constant is less than 2 for all digraphs.

(10) In [147], Yang and Cao proved the monotonicity of the mixed weak searching problem using the crusade method proposed by Bienstock and Seymour [24]. But they applied the LaPaugh's method to prove the monotonicity of the weak searching problem. Since LaPaugh's method is more complicated than the crusade method, an interesting open problem is how to establish a relation between the mixed weak searching and the weak searching so that the monotonicity of the former implies the monotonicity of the latter.

(11) Many relations between different search numbers are described by inequalities, such as Theorems 2.1 and 3.16. The related open problems are to find the necessary and sufficient conditions for graphs or digraphs such that equalities hold. For example, for which graphs does the edge search number equal the node search number?

(12) Bodlaender and Kloks [27] gave a polynomial time algorithm for computing the pathwidth of a graph with constant treewidth. Since the edge search number of a graph equals the pathwidth of its 2-expansion, the edge search number of a graph with constant treewidth is polynomial time computable. However, the exponent in the running time of this algorithm is large. Even for a graph with treewidth two, it takes at least $\Omega(n^{11})$ time. The large exponent makes their algorithm impractical. Megiddo et al. [108] presented a linear time algorithm for computing the edge search number of trees, and Peng et al. [118] proposed a linear time algorithm to compute the optimal search strategy of trees. Yang et al. [152] presented a linear time algorithm for computing the edge search number of unicyclic graphs and some special cycle-disjoint graphs. The problem then, is how to design efficient algorithms for computing the search number of graphs with small treewidth, such as outerplanar graphs that have treewidth 2.

Part 2. Cops and Robbers

Cops and Robbers is one aspect of graph searching that has received much recent attention. In this two-player game of perfect information, a set of cops tries to capture a robber by moving at unit speed from vertex-to-vertex. More precisely, *Cops and Robbers* is a game played on a reflexive graph (that is, there is a loop at each vertex). There are two players consisting of a set of *cops* and a single *robber*. The game is played over a sequence of discrete time-steps or *rounds*, with the cops going first in round 0 and then playing alternate time-steps. The cops and robber occupy vertices; for simplicity, the player is identified with the vertex they occupy. The set of cops is referred to as C and the robber as R. When a player is ready to move in a round they must move to a neighbouring vertex. Because of the loops, players can *pass*, or remain on their own vertex. Observe that any subset of C may move in a given round.

The cops win if after some finite number of rounds, one of them can occupy the same vertex as the robber (in a reflexive graph, this is equivalent to the cop landing on the robber). This is called a *capture*. The robber wins if he can evade capture indefinitely. A *winning strategy for the cops* is a set of rules that if followed, result in a win for the cops. A *winning strategy for the robber* is defined analogously.

If a cop is placed at each vertex, then the cops are guaranteed to win. Therefore, the minimum number of cops required to win in a graph G is a well-defined positive integer, named the *cop number* (or *copnumber*) of the graph G. The notation $c(G)$ is used for the cop number of a graph G. If $c(G) = k$, then G is *k-cop-win*. In the special case $k = 1$, G is *cop-win* (or *copwin*).

The game of Cops and Robbers was first considered by Quilliot [121] in his doctoral thesis, and was independently considered by Nowakowski and Winkler [115]. Both [115, 121] refer only to one cop. The introduction of the cop number came in 1984 with Aigner and Fromme [2]. Many papers have now been written on cop number since these three early works; see the surveys [5, 81] and the recent book of Bonato and Nowakowski [37]. Cops and Robbers has even found recent application in artificial intelligence and so-called *moving target search*; see Isaza et al. [89] and Moldenhauer et al. [112].

Part 2 is a selective survey of recent results on Cops and Robbers, focusing on algorithmic and probabilistic results, as well as the game in infinite graphs. A much more exhaustive survey with proofs may be found in the book [37]. This part begins with the case of cop-win graphs in Section 6. Such graphs have a good characterization in terms of an elimination ordering of their vertices. In Section 7 a discussion of graphs with higher cop number is provided. The highlight here is Meyniel's conjecture, which gives an upper bound on the cop number of connected graphs. The cop number in graph classes is discussed in Section 8, algorithmic results in Section 9, random graphs in Section 10,

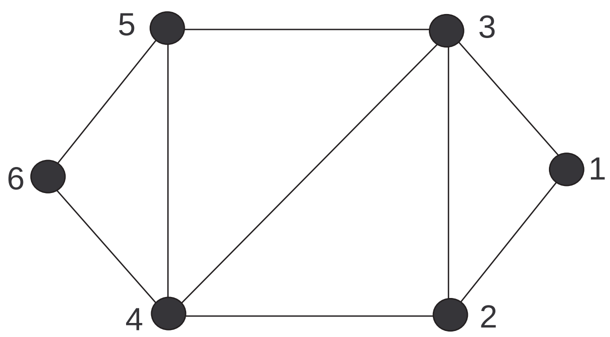

FIGURE 1. A cop-win ordering of a cop-win graph.

and infinite graphs in Section 11. Part 2 closes with twelve of the most important open problems in the field.

6. Cop-win graphs

The game of Cops and Robbers historically first considered only the case of one cop, and that is the focus in the present section. The *closed neighbor set* of a vertex x, written $N[x]$, is the set of vertices joined to x (including x itself). A vertex u is a *corner* if there is some vertex v such that $N[u] \subseteq N[v]$.

A graph is *dismantlable* if some sequence of deleting corners results in the graph K_1. For example, each tree is dismantlable: delete leaves repeatedly until a single vertex remains. The same approach holds with chordal graphs, which always contains at least two simplicial vertices (that is, vertices whose neighbor sets are cliques). The following result characterizes cop-win graphs.

Theorem 6.1 ([115]). *A graph is cop-win if and only if it is dismantlable.*

Cop-win (or dismantlable) graphs have a recursive structure, made explicit in the following sense. Observe that a graph is dismantlable if the vertices can be labeled by positive integers $[n] = \{1, 2, \ldots, n\}$, in such a way that for each $i < n$, the vertex i is a corner in the subgraph induced by $\{i, i+1, \ldots, n\}$. This ordering of $V(G)$ is called a *cop-win ordering*. See Figure 1 for a graph with vertices labeled by a cop-win ordering. Cop-win orderings are sometimes called *elimination orderings*, vertices from lower to higher index are deleted until only vertex n remains.

Let H be an induced subgraph of G formed by deleting one vertex. The graph H is a *retract* of G if there is a homomorphism f from G onto H so that $f(x) = x$ for $x \in V(H)$; that is, f is the identity on H. The map f is called a *retraction* (or 1-*point retraction* or *fold*). For example, the subgraph formed by deleting a vertex of degree 1 is a retract. If u is a corner, then the mapping

$$f(x) = \begin{cases} v & \text{if } x = u \\ x & \text{else} \end{cases}$$

is a retraction (recall that graphs are reflexive, so edges may map to a single vertex).

Retracts play an important role in characterizing cop-win graphs. The next theorem, due to Berarducci and Intrigila, shows that the cop number of a retract never increases.

Theorem 6.2 ([19])**.** *If H is a retract of G, then $c(H) \leq c(G)$.*

Cop-win orderings suggest a kind of linear structure to cop-win graphs; it roughly suggests that by "sweeping" from largest index vertex to smallest in the ordering, the robber can be captured. This intuition is made precise by the following winning strategy for the cops—first made explicit by Clarke and Nowakowski [50]—in a cop-win graph exploiting the cop-win ordering.

Cop-win (or No-backtrack) Strategy [50]. Assume that $[n]$ is a cop-win ordering of G, and for $1 \leq i \leq n$ define

$$G_i = G \restriction \{n, n-1, \ldots, i\}.$$

Note that $G_1 = G$ and G_n is just the vertex n. For each $1 \leq i \leq n-1$, let $f_i : G_i \to G_{i+1}$ be the retraction map from G_i to G_{i+1} mapping i onto a vertex that covers i in G_i. Define F_1 to be the identity mapping on $G : F_1(x) = x$ for all $x \in V(G)$. For $2 \leq i \leq n$ define

$$F_i = f_{i-1} \circ \cdots \circ f_2 \circ f_1.$$

In other words, the F_i is the mapping formed by iteratively retracting corners $1, 2, \ldots i-1$. As the f_i are homomorphisms, so are the F_i (recall that graphs are reflexive). Further, for all i, as the f_i are retractions, $F_i(x)$ and $F_{i+1}(x)$ are either equal or joined. If the robber is on vertex x in G, then think of $F_i(x)$ as the robber's *shadow* on G_i. See Figure 2.

The Cop-win Strategy is described as follows. The cop begins on G_n (the vertex n), which is the shadow of the robber's position under F_n (note that everything in G maps to n under F_n). Suppose that the robber is on u and the cop is occupying the shadow of the robber in

G_i equaling $F_i(u)$. If the robber moves to v, then the cop moves onto the image $F_{i-1}(v)$ of R in the larger graph G_{i-1}.

Theorem 6.3 ([50])**.** *The Cop-win Strategy results in a capture for the cop in at most n moves.*

If both players are playing *optimally* (that is, the cop is trying to minimize the length of the game, while the robber is trying to maximize it), then in cop-win graph, a cop can win in no more than $n-3$ moves if $n \geq 5$. See Bonato et al. [33].

Note that the dismantling characterization of Theorem 6.1 fails badly for infinite graphs. For example, an infinite one-way path (or *ray*) is dismantlable (if infinitely many vertex deletions are allowed), but fails to be cop-win. There is a characterization of cop-win graphs of any order (finite or infinite) included here, owing to Nowakowski and Winkler [115].

Define a relation $\preceq$ on vertices. The relation is defined recursively on ordinals, with $x \leq_0 x$ for all vertices u (in other words, $\leq_0$ is just the *diagonal* or *equality relation* on $V(G)$). Observer that $u \leq_\alpha v$ will mean that when a robber is on vertex u, a cop is on vertex v and it is the robber's turn to move, the robber will lose in at most α rounds. For an ordinal α, define $u \leq_\alpha v$ if and only if for each $a \in N[u]$ there exists a $b \in N[v]$ such that $a \leq_\beta b$ for some $\beta < \alpha$. Let ρ be the least ordinal such that $\leq_\rho = \leq_{\rho+1}$ and define $\preceq = \leq_\rho$.

Note that if $\rho < \alpha$, then the relation $\leq_\rho$ is a subset of $\leq_\alpha$. As such relations are bounded above in cardinality, the ordinal ρ exists. A binary relation on a set X is *trivial* if it equals the Cartesian product of X with itself, $X \times X$.

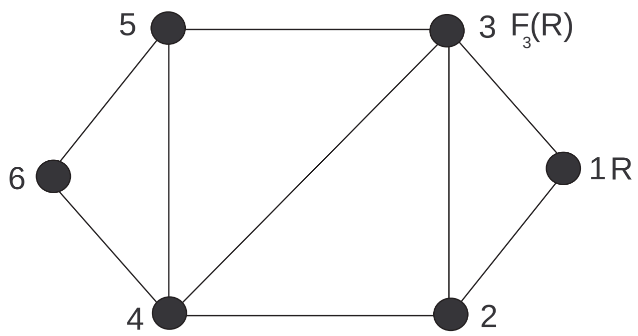

FIGURE 2. The robber and his shadow $F_3(R) = f_2 \circ f_1(R)$.

Theorem 6.4 ([115]). *A graph G is cop-win if and only if the relation $\preceq$ on $V(G)$ is trivial.*

A generalization of Theorem 6.4 has been found for graphs with higher cop number. Given a graph G, define the *kth strong power* of G, written $G^k_{\boxtimes}$, to have vertices the ordered k-tuples from G, with two tuples joined if they are equal or joined in each coordinate. See Figure 3 for an example. The positions of k-many cops in G are identified with a single vertex in $G^k_{\boxtimes}$. The definition of the strong power allows us to simulate movements of the cops in G by movements of a single cop in $G^k_{\boxtimes}$.

Let $P = G^k_{\boxtimes}$. For $i \in \mathbb{N}$, the relation $\leq_i$ on $V(G) \times V(P)$ is defined as follows by induction on i. For $x \in V(G)$ and $p \in V(P)$, $x \leq_0 p$ if in position p, at least one of the k cops is occupying x. For $i > 0$, $x \leq_i p$ if and only if for each $u \in N[x]$ there exists a $v \in N[p]$ such that $u \leq_j v$ for some $j < i$. Just as in the cop-win case, the relations $\leq_i$ are non-decreasing sets in i, and hence, there is an ordinal M such that $\leq_M = \leq_{M+1}$ and set $\preceq = \leq_M$. Although the notation $\preceq$ in this case clashes with the one for cop-win graphs, it is used again here for simplicity.

Theorem 6.5 ([49]). *A graph G is k-cop-win if and only if there exists $p \in V(P)$ such that $x \preceq p$ for every $x \in V(G)$.*

7. Higher cop number

Graphs with higher cop number are much less understood than cop-win graphs. Hence, the focus is on finding bounds on the cop number.

7.1. Upper bounds.

An elementary upper bound for the cop number is

$$c(G) \leq \gamma(G), \tag{7.1}$$

where $\gamma(G)$ is the domination number of G. In general graphs, the inequality (7.1) is far from tight (consider a path, for example).

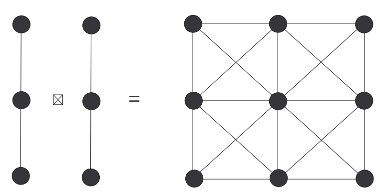

Figure 3. The strong power $(P_3)^2_{\boxtimes}$

We do not know how large the cop number of a connected graph can be as a function of its order. For a positive integer n, let $c(n)$ be the maximum value of $c(G)$, where G is of order n. *Meyniel's conjecture* states that

$$c(n) = O(\sqrt{n}).$$

The conjecture was mentioned in Frankl's paper [74] as a personal communication to him by Henri Meyniel in 1985 (see page 301 of [74] and reference [8] in that paper). Meyniel's conjecture stands out as one of the deepest (if not *the* deepest) problems on the cop number.

For many years, the best known upper bound for general graphs was the one proved by Frankl [74].

Theorem 7.1 ([74]). *If n is a positive integer, then*

$$c(n) \leq O\left(n\frac{\log\log n}{\log n}\right).$$

The key to proving Theorem 7.1 is the notion of cops guarding an isometric path. For a fixed integer $k \geq 1$, an induced subgraph H of G is *k-guardable* if, after finitely many moves, k cops can move only in the vertices of H in such a way that if the robber moves into H at round t, then he will be captured at round $t+1$. For example, a clique or a closed neighbor set (that is, a vertex along with its neighbors) in a graph are 1-guardable. Given a connected graph G, the *distance* between vertices u and v in G, is denoted $d_G(u,v)$. A path P in G is *isometric* if for all vertices u and v of P,

$$d_P(u,v) = d_G(u,v).$$

The following theorem of Aigner and Fromme [2] on guarding isometric paths has found a number of applications.

Theorem 7.2. [2] *An isometric path is* 1*-guardable.*

In 2008 Chinifooroshan [45] gave an improved upper bound once again using a guarding argument.

Theorem 7.3 ([45]). *For a positive integer n*

$$c(n) = O\left(\frac{n}{\log n}\right). \tag{7.2}$$

The bound (7.2), therefore, represents the first important step forward in proving Meyniel's conjecture in over 25 years. The key to proving (7.2) comes again from the notion of guarding an induced subgraph.

An improvement exists to the bound (7.2) in Theorem 7.3. The following theorem was proved independently by three sets of authors.

Theorem 7.4 ([75, 103, 130]). *For a positive integer n*

$$c(n) \leq O\left(\frac{n}{2^{(1-o(1))\sqrt{\log_2 n}}}\right). \tag{7.3}$$

The bound in (7.3) is currently the best upper bound for general graphs that is known, but it is still far from proving Meyniel's conjecture or even the soft version of the conjecture.

7.2. **Lower bounds.** A useful theorem of Aigner and Fromme [2] is the following. The girth of a graph is the length of minimum order cycle. The *minimum degree* of G is written $\delta(G)$.

Theorem 7.5 ([2]). *If G has girth at least 5, then $c(G) \geq \delta(G)$.*

Frankl [74] proved the following theorem generalizing Theorem 7.5 (which is the case $t = 1$).

Theorem 7.6 ([2]). *For a fixed integer $t \geq 1$, if G has girth at least $8t - 3$ and $\delta(G) > d$, then $c(G) > d^t$.*

Meyniel's conjecture states that the cop number is at most approximately $\sqrt{n}$. For graphs with large cop number near the conjectured bound, consider projective planes. A *projective plane* consists of a set of points and lines satisfying the following axioms.

(1) There is exactly one line incident with every pair of distinct points.
(2) There is exactly one point incident with every pair of distinct lines.
(3) There are four points such that no line is incident with more than two of them.

Finite projective planes have $q^2 + q + 1$ points for some integer $q > 0$ (called the *order* of the plane). For a given projective plane P, define $G(P)$ to be the bipartite graph with red vertices representing the points of P, and the blue vertices representing the lines. Vertices of different colors are joined if they are incident. This is the *incidence graph of P*. See Figure 4 for $G(P)$, where P is the Fano plane (that is, the projective plane of order 2). The incidence graph of the Fano plane is isomorphic to the famous *Heawood graph*.

Hence, Theorem 7.5 proves that $c(G(P)) \geq q + 1$. As proven in Prałat [119], $c(G(P)) = q + 1$. However, the orders of $G(P)$ depend on the orders of projective planes. The only orders where projective planes are known to exist are prime powers; indeed, this is a deep conjecture in finite geometry. What about integers which are not prime powers?

An infinite family of graphs $(G_n : n \geq 1)$ is *Meyniel extremal* if there is a constant d such that for sufficiently large n, $c(G_n) \geq d\sqrt{|V(G_n)|}$.

Recall the famous *Bertrand postulate.*

Theorem 7.7 ([44])**.** *For any integer $x > 1$, there is a prime in the interval $(x, 2x)$.*

In Prałat [119] a Meyniel extremal family was given using incidence graphs of projective planes and Theorem 7.7. Using Bertrand's postulate, it was shown that

$$c(n) \geq \sqrt{\frac{n}{8}}$$

for $n \geq 72$. Using this theorem and a result from number theory, it was shown in Prałat [119] that for sufficiently large n,

$$c(n) \geq \sqrt{\frac{n}{2}} - n^{0.2625}. \tag{7.4}$$

Define m_k to be the minimum order of a connected graph G satisfying $c(G) \geq k$. Define M_k to be the minimum order of a connected k-cop-win graph. It is evident that the m_k are monotonically increasing, and $m_k \leq M_k$. A recent work [14] establishes that

$$m_3 = 10.$$

The fact that $m_3 \geq 10$ follows by a computer search. The upper bound follows by considering the Petersen graph, which is 3-cop-win. In fact,

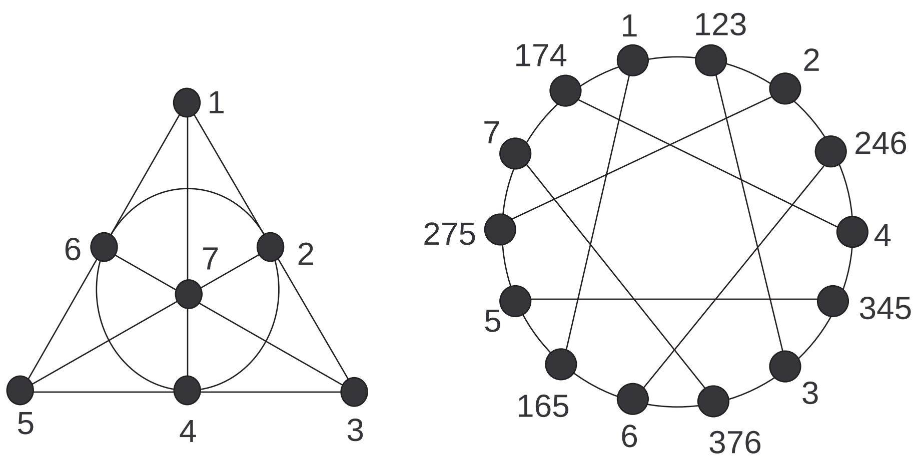

Figure 4. The Fano plane and its incidence graph. Lines are represented by triples.

a computer search in Baird and Bonato [14] shows the surprising fact that the Petersen graph is the *unique* smallest order isomorphism type of connected graph with cop number 3.

Define $f_k(n)$ to be the number of non-isomorphic connected k-cop-win graphs of order n (that is, the *unlabelled* graphs G of order n with $c(G) = k$). Define $g(n)$ to be the number of non-isomorphic (not necessarily connected) graphs of order n, and $g_c(n)$ the number of non-isomorphic connected graphs of order n. Trivially, for all k, $f_k(n) \leq g(n)$. The table below presents the values of g, g_c, f_1, and f_2 for small orders. The values of g and g_c come from [133], f_1 was computed by checking for cop-win orderings [115], while f_2 and f_3 were computed using Algorithm 1 given in Section 9.

order n	$g(n)$	$g_c(n)$	$f_1(n)$	$f_2(n)$	$f_3(n)$
1	1	1	1	0	0
2	2	1	1	0	0
3	4	2	2	0	0
4	11	6	5	1	0
5	34	21	16	5	0
6	156	112	68	44	0
7	1044	853	403	450	0
8	12346	11117	3791	7326	0
9	274668	261080	65561	195519	0
10	12005168	11716571	2258313	9458257	1

The next theorem sets up an unexpected connection between Meyniel's conjecture and the order of m_k.

Theorem 7.8 ([14]). (1) $m_k = O(k^2)$.

(2) *Meyniel's conjecture is equivalent to the property that*

$$m_k = \Omega(k^2).$$

Hence, if Meyniel's conjecture holds, then Theorem 7.8 implies that $m_k = \Theta(k^2)$.

8. Graph classes

Planar graphs have inspired some of the deepest results in graph theory, most notably the Four Color Theorem (which states that every planar graph has chromatic number at most four; see Appel, Haken, and Koch [12]). A graph is *planar* if it can be drawn in $\mathbb{R}^2$ without any two of its edges crossing. Aigner and Fromme [2] showed in fact that planar graphs require no more than three cops.

Theorem 8.1 ([2]). *If G is a planar graph, then $c(G) \leq 3$.*

The idea of the proof of Theorem 8.1 is to increase the *cop territory*; that is, vertices that if the robber moved to he would be caught. Hence, the number of vertices the robber can move to without being caught is eventually is reduced to the empty set, and so the robber is captured.

For a fixed graph H, Andreae [9] generalized this result by proving that the cop number of a K_5-minor-free graph (or $K_{3,3}$-minor-free graph) is at most 3 (recall that planar graphs are exactly those which are K_5-minor-free and $K_{3,3}$-minor-free). Andreae [10] also proved that for any graph H the cop number of the class of H-minor-free graphs is bounded above by a constant.

Less is known about the cop number of graphs with positive genus. As such, the survey of such graphs is brief. The main conjecture in this area is due to Schroeder. In [129], Schroeder conjectured that if G is a graph of genus g, then $c(G) \leq g + 3$. Quilliot [123] had shown the following.

Theorem 8.2 ([123]). *If G is a graph of genus g, then $c(G) \leq 2g + 3$.*

In the same paper as the conjecture, Schroeder showed the following.

Theorem 8.3 ([129]). *If G is a graph of genus g, then*

$$c(G) \leq \left\lfloor \frac{3g}{2} \right\rfloor + 3.$$

Schroeder also proved the following theorem.

Theorem 8.4 ([129]). *If G is a graph that can be embedded on a torus, then $c(G) \leq 4$.*

A graph G is *outerplanar* if it has an embedding in the plane with the following properties.

(1) Every vertex lies on a circle.
(2) Every edge of G either joins two consecutive vertices around the circle or is a chord across the circle.
(3) If two chords intersect, then they do so at a vertex.

Clarke proved the next result in her doctoral thesis.

Theorem 8.5 ([48]). *If G be an outerplanar graph, then $c(G) \leq 2$.*

Theorem 8.5 was generalized by Theis [142] to series-parallel graphs.

Lu and Peng [103] showed that Meyniel's conjecture holds in the class of graphs with diameter two. The proof uses the notion of guarding subgraph, but also uses a randomized argument.

Theorem 8.6 ([103]). *If G is a graph on n vertices with diameter two, then*

$$c(G) \leq 2\sqrt{n} - 1. \tag{8.1}$$

The same bound (8.1) was also shown in [103] in the case when G is bipartite and of diameter at most three. The incidence graphs of projective planes are bipartite of diameter three, and so show that the bound (8.1) is asymptotically tight in that class.

9. Algorithmic results

Another approach to the Cops and Robbers game is an algorithmic one. Consider the following two graph decision problems.

k-COP NUMBER: Given a graph G and a positive integer k, is $c(G) \leq k$?

k-FIXED COP NUMBER: Given a graph G and a fixed positive integer k, is $c(G) \leq k$?

The main difference between the two problems is that in k-COP NUMBER the integer k may be a function of n, and so grows with n. In k-FIXED COP NUMBER, k is fixed and not part of the input, and so is independent of n.

There is the following result.

Theorem 9.1 ([19, 83]). *The problem FIXED COP NUMBER is in* $\boldsymbol{P}$.

An algorithm from Bonato et al. [31] is described which may be used to prove Theorem 9.1. Given a graph G, recall that the kth *strong power of* G, written $G^k_{\boxtimes}$, is the strong product of G with itself k times. For a set X, define 2^X to be the set of subsets of X. For $S \subseteq V(G)$, define $N_G[S]$ to be the union of the closed neighbor sets of vertices in S.

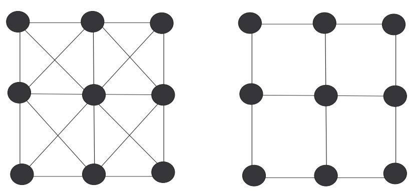

FIGURE 5. The graph on the left is a cop-win non-outerplanar graph, while the graph on the right is non-outerplanar with cop number two.

Algorithm 1 CHECK K COP NUMBER

Require: $G = (V, E)$, $k \geq 1$
initialize $f(u)$ to $V(G) \setminus N_G[u]$, for all $u \in V(G^k_{\boxtimes})$
repeat
  for all $uv \in E(G^k_{\boxtimes})$ **do**
    $f(u) \leftarrow f(u) \cap N_G[f(v)]$
    $f(v) \leftarrow f(v) \cap N_G[f(u)]$
  end for
until the value of f is unchanged
if there exists $u \in V(G^k_{\boxtimes})$ such that $f(u) = \emptyset$ **then**
  $c(G) \leq k$
else
  $c(G) > k$
end if

Theorem 9.2 ([31]). *Suppose $k \geq 1$ is an integer. Then $c(G) > k$ if and only if there is a mapping $f : V(G^k_{\boxtimes}) \to 2^{V(G)}$ with the following properties.*

(1) *For every $u \in V(G^k_{\boxtimes})$,*
$$\emptyset \neq f(u) \subseteq V(G) \setminus N_G[u].$$
(2) *For every $uv \in E(G^k_{\boxtimes})$,*
$$f(u) \subseteq N_G[f(v)].$$

Consider Algorithm 1 for determining whether $c(G) \leq k$ based on Theorem 9.2. The following theorem gives Theorem 9.1 as a corollary.

Theorem 9.3 ([31]). *Algorithm 1 runs in time $O(n^{3k+3})$.*

If k is not fixed (and hence, can be a function of n), then the problem becomes less tractable.

Theorem 9.4 ([73]). *The problem k-COP NUMBER is **NP**-hard.*

Theorem 9.4 does not say that k-COP NUMBER is in **NP**; that is an open problem! See Section 12 below. Theorem 9.4 is proved in Fomin et al. [73] by using a reduction from the following **NP**-complete problem:

DOMINATION: Given a graph G and an integer $k \geq 2$, is $\gamma(G) \leq k$?

10. RANDOM GRAPHS

Random graphs are a central topic in graph theory; however, only recently have researchers considered the cop number of random graphs.

Define a probability space on graphs of a given order $n \geq 1$ as follows. Fix a vertex set V consisting of n distinct elements, usually taken as $[n]$, and fix $p \in [0, 1]$. Define the space of *random graphs of order n with edge probability p*, written $G(n, p)$ with sample space equaling the set of all $2^{\binom{n}{2}}$ (labeled) graphs with vertex set V, and

$$\mathbb{P}(G) = p^{|E(G)|}(1-p)^{\binom{n}{2}-|E(G)|}.$$

Informally, $G(n, p)$ may be viewed as the space of graphs with vertex set V, so that two distinct vertices are joined independently with probability p. Even more informally: toss a (biased) coin to determine the edges of your graph. Hence, V does not change, but the number of edges is not fixed: it varies according to a binomial distribution with expectation $\binom{n}{2}p$. Despite the fact that $G(n, p)$ is a space of graphs, it is often called *the random graph of order n with edge probability p*. Random graphs were introduced in a series of papers by Erdős and Rényi [61, 59, 60]. See the book [28] for a modern reference.

The cases when p is fixed are considered, and when it is a function of n. Graph parameters, such as the cop number, become random variables in $G(n, p)$. For notational ease, the cop number of $G(n, p)$ is referred to simply by $c(G(n, p))$.

An event holds *asymptotically almost surely* (or *a.a.s.* for short) if it holds with probability tending to 1 as $n \to \infty$. For example, if p is constant, then a.a.s. $G(n, p)$ is diameter two and not planar.

10.1. **Constant p.** The cop number of $G(n, p)$ was studied in Bonato et al. [35] for constant p, where the following result was proved. For $p \in (0, 1)$ or $p = p(n) = o(1)$, define

$$\mathbb{L}n = \log_{\frac{1}{1-p}} n.$$

Theorem 10.1 ([35]). *Let $0 < p < 1$ be fixed. For every real $\varepsilon > 0$, a.a.s.*

$$(1-\varepsilon)\mathbb{L}n \leq c(G(n, p)) \leq (1+\varepsilon)\mathbb{L}n. \tag{10.1}$$

In particular,

$$c(G(n, p)) = \Theta(\log n).$$

The upper bound in Theorem 10.1 follows by considering the domination number of $G(n, p)$ [56], while the lower bounds follows by considering an adjacency property. If the case of $p = 1/2$ is considered, then $G(n, p)$ corresponds to the case of uniformly choosing a labeled graph of order n from the space of all such graphs. Hence, Theorem 10.1 may be interpreted as saying "most" finite graphs of order n have cop number approximately $\log n$.

Properties of randomly chosen k-cop-win graphs (with the uniform distribution) are described next. For this, it is equivalent to work in the probability space $G(n, 1/2)$. Let **cop-win** be the event in $G(n, 1/2)$ that the graph is cop-win and let **universal** be the event that there is a universal vertex. If a graph has a universal vertex w, then it is cop-win; in a certain sense, graphs with universal vertices are the simplest cop-win graphs. The probability that a random graph is cop-win can be estimated as follows:

$$\begin{aligned}\mathbb{P}(\textbf{cop-win}) \geq \mathbb{P}(\textbf{universal}) &= n2^{-n+1} - O(n^2 2^{-2n+3}) \\ &= (1+o(1))n2^{-n+1}.\end{aligned}$$

A recent surprising result of Bonato et al. [36] showed this lower bound is in fact the correct asymptotic value for $\mathbb{P}(\textbf{cop-win})$.

Theorem 10.2 ([36])**.** *In $G(n, 1/2)$,*

$$\mathbb{P}(\textbf{cop-win}) = (1+o(1))n2^{-n+1}.$$

Hence, almost all cop-win graphs contain a universal vertex.

10.2. **The dense case.** Now consider the cop number of *dense* random graphs, with average degree pn at least $\sqrt{n}$. The main results in this case were given in Bonato et al. [38].

Theorem 10.3 ([38])**.** (1) *Suppose that $p \geq p_0$ where p_0 is the smallest p for which*

$$p^2/40 \geq \frac{\log\big((\log^2 n)/p\big)}{\log n}$$

holds. Then a.a.s.

$$\mathbb{L}n - \mathbb{L}\big((p^{-1}\mathbb{L}n)(\log n)\big) \leq c(G(n,p)) \leq \mathbb{L}n - \mathbb{L}\big((\mathbb{L}n)(\log n)\big) + 2.$$

(2) *If $(2\log n)/\sqrt{n} \leq p = o(1)$ and $\omega(n)$ is any function tending to infinity, then a.a.s.*

$$\mathbb{L}n - \mathbb{L}\big((p^{-1}\mathbb{L}n)(\log n)\big) \leq c(G(n,p)) \leq \mathbb{L}n + \mathbb{L}(\omega(n)).$$

By Theorem 10.3, the following corollary gives a concentration result for the cop number. In particular, for a wide range of p, the cop number of $G(n, p)$ concentrates on just the one value $\mathbb{L}n$.

Corollary 10.4 ([38])**.** *If $p = n^{-o(1)}$ and $p < 1$, then a.a.s.*

$$c(G(n,p)) = (1+o(1))\mathbb{L}n.$$

From Theorem 10.3 part (1) it follows that if p is a constant, then there is the concentration result that

$$c(G(n,p)) = \mathbb{L}n - 2\mathbb{L}\log n + \Theta(1) = (1+o(1))\mathbb{L}n.$$

10.3. **The sparse case and Zig-Zag theorem.** Bollobás, Kun, and Leader [29] established the following bounds on the cop number in the sparse case, when the expected degree is $np = O(n^{1/2})$.

Theorem 10.5 ([29])**.** *If* $p(n) \geq 2.1 \log n/n$, *then a.a.s.*

$$\frac{1}{(np)^2} n^{\frac{1}{2} \frac{\log\log(np)-9}{\log\log(np)}} \leq c(G(n,p)) \leq 160000\sqrt{n}\log n\,. \tag{10.2}$$

In particular, Theorem 10.5 proves Meyniel's conjecture for random graphs, up to a logarithmic factor of n from the upper bound in (10.2). Recent work by Prałat and Wormald [120] removes the $\log n$ factor in the upper bound of (10.2) and hence, proves the Meyniel bound for random graphs.

It would be natural to assume that the cop number of $G(n,p)$ is close to $\sqrt{n}$ also for $np = n^{\alpha+o(1)}$, where $0 < \alpha < 1/2$. The so-called "Zig-Zag Theorem" of Łuczak and Prałat [104] demonstrated that the actual behaviour of $c(G(n,p))$ is much more complicated.

Theorem 10.6 (Zig-Zag Theorem, [104])**.** *Let* $0 < \alpha < 1$, *and* $d = d(n) = np = n^{\alpha+o(1)}$.

(1) *If* $\frac{1}{2j+1} < \alpha < \frac{1}{2j}$ *for some* $j \geq 1$, *then a.a.s.*

$$c(G(n,p)) = \Theta(d^j)\,.$$

(2) *If* $\frac{1}{2j} < \alpha < \frac{1}{2j-1}$ *for some* $j \geq 1$, *then a.a.s.*

$$\Omega\left(\frac{n}{d^j}\right) = c(G(n,p)) = O\left(\frac{n\log n}{d^j}\right)\,.$$

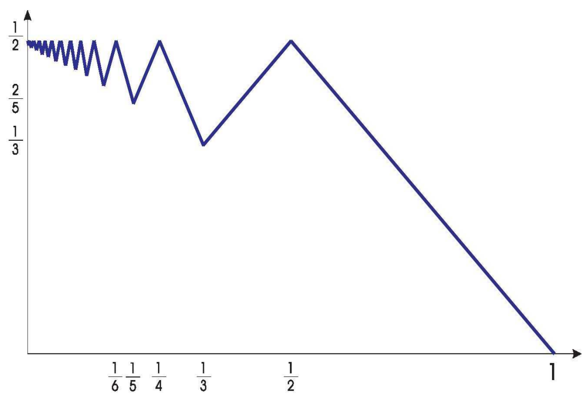

FIGURE 6. The zig-zag shaped graph of the cop number of $G(n,p)$.

Consider the function $f : (0, 1) \to \mathbb{R}$ defined by

$$f(x) = \frac{\log \overline{c}(G(n, n^{x-1}))}{\log n},$$

where $\overline{c}(G(n, n^{x-1}))$ is the median of the cop number for $G(n, p)$. See Figure 6, which justifies the Theorem 10.6's moniker. In particular,

$$f(x) = \begin{cases} \alpha j & \text{if } \frac{1}{2j+1} \leq \alpha < \frac{1}{2j} \text{ for some } j \geq 1; \\ 1 - \alpha j & \text{if } \frac{1}{2j} \leq \alpha < \frac{1}{2j-1} \text{ for some } j \geq 1. \end{cases}$$

11. Infinite graphs

Infinite graphs exhibit properties often quite different than finite ones. In this regard, the cop number is no exception. For example, the *ray* (that is, the one-way infinite path) is an infinite tree with infinite cop number: one robber can always stay ahead of finitely many cops.

11.1. **Cop density.**

When dealing with countable graphs, note that they are limits of chains of finite graphs. To analyze the cop number of infinite graphs, consider the *cop density* of a finite graph first introduced in [35]. Define

$$D_c(G) = \frac{c(G)}{|V(G)|}.$$

Note that $D_c(G)$ is a rational number in $[0, 1]$. Every countable graph G is the limit of a chain of finite graphs, and there are infinitely many distinct chains with limit G. Suppose that $G = \lim_{n\to\infty} G_n$, where $\mathcal{C} = (G_n : n \in \mathbb{N})$ is a fixed chain of induced subgraphs of G. The chain $\mathcal{C}$ is a *full chain for* G. Define

$$D(G, \mathcal{C}) = \lim_{n\to\infty} D_c(G_n),$$

if the limit exists (and then it is a real number in $[0, 1]$). This is the *cop density of* G *relative to* $\mathcal{C}$; if $\mathcal{C}$ is clear from context, this is referred to as the *cop density of* G.

The *upper cop density of* G, written $UD(G)$, is defined as

$$\sup\{D(G, \mathcal{C}) : \mathcal{C} \text{ is a full chain for } G\}.$$

Note that $UD(G)$ does not depend on the chain, and is a parameter of G.

For a positive integer n, a graph G *is strongly* n*-e.c.* if for all disjoint finite sets of vertices A and B from G with $|A| \leq n$, there is a vertex z correctly joined to A and B. Note that the infinite random graph is the unique isomorphism type of countable graph which is strongly n-e.c. for all n; see [41]. The following result, proved in Bonato et al. [35], finds connections between infinite-cop-win graphs and the strongly 0- and 1-e.c. properties.

Theorem 11.1 ([35])**.** (1) *If G is strongly 1-e.c., then $c(G)$ is infinite.*
(2) *If $c(G)$ is infinite, then G satisfies the strongly 0-e.c. property.*

If G is strongly 1-e.c., then the cop density of G may be *any* real number in $[0,1]$. This property applies, therefore, to a large number of graphs: for each $n \geq 0$, there are $2^{\aleph_0}$ many non-isomorphic countable graphs that are strongly n-e.c. but not strongly $(n+1)$-e.c.; see Theorem 4.1 of [32].

Theorem 11.2 ([35])**.** *If G is strongly 1-e.c., then for all $r \in [0,1]$, there is a chain $\mathcal{C}$ in G such that $D(G,\mathcal{C}) = r$.*

The next result completely characterizes the upper cop density of a graph G: $UD(G)$ takes on one of the two values 0 or 1, and equals 1 exactly when G is strongly 0-e.c. This fact, proven in [35], is somewhat unexpected.

Theorem 11.3 ([35])**.** *The following are equivalent.*
(1) $UD(G) = 1$.
(2) $UD(G) > 0$.
(3) *G is strongly 0-e.c.*

We note that the strongly 0-e.c. graphs are precisely the spanning subgraphs of the infinite random graph; see Cameron [41].

11.2. **Chordal graphs.** As another instance where results from finite graphs do not translate to infinite ones, consider chordal graphs. The graph G is *chordal* if each cycle of length at least four has a chord. A vertex of G is *simplicial* if its neighborhood induces a complete graph. Every finite chordal graph contains at least two simplicial vertices, and the deletion of a simplicial vertex leaves a chordal graph. As a simplicial vertex is a corner, a finite chordal graph is dismantlable and so cop-win.

However, an infinite tree containing a ray is chordal but not cop-win. Such trees have infinite diameter. Inspired by a question of Martin Farber which asked if infinite chordal graphs (more generally, bridged graphs) of finite diameter are cop-win, it was shown in Hahn et al. [82] that there exist infinite chordal graphs of diameter two that are not cop-win. The difficulty lies in finding examples with finite diameter.

Theorem 11.4 ([82])**.** *For each infinite cardinal κ, there exist chordal, robber-win graphs of order κ with diameter two.*

11.3. **Large families of cop-win graphs.** From Theorem 11.4, the cop number of infinite graphs behaves rather differently than in the finite case. Vertex-transitive cop-win finite graphs are cliques; see Nowakowski and Winkler [115]. However, this fails badly in the infinite case, which is the focus of this section.

A class of graphs is *large* if for each infinite cardinal κ there are 2^κ many non-isomorphic graphs of order κ in the class. In other words, a large class contains as many as possible non-isomorphic graphs of each infinite cardinality. For example, the classes of all graphs, all trees, and all k-chromatic graphs for k finite are large. Recall that a graph G is *vertex-transitive* if for each pair of vertices x and y there is an automorphism of G mapping x to y. The following result of Bonato et al. [34] showed that for any integer $k > 0$ there are large families of k-cop-win graphs that are vertex-transitive.

Theorem 11.5 ([34]). *The class of cop-win, vertex-transitive graphs with the property that the cop can win in two moves is large.*

To describe the large family in Theorem 11.5, recall some properties of the strong product of a set of graphs over a possibly infinite index set. Let I be an index set. The *strong product* of a set $\{G_i : i \in I\}$ of graphs is the graph $\boxtimes_{i\in I} G_i$ defined by

$$\begin{aligned} V(\boxtimes_{i\in I} G_i) &= \{f : I \to \bigcup_{i\in I} V(G_i) : f(i) \in V(G_i) \text{ for all } i \in I\}, \\ E(\boxtimes_{i\in I} G_i) &= \{fg :\ f \neq g \text{ and for all } i \in I, \\ f(i) &= g(i) \text{ or } f(i)g(i) \in E(G_i)\}. \end{aligned}$$

products exhibit unusual properties if there are infinitely many factors.

Fix a vertex $f \in V(\boxtimes_{i\in I} G_i)$ and define the *weak strong product* of $\{G_i : i \in I\}$ *with base* f as the subgraph $\boxtimes_f^I G_i$ of $\boxtimes_{i\in I} G_i$ induced by the set of all $g \in V(\boxtimes_{i\in I} G_i)$ such that $\{i \in I : g(i) \neq f(i)\}$ is finite. The graph $\boxtimes_f^I G_i$ is connected if each factor is, and if $|I| \leq \kappa$ and $|V(G_i)| \leq \kappa$ for each $i \in I$, then $|V(\boxtimes_f^I G_i)| \leq \kappa$. For $i \in I$, the *projection mapping* $\pi_i : \boxtimes_f^I G_i \to G_i$ is defined by $\pi_i(g) = g(i)$.

Let $\{G_i : i \in I\}$ be a set of isomorphic copies of G. Denote by $\boxtimes^I G$ the strong product $\boxtimes_{i\in I} G_i$. If $f \in V(\boxtimes^I G)$ is fixed, denote by G_f^I the weak strong product $\boxtimes_f^I G$ with base f. One particular power of a graph is of special interest as it allows us to construct vertex-transitive graphs out of non-transitive ones. Let κ be a cardinal, and let H be a graph of order κ. Let $I = \kappa \times V(H)$ and define $f : I \to V(H)$ by $f(\beta, v) = v$. The power H_f^I of H with base f will be called the *canonical power* of H and will be denoted by H^H.

As automorphisms of the factors applied coordinate-wise yield an automorphism of the product, it follows that the weak strong product of vertex-transitive graphs is vertex-transitive. However, the following technical lemma from [34] demonstrates the paradoxical property that if there are infinitely many factors, the weak strong product may be vertex-transitive even if none of the factors are!

Lemma 11.6 ([34])**.** *If H is an infinite graph, then the canonical power of H^H is vertex-transitive.*

12. A dozen problems on Cops and Robbers

To serve as a record and a challenge, twelve open problems on Cops and Robbers are stated. These are arguably the most central (and relevant) problems in the field at the moment. Solutions to problems (1) and (11), in particular, would be exceptional breakthroughs.

(1) Most likely the deepest open problem in the area is to settle *Meyniel's conjecture*: If G is a graph of order n, then
$$c(G) = O(\sqrt{n}). \tag{12.1}$$
(2) An easier problem than (1) would to settle the so-called *soft Meyniel's conjecture*: For a fixed constant $\epsilon > 0$,
$$c(n) = O(n^{1-\epsilon}),$$
(3) Meyniel's conjecture remains open for familiar graph classes. For example, does it hold in graphs whose chromatic number is bounded by some constant k?
(4) While the parameters m_k are non-increasing, an open problem is to determine whether the M_k are in fact non-increasing. A possibly more difficult problem is to settle whether $m_k = M_k$ for all $k \geq 1$.
(5) Can the lower bound (7.4)
$$c(n) \geq \sqrt{\frac{n}{2}} - n^{0.2625},$$
be improved for sufficiently large n?
(6) For $k > 1$, it is conjectured in [36] that almost all k-cop-win graphs in fact have a dominating set of cardinality k. If this is the case, then the number labeled k-cop-win graphs of order n satisfies
$$F_k(n) = 2^{o(n)} \left(2^k - 1\right)^{n-k} 2^{\binom{n-k}{2}}.$$
(7) Characterize the planar graphs with cop number 1, 2, and 3.

(8) Determine the cop number of the giant component of $G(n, p)$, where $p = \Theta(1/n)$. In particular, show that the cop number is a.a.s. $O(\sqrt{n})$.
(9) *Schroeder's conjecture*: If G is a graph of genus g, then $c(G) \leq g + 3$.
(10) Is the decision problem k-COP NUMBER in **NP**?
(11) Is COP NUMBER **EXPTIME**-complete? Goldstein and Reingold [80] proved that the version of the Cops and Robbers game on directed graphs is **EXPTIME**-complete. They also proved that the version of the game on undirected graphs when the cops and the robber are given their initial positions is also **EXPTIME**-complete.
(12) Are there large classes of infinite cop-win graphs whose members are k-chromatic, where $k \geq 2$ is an integer?

13. Conclusion

The chapter focused on two major aspects of graph searching: *searching* games and *Cops and Robbers* games. A broad overview of searching was given for both graphs and digraphs, focusing on the cases when the robber is invisible and active, or visible and lazy. Related games such as minimum cost searching and pebbling were also considered. Structural characterizations were given in these cases, along with complexity bounds on various searching parameters In the game of Cops and Robbers, a summary of known bounds and techniques were presented, focusing on Meyniel's conjecture as one of the most important problems on the cop number. Results were presented on algorithmic and probabilistic aspects of the cop number, and results were considered on infinite graphs. For both searching and Cops and Robbers, several problems and conjectures were given.

Given the broad diversity of techniques, applications, and problems in graph searching, it is evident that the field is now occupying an increasingly central position in graph theory and theoretical computer science. The subject has seen an explosive growth of interest in recent years. The wealth of new research on graph searching points to many more years of fruitful research in the field.

Recommended Reading

[1] I. Adler, Directed tree-width examples, *Journal of Combinatorial Theory, Series B*, 97:718–725, 2007.
[2] M. Aigner, M. Fromme, A game of cops and robbers, *Discrete Applied Mathematics*, 8:1–12, 1984.

[3] N. Alon, P. Pralat, R. Wormald, Cleaning regular graphs with brushes, *SIAM Journal on Discrete Mathematics*, 23:233–250, 2008.

[4] N. Alon, J. Spencer, *The Probabilistic Method*, Wiley, New York, 2000.

[5] B. Alspach, Sweeping and searching in graphs: a brief survey, *Matematiche*, 5:95–37,2006.

[6] B. Alspach, D. Dyer, D. Hanson, B. Yang, Time constrained Searching, *Theoretical Computer Science*, 399:158–168, 2008.

[7] B. Alspach, D. Dyer, D. Hanson, B. Yang, Arc searching digraphs without jumping, In: *Proceedings of the 1st International Conference on Combinatorial Optimization and Applications (COCOA'07)*, 2007.

[8] B. Alspach, D. Dyer, D. Hanson, B. Yang, Lower bounds on edge searching, In: *Proceedings of the 1st International Symposium on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies (ESCAPE'07)*, Lecture Notes in Computer Science, Vol. 4614, 516–527, 2007.

[9] T. Andreae, Note on a pursuit game played on graphs, *Discrete Applied Mathematics*, 9:111–115, 1984.

[10] T. Andreae, On a pursuit game played on graphs for which a minor is excluded, *Journal of Combinatorial Theory, Series B*, 41:37–47, 1986.

[11] R.P. Anstee, M. Farber, On bridged graphs and cop-win graphs, *Journal of Combinatorial Theory, Series B*, 44:22–28, 1988.

[12] K. Appel, W. Haken, J. Koch, Every planar map is four colorable, *Illinois Journal of Mathematics*, 21:439–567, 1977.

[13] S. Arnborg, D. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM Journal on Algebraic Discrete Methods*, 8:277–284, 1987.

[14] W. Baird, A. Bonato, Meyniel's conjecture on the cop number: a survey, Preprint 2011.

[15] J. Barat, Directed path-width and monotonicity in digraph searching, *Graphs and Combinatorics*, 22:161–172, 2005.

[16] L. Barrière, P. Flocchini, P. Fraigniaud, N. Santoro, Capture of an intruder by mobile agents, In: *Proceedings of the 14th annual ACM symposium on Parallel algorithms and architectures (SPAA'02)*, 2002.

[17] L. Barrière, P. Fraigniaud, N. Santoro, D. Thilikos, Searching is not jumping, In: *Proceedings of the 29th Workshop on Graph Theoretic Concepts in Computer Science (WG'03)*, 2003.

[18] M. Bender, A. Fernández, D. Ron, A. Sahai, and S. Vadhan, The power of a pebble: exploring and mapping directed graphs, In: *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC'98)*, 1998.

[19] A. Berarducci, B. Intrigila, On the cop number of a graph, *Advances in Applied Mathematics*, 14:389–403, 1993.

[20] D. Berwanger, A. Dawar, P. Hunter, S. Kreutzer, DAG-width and parity games, In: *Proceedings of the 23rd Symposium of Theoretical Aspects of Computer Science (STACS'06)*, 2006.

[21] D. Berwanger, E. Grädel, Entanglement – a measure for the complexity of directed graphs with applications to logic and games, In: *Proceedings of the 11th International Conference on Logic for Programming, Artificial Intelligence*, 2004.

[22] T. Biedl, T. Chan, Y. Ganjali, M. T. Hajiaghayi, D. Wood, Balanced vertex-ordering of graphs, *Discrete Applied Mathematics*, 148:27–48, 2005.

[23] D. Bienstock, Graph searching, path-width, tree-width and related problems (a survey), *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, 5:33–49, 1991.

[24] D. Bienstock, P. Seymour, Monotonicity in graph searching, *Journal of Algorithms*, 12:239–245, 1991.

[25] H.L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, *Theoretical Computer Science*, 209: 1-45, 1998.

[26] H. Bodlaender, F. Fomin, Approximation of pathwidth of outerplanar graphs. *Journal of Algorithms*, 43: 190–200, 2002.

[27] H. Bodlaender, T. Kloks, Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21: 358–402, 1996.

[28] B. Bollobás, *Random graphs, Second edition,* Cambridge Studies in Advanced Mathematics **73** Cambridge University Press, Cambridge, 2001.

[29] B. Bollobás, G. Kun, I. Leader, Cops and robbers in a random graph, Preprint 2011.

[30] A. Bonato, *A Course on the Web Graph*, American Mathematical Society Graduate Studies Series in Mathematics, Providence, Rhode Island, 2008.

[31] A. Bonato, E. Chiniforooshan, P. Prałat, Cops and Robbers from a distance, *Theoretical Computer Science* **411** (2010) 3834–3844.

[32] A. Bonato, D. Delić, On a problem of Cameron's on inexhaustible graphs, *Combinatorica*, 24:35–51, 2004.

[33] A. Bonato, G. Hahn, P.A. Golovach, J. Kratochvíl, The capture time of a graph, *Discrete Mathematics*, 309:5588–5595, 2009.

[34] A. Bonato, G. Hahn, C. Tardif, Large classes of infinite k-cop-win graphs, *Journal of Graph Theory*, 65:334–242, 2010.

[35] A. Bonato, G. Hahn, C. Wang, The cop density of a graph, *Contributions to Discrete Mathematics*, 2:133–144, 2007.

[36] A. Bonato, G. Kemkes, P. Prałat, Almost all cop-win graphs contain a universal vertex, Preprint 2011.

[37] A. Bonato, R.J. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, Providence, Rhode Island, 2011.

[38] A. Bonato, P. Prałat, C. Wang, Pursuit-evasion in models of complex networks, *Internet Mathematics*, 4:419–436, 2009.

[39] R. Breisch, An intuitive approach to speleotopology, *Southwestern Cavers*, 6:72–78, 1967.

[40] D. Bunde, E. Chambers, D. Cranston, K. Milans, D. West, Pebbling and optimal pebbling in graphs, *Journal of Graph Theory*, 57: 215–238, 2008.

[41] P.J. Cameron, The random graph, In: *The Mathematics of Paul Erdős, II*, Algorithms and Combinatorics, **14**, Springer, Berlin, 1997, pp. 333–351.

[42] J. Chalopin, V. Chepoi, N. Nisse, Y. Vaxés, Cop and robber games when the robber can hide and ride, Technical Report, *INRIA-RR7178*, Sophia Antipolis, France, Jan. 2010.

[43] M. Chastand, F. Laviolette, N. Polat, On constructible graphs, infinite bridged graphs and weakly cop-win graphs, *Discrete Mathematics*, 224:61–78, 2000.

[44] P. Chebyshev, Mémoire sur les nombres premiers, *Mém. Acad. Sci. St. Pétersbourg*, 7:17–33, 1850.

[45] E. Chiniforooshan, A better bound for the cop number of general graphs, *Journal of Graph Theory*, 58:45–48, 2008.

[46] F.R.K. Chung, On the cutwidth and the topological bandwidth of a tree, *SIAM Journal of Algebraic Discrete Methods* 6:268–277, 1985.

[47] F.R.K. Chung, Pebbling in Hypercubes, *SIAM Journal on Discrete Mathematics* 2:467-472, 1989.

[48] N.E. Clarke, *Constrained Cops and Robber*, Ph.D. Thesis, Dalhousie University, 2002.

[49] N.E. Clarke, G. MacGillivray, Characterizations of k-copwin graphs, Preprint 2011.

[50] N.E. Clarke, R.J. Nowakowski, Cops, robber, and traps, *Utilitas Mathematica* 60:91–98, 2001.

[51] S. Cook, R. Sethi, Storage requirements for deterministic polynomial time recognizable languages, *Journal of Computer and System Sciences* 13:25-37, 1976.

[52] B. Crull, T. Cundiff, P. Feltman, G. Hurlbert, L. Pudwell, Z. Szaniszlo, Z. Tuza, The cover pebbling number of graphs, *Discrete Mathematics*, 296:15-23, 2005.

[53] N. Dendris, L. Kirousis, D. Thilikos, Fugitive-search games on graphs and related parameters, *Theoretical Computer Science*, 172: 233–254, 1997.

[54] D. Dereniowski, From Pathwidth to Connected Pathwidth, In: *Preceedings of the 28th Symposium on Theoretical Aspects of Computer Science (STACS11)*, 2011.

[55] R. Diestel, *Graph theory*, Springer-Verlag, New York, 2000.

[56] P.A. Dreyer, Applications and variations of domination in graphs, Ph.D. Dissertation, Department of Mathematics, Rutgers University, 2000.

[57] D. Dyer, B. Yang, Öznur Yaşar, On the fast searching problem, In: *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management (AAIM'08)*, 2008.

[58] J. Ellis, I. Sudborough, J. Turner, The vertex separation and search number of a graph, *Information and Computation*, 113:50–79, 1994.

[59] P. Erdős, A. Rényi, On random graphs I, *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[60] P. Erdős, A. Rényi, On the evolution of random graphs, *Publ. Math. Inst. Hungar. Acad. Sci.*, 5:17–61, 1960.

[61] P. Erdős, A. Rényi, Asymmetric graphs, *Acta Mathematica Academiae Scientiarum Hungaricae*, 14:295–315, 1963.

[62] M. Fellows, M. Langston, On search, decision and the efficiency of polynomial time algorithm, In: *Proceedings of the 21st ACM Symposium on Theory of Computing (STOC'89)*, 1989.

[63] P. Fraigniaud, N. Nisse, Monotony properties of connected visible graph searching, *Information and Computation*, 206:1383–1393, 2008.

[64] M. Frankling, Z. Galil, M. Yung, Eavesdropping games: a graph-theoretic approach to privacy in distributed systems, *Journal of ACM*, 47:225–243, 2000.

[65] F.V. Fomin, P. Fraigniaud, N. Nisse, Nondeterministic graph searching: From pathwidth to treewidth, *Algorithmica*, 53:358–373, 2009.

[66] F.V. Fomin, P. Golovach, Graph searching and interval completion, *SIAM Journal on Discrete Mathematics*, 13:454-464, 2000.

[67] F.V. Fomin, P. Golovach, A. Hall, M. Mihalak, E. Vicari, P. Widmayer, How to guard a graph? In: *Proceedings of 15th International Symposium on Algorithms and Computation (ISAAC'08)*, 2008.

[68] F.V. Fomin, P. Golovach, D. Lokshtanov, Guard games on graphs: keep the intruder Out, In: *Proceedings of the 7th International Workshop on Approximation and Online Algorithms (WAOA'09)*, 2010.

[69] F.V. Fomin, D. Kratsch, H. Müller, On the domination search number, *Discrete Applied Mathematics*, 127:565–580, 2003.

[70] F.V. Fomin, N. Petrov, Pursuit-evasion and search problems on graphs, In: *Proceedings of the Twenty-seventh Southeastern International Conference on Combinatorics, Graph Theory and Computing*, 1996.

[71] F.V. Fomin, D. Thilikos, On the monotonicity of games generated by symmetric submodular functions, *Discrete Applied Mathematics*, 131:323–335, 2003.

[72] F.V. Fomin, D. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science*, 399:236–245, 2008.

[73] F.V. Fomin, P.A. Golovach, J. Kratochvíl, N. Nisse, Pursuing fast robber in graphs, *Theoretical Computer Science*, 411:1167–1181, 2010.

[74] P. Frankl, Cops and robbers in graphs with large girth and Cayley graphs, *Discrete Applied Mathematics*, 17:301–305, 1987.

[75] A. Frieze, M. Krivelevich, P. Loh, Variations on Cops and Robbers, accepted to *Journal of Graph Theory.*

[76] R. Ganian, P. Hliněný, J. Kneis, A. Langer, J. Obdržálek, P. Rossmanith, On digraph width measures in parameterized algorithmics, In: *Preceedings of the 4th International Workshop on Parameterized and Exact Computation (IWPEC 2009)*, 2009.

[77] R. Ganian, P. Hliněný, J. Kneis, D. Meister, J. Obdržálek, P. Rossmanith, S. Sikdar, Are there any good digraph width measures? In: *Preceedings of the 5th International Workshop on Parameterized and Exact Computation (IWPEC 2010)*, 2010.

[78] S. Gaspers, M.E. Messinger, R. Nowakowski, P. Pralat, Clean the graph before you draw it, *Information Processing Letters*, 109:463–467, 2009.

[79] G. Gottlob, N. Leone, F. Scarcello, Robbers, marshals, and guards: game theoretic and logical characterizations of hypertree width, *Journal of Computer and System Sciences*, 66:775–808, 2003.

[80] A.S. Goldstein, E.M. Reingold, The complexity of pursuit on a graph, *Theoretical Computer Science*, 143:93–112, 1995.

[81] G. Hahn, Cops, robbers and graphs, *Tatra Mountain Mathematical Publications*, 36:163–176, 2007.

[82] G. Hahn, F. Laviolette, N. Sauer, R.E. Woodrow, On cop-win graphs, *Discrete Mathematics*, 258:27–41, 2002.

[83] G. Hahn, G. MacGillivray, A characterization of k-cop-win graphs and digraphs, *Discrete Mathematics*, 306:2492–2497, 2006.

[84] C.W. Henson, A family of countable homogeneous graphs, *Pacific Journal of Mathematics*, 38:69–83, 1971.

[85] D. Herscovici, Graham's pebbling conjecture on products of cycles, *Journal of Graph Theory* 42:141–154, 2003.

[86] J. Hopcroft, W. Paul, L. Valiant, On time versus space, *Journal of the ACM*, 24:332-337, 1977.

[87] P. Hunter, Complexity and infinite games on finite graphs, Ph.D. Thesis, University of Cambridge, Computer Laboratory, 2007.

[88] P. Hunter, S. Kreutzer, Digraph Measures: Kelly Decompositions, Games, and Orderings, *Theoretical Computer Science*, 399:206–219, 2008.

[89] A. Isaza, J. Lu, V. Bulitko, R. Greiner, A cover-based approach to multi-agent moving target pursuit, In: *Proceedings of The 4th Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2008.

[90] T. Johnson, N. Robertson, P. Seymour, R. Thomas, Directed tree-width, *Journal of Combinatorial Theory, Series B*, 82: 138–154, 2001.

[91] B. Kalyanasundaram, G. Schnitger, On the power of white pebbles, In: *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 1988.

[92] M. Kanté, The rank-width of directed graphs, Preprint 2008.

[93] K. Kara, J. Kratochvil, D. Wood, On the Complexity of the Balanced Vertex Ordering Problem, *Discrete Mathematics and Theoretical Computer Science*, 9:193–202, 2007.

[94] T. Kasai, A. Adachi, S. Iwata, Classes of pebble games and complete problems, *SIAM Journal on Computing*, 8:574-586, 1979.

[95] N.G. Kinnersley, The vertex separation number of a graph equals its path-width, *Information Processing Letters*, 42:345–350, 1992.

[96] L.M. Kirousis, C.H. Papadimitriou, Interval graphs and searching, *Discrete Mathematics*, 55:181-184, 1985.

[97] L.M. Kirousis, C.H. Papadimitriou, Searching and pebbling, *Theoretical Computer Science*, 47:205–216, 1986.

[98] M. Klawe, A tight bound for black and white pebbles on the pyramids, *Journal of ACM*, 32:218–228, 1985.

[99] S. Kreutzer, S. Ordyniak, Digraph decompositions and monotonocity in digraph searching, In: *Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'08)*, 2008.

[100] S. Kreutzer, S. Ordyniak, Distance d-Domination Games, In: *Proceedings of the 35th International Workshop on Graph-Theoretic Concepts in Computer Science (WG'09)*, 2009.

[101] A. S. LaPaugh, Recontamination does not help to search a graph. *Journal of ACM*, 40:224–245, 1993.

[102] T. Lengauer, R. Tarjan, Asymptotically tight bounds on time-soace trade-offs in a pebble game, Journal of ACM, 29:1087–1130, 1982.

[103] L. Lu, X. Peng, On Meyniel's conjecture of the cop number, Preprint 2011.

[104] T. Łuczak, P. Prałat, Chasing robbers on random graphs: zigzag theorem, *Random Structures and Algorithms*, 37:516–524, 2010.

[105] F. Makedon, C. Papadimitriou, I. Sudborough, Topological Bandwidth, *SIAM Journal on Algebraic and Discrete Methods*, 6:418–444, 1985.

[106] F. Makedon, I. Sudborough, On minimizing width in linear layouts, *Discrete Applied Mathematics*, 23:243-265, 1989.

[107] F. Mazoit, N. Nisse, Monotonicity property of non-deterministic graph searching, *Theoretical Computer Science*, 399:169–178, 2008.

[108] N. Megiddo, S. L. Hakimi, M. Garey, D. Johnson, C. H. Papadimitriou, The complexity of searching a graph, *Journal of ACM* 35:18–44, 1998.

[109] M.E. Messinger, R.J. Nowakowski, P. Pralat, Cleaning a network with brushes, *Theoretical Computer Science*, 399 (2008) 191–205.

[110] K. Milans, B. Clark, The complexity of graph pebbling, *SIAM Journal on Discrete Mathematics* 20:769–798, 2006.

[111] D. Moews, Pebbling graphs, *Journal of Combinatorial Theory, Series B*, 55:244–252, 1992.

[112] C. Moldenhauer, N. Sturtevant, Evaluating strategies for running from the cops, In: *Proceedings of IJCAI*, 2009.

[113] H. Nagamochi, Cop-robber guarding game with cycle robber region, In: *Proceedings of the 3rd International Frontiers of Algorithmics Workshop (FAW'09)*, 2009.

[114] R. Nowakowski, Search and sweep numbers of finite directed acyclic graphs, *Discrete Applied Mathematics*, 41:1–11, 1993.

[115] R.J. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Mathematics*, 43:235–239, 1983.

[116] J. Obdržálek, DAG-width: connectivity measure for directed graphs, In: *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*, 2006.

[117] T.D. Parsons, Pursuit-evasion in graphs, *Theory and Applications of Graphs*, Y. Alavi and D. R. Lick, eds., Springer, 426–441, 1976.

[118] S. Peng, C. Ho, T. Hsu, M. Ko, and C. Tang, Edge and Node Searching Problems on Trees, *Theoretical Computer Science*, 240: 429–446, 2000.

[119] P. Prałat, When does a random graph have constant cop number?, *Australasian Journal of Combinatorics*, 46:285–296, 2010.

[120] P. Prałat, N. Wormald, Meyniel's conjecture holds in random graphs, Preprint 2011.

[121] A. Quilliot, Jeux et pointes fixes sur les graphes, *Thèse de 3ème cycle*, Université de Paris VI, 1978, 131–145.

[122] A. Quilliot, *Problèmes de jeux, de point Fixe, de connectivité et de represésentation sur des graphes, des ensembles ordonnés et des hypergraphes*, Thèse d'Etat, Université de Paris VI, 1983, 131–145.

[123] A. Quilliot, A short note about pursuit games played on a graph with a given genus, *Journal of Combinatorial Theory, Series B*, 38:89–92, 1985.

[124] B. Reed, Treewidth and tangles: A new connectivity measure and some applications, *Surveys in Combinatorics*, R.A. Bailey, ed., Cambridge University Press, UK, 87-162, 1997.

[125] D. Richerby, D. Thilikos, Graph searching in a crime wave, *SIAM Journal on Discrete Mathematics*, 23:349–368, 2009.

[126] N. Robertson, P. Seymour, Graph minors I: excluding a forest, *Journal of Combinatorial Theory, Series B*, 35:39–61, 1983.

[127] N. Robertson, P. Seymour, Graph minors II: Algorithmic aspects of tree-width, *Journal of Algorithms*, 7:309–322, 1986.

[128] M. Safari, D-width: A more natural measure for directed tree-width, In: *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS'05)*, 2005.

[129] B.S.W. Schroeder, The copnumber of a graph is bounded by $\lfloor \frac{3}{2}\text{genus}(G)\rfloor + 3$, Categorical perspectives (Kent, OH, 1998), Trends Math., Birkhäuser, Boston, MA, 2001, 243–263.

[130] A. Scott, B. Sudakov, A new bound for the cops and robbers problem, accepted to *SIAM Journal of Discrete Mathematics.*

[131] P. Seymour, R. Thomas, Graph searching and a min-max theorem for tree-width, *Journal of Combinatorial Theory, Series B*, 58:22–33, 1993.
[132] P. Seymour, R. Thomas, Call routing and the ratcatcher, *Combinatorica*, 14:217-241, 1994.
[133] N.J.A. Sloane, Sequences A000088 and A001349, *The On-Line Encyclopedia of Integer Sequences* published electronically at http://oeis.org, 2011.
[134] M. Sipser, *Introduction to the Theory of Computation (2nd edition ed.)*, Thomson Course Technology, 2006.
[135] D. Stanley, B. Yang, Fast searching games on graphs, accepted to *Journal of Combinatorial Optimization.*
[136] K. Sugihara, I. Suzuki, Optimal algorithms for a pursuit-evasion problem in grids, *SIAM Journal on Discrete Mathematics* 2:126-143, 1989.
[137] K. Sugihara, I. Suzuki, M. Yamashita, The searchlight scheduling problem, *SIAM Journal on Computing*, 19:1024-1040, 1990.
[138] I. Suzuki, Y. Tazoe, M. Yamashita, T. Kameda, Searching a polygonal region from the boundary, *International Journal of Computational Geometry and Applications* 11:529-553, 2001.
[139] I. Suzuki, M. Yamashita, Searching for a mobile intruder in a polygonal region, *SIAM Journal on Computing*, 21:863-888, 1992.
[140] A. Takahashi, S. Ueno, Y. Kajitani, Minimal acyclic forbidden minors for the family of graphs with bounded path-width, *Discrete Applied Mathematics*, 127:293-304, 1994.
[141] A. Takahashi, S. Ueno, Y. Kajitani, Mixed searching and proper-path-width, *Theoretical Computer Science*, 137: 253–268, 1995.
[142] D.O. Theis, The cops and robber game on series-parallel graphs, accepted to *Graphs and Combinatorics.*
[143] D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.
[144] B. Yang, Strong-mixed searching and pathwidth, *Journal of Combinatorial Optimization*, 13:47–59, 2007.
[145] B. Yang, Y. Cao, Monotonicity in digraph search problems, *Theoretical Computer Science*, 407:532–544, 2008.
[146] B. Yang, Y. Cao, Monotonicity of strong searching on digraphs, *Journal of Combinatorial Optimization*, 14:411–425, 2007.
[147] B. Yang, Y. Cao, On the monotonicity of weak searching, In: *Proceedings of the 14th International Computing and Combinatorics Conference (COCOON'08)* 2008.
[148] B. Yang, Y. Cao, Digraph searching, directed vertex separation and directed pathwidth, *Discrete Applied Mathematics*, 156:1822–1837, 2008.
[149] B. Yang, D. Dyer, and B. Alspach, Sweeping graphs with large clique number (extended abstract), In: *Proceedings of 15th International Symposium on Algorithms and Computation (ISAAC'04)*, 2004.
[150] B. Yang, D. Dyer, B. Alspach, Sweeping graphs with large clique number, *Discrete Mathematics*, 309:5770–5780, 2009.
[151] B. Yang, Fast edge-searching and fast searching on graphs, *Theoretical Computer Science*, 412:1208–1219, 2011.
[152] B. Yang, R. Zhang and Y. Cao, Searching cycle-disjoint graphs, In: *Proceedings of the 1st International Conference on Combinatorial Optimization and*

Applications (COCOA'07), Lecture Notes in Computer Science, Vol. 4616, 32–43, 2007.

[153] Öznur Yaşar, D. Dyer, D. Pike, M. Kondratieva, Edge searching weighted graphs, *Discrete Applied Mathematics*, 157:1913–1923, 2009.

Department of Mathematics, Ryerson University, Toronto, ON, Canada, M5B 2K3

E-mail address: abonato@ryerson.ca

Department of Computer Science, University of Regina, Regina, SK, Canada, S4S 0A2

E-mail address: boting@cs.uregina.ca