NEW BOUNDS FOR SEARCH NUMBERS USING PROBABILISTIC AND ALGORITHMIC TECHNIQUES

YASHAR TAVAKOLI

New Bounds for Search Numbers Using Probabilistic and Algorithmic Techniques

by

©Yashar Tavakoli

A Thesis submitted to the School of Graduate Studies in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mathematics and Statistics

Memorial University of Newfoundland

April 2012

St. John's Newfoundland

Abstract

Graph searching is a well-studied subject in graph theory. This thesis concentrates on the magnitude of two different search numbers. First, a new upper bound on the fast search number of a general graph is given. The new result improves the existing bound on the fast search number which is given by the brush number. Based on the improved result, an upper bound for almost all graphs is obtained. Next, using an existing lower bound on the fast search number, a lower bound on the fast search number of almost all graphs is derived. Finally, the only existing upper bound on the node search number of a general graph is improved.

Acknowledgements

I would like to thank my supervisor Danny Dyer, whose support made this thesis possible.

Table of Contents

List of Figures

Chapter 1

Introduction

Imagine there are several tunnels with possible interconnections. Also assume there are some intruders hiding in the tunnels. We want to assign a number of searchers to clean the tunnels, i.e. to search for intruders and capture them. Our final goal is to make sure that all the tunnels are clean. How can we do that? The answer is that depending on what we assume, there would be different ways to complete the job. For example, the searchers could locate the intruders by just looking in a tunnel from one end to the other. Or searchers might have limited eyesight or the tunnels might be so long that they could not see throughout the tunnel. Then perhaps they need to traverse each tunnel to capture the intruders. We also may require that once a tunnel is cleaned, it should not be allowed to be "recontaminated" again by the presence of the intruders. Independent of what our assumptions are, since cost is always a concern to us, one can ask what is the minimum number of searchers needed to search a system of tunnels? Can we compute this number efficiently?

A system of tunnels or similar systems can be well-formulated by mathematical objects called graphs, and the theory of graphs is completely capable of modeling this

searching. Depending on what we require, different models can be introduced. The original model occurs in [27], though an earlier motivation was given by [8]. We can define the search number of a graph as the minimum number of searchers needed to clean a graph in each model. Applying the underlying graph theoretical concepts, one can solve and address different problems in graph searching. Being defined as abstract mathematical systems, these models are indeed of theoretical importance. We will see connections between searching models and other problems in graph theory. This fact makes searching even more interesting from a theoretical point of view. Nevertheless many searching models have direct applications in computer science, stretching from networking [13] to VLSI design [11].

This study concentrates on bounding different search numbers in terms of more straightforward graph parameters. Also we will investigate the interrelations between search numbers and other, less intuitive, graph parameters. We mainly consider two search models in this thesis.

Returning to our searching assumptions, suppose that searchers must traverse the edges to clean them. This searching model, first defined in [25], is known as edge searching. If we require that in each step an edge must be searched, then the outlined model is called fast searching. This model was first introduced in [10]. The fast search number of a graph is the minimum number of searchers needed to fast search the graph. From [10] we know that the fast search number of a graph is greater than or equal to half of the number of vertices of odd degree. The fast search number is exactly half of the number of vertices with odd degree if the graph is a tree. The same paper investigates the fast search number of bipartite graphs in different cases. The fast search number of cubic and Halin graphs have been expressed in terms of the

number of odd vertices and the number of leaf blocks respectively [31]. On the computational front, [32] proves the problem of deciding whether the fast search number of a graph is less than or equal to an integer is NP-complete and remains NP-complete even for Eulerian graphs.

In the fast search model, one can require that all the edges incident to a vertex must be searched simultaneously. This restricted fast searching is called brushing. The concept of brushing and the brush number of a graph have been recently introduced in [26]. An upper bound is known for the brush number of a general graph [3]. Interestingly brushing and fast searching are deeply interwoven. In fact, brushing is a restriction of fast searching. This implies that a lower bound on the brush number is also a bound on the fast search number. In this thesis, after reviewing a few mathematical concepts in Chapter 2, in Chapter 3 we will give a bound on the fast search number that is better than the only known bound for the brush number. Based on the improved result, an upper bound on the fast search number for almost all graphs will be obtained. We will then obtain an asymptotic lower bound for almost all graphs.

The next model we study in this thesis is node searching, first introduced in [22]. Here two searchers are required to be placed at the ends of an edge to search the edge, and recontamination is possible. The node search number of a graph is the minimum number of searchers needed to node search the graph. Our aim again is to bound the node search number in terms of other graph parameters. In fact we know that the node search number, the vertex separation number, and the pathwidth of a given graph are different manifestations of a single idea [21, 22]. The more-studied parameter of these three is pathwidth. So in order to study the node search number, we need to investigate the existing results for pathwidth. Now considering the inter-

relation between pathwidth and node search number we can state the following results.

The node search number of a planar graph is asymptotically bounded above by the square root of its number of vertices [5]. Also the node search number of a cubic or sub-cubic graph is asymptotically at most one sixth of the number of its vertices [12]. There are also other more specific graphs with known upper bounds for node search number including outerplanar, Halin, permutation, compatibility and cocomparability graphs, and cographs [6, 7, 14, 18, 24]. But the only known general upper bound is due to [23], which is in terms of the number of vertices and the number of edges of a graph. This general bound works for a general graph as long as it is sparse. In Chapter 4 of this thesis, we will obtain upper bounds on the node search number of general graphs, including dense graphs.

Finally in the last chapter, we consider open problems within the scope of the study.

Chapter 2

Basic Definitions and Theorems

2.1 Probability Theory

Probability theory is a major tool in the present research. The materials in this section are primarily from [14], unless otherwise stated. The three most important concepts in probability theory are *experiment*, *event* and *probability*.

In the mathematical theory of probability, an experiment is defined with a set Ω (also called the sample space), whose elements must represent all outcomes (of the experiment). It would be convenient if all subsets of Ω could be considered events. But for technical reasons, events are defined as a limited collection $\mathcal{A}$ of subsets of Ω.

For a given experiment ω, events $\mathcal{A}$ are subsets of Ω, which form a σ-field. That is, the collection $\mathcal{A}$ of events is defined to have the following properties. If $A \in \mathcal{A}$, then $A \subseteq \Omega$. Furthermore, (i) $\Omega \in \mathcal{A}$; (ii) if A_j, $j \in \mathbb{N}$, all belong to $\mathcal{A}$, then their union is in $\mathcal{A}$; and (iii) if A belongs to $\mathcal{A}$, then so does the complement of A. This definition says that events form an abstract collection which is closed under the operations of

union, intersection, and complement for countable sequences. We define *probability* and *probability space.*

Definition 2.1. Given a sample set Ω and a σ-field $\mathcal{A}$ of subsets of Ω, for any $A \in \mathcal{A}$, a probability Pr is a real-valued function on $\mathcal{A}$ satisfying

1. $0 \leq \Pr[A] \leq 1$,
2. $\Pr[\Omega] = 1$,
3. If T is either a finite or denumerably infinite set of positive integers and if the events A_t, $t \in T$, are mutually exclusive (disjoint), that is $A_i \cap A_j = \varnothing$ for $i \neq j$, then
$$\Pr\left[\bigcup_t A_t\right] = \sum_t \Pr[A_t], \text{ for } A_t \in \mathcal{A}.$$

The triple $(\Omega, \mathcal{A}, \Pr)$ is called a probability space. Also the elements of Ω and $\mathcal{A}$ are called *simple events* and *events* respectively.

Now having defined probability space, we state the following.

Theorem 2.2. Let $\{A_n\}$ be a sequence of events in an arbitrary probability space. Then we have $\lim_{n\to\infty} \Pr[A_n] = \Pr\left[\lim_{n\to\infty} A_n\right]$.

The models considered in this thesis are all discrete and hence Ω is finite. Therefore here we deal with a somewhat simpler space in which $\mathcal{A} = 2^{\Omega}$. We construct this space as follows.

Without loss of generality, let $\Omega = \{1, 2, \ldots, n\}$ and let $\mathcal{A}$ be the set of all subsets of Ω. Let the numbers $\Pr[k]$, $1 \leq k \leq n$, satisfy (i) $\Pr[\{k\}] \geq 0$ for all k and (ii)

$\sum_{k=1}^{n} \Pr[\{k\}] = 1$. Define $\Pr[A] = \sum_{a_j \in A} \Pr[a_j]$ for all $A \in \mathcal{A}$. It is easy to show that Pr is a probability on $\mathcal{A}$. In particular if we take $\Pr[\{k\}] = 1/n$, the underlying probability space is called *uniform*. When we say an *object* is chosen *randomly* from n objects, we imply that the underlying probability space of this selection is a discrete uniform distribution constructed similarly as above with Ω consisting of the n objects. A *permutation* on a set is defined to be an ordering of the elements of the set. Let A be a set, and σ be a permutation on A. Then for $a \in A$, we denote the natural position of a in σ by $\sigma_A(a) \in \mathbb{N}$. Accordingly, a *random permutation* σ *on* a set of n objects is a random selection of a permutation from the set of all permutations of n objects.

On a given probability space, a finite collection $C_1, C_2, \ldots, C_T$ of events is called a partition (of Ω) if $C_i \cap C_j = \varnothing$ for $i \neq j$ and $\Pr\left[\cup_{j=1}^{T} C_j\right] = 1$. A function X defined on Ω is a *simple random variable*, if there is a partition $C_1, C_2, \ldots, C_T$ of Ω such that X is (a finite) constant on each C_j. In this thesis we use simple random variables which we will call random variables.

Let X be a random variable with values $x_1, x_2, \ldots, x_T$ and with partition $C_j = \{\omega \in \Omega \,|\, X(\omega) = x_j\}$, $1 \leq j \leq T$. The sequence $p_j = \Pr[C_j] = \Pr[X = x_j]$, $1 \leq j \leq T$, with $p_j > 0$ is called a *distribution* of X. The value

$$\mathrm{E}[X] = x_1 p_1 + x_2 p_2 + \cdots + x_T p_T \,,$$

is called the *expected value*, or *expectation* of X. In the present study, we apply the *probabilistic method*. In this way the following lemmas [4] will be needed.

Lemma 2.3. (**The Expectation Principle**) Let X be a random variable. Then

in the underlying probability space, there exists $\omega \in \Omega$ such that $X(\omega) \geq \mathrm{E}[X]$ and there exists $\omega' \in \Omega$ such that $X(\omega') \leq \mathrm{E}[X]$.

Lemma 2.4. (**The Linearity of Expectation**) Let $X_1, X_2, \ldots, X_n$ and $c_1, c_2, \ldots, c_n$ be random variables and real constants respectively. Introduce a new random variable X with $X = c_1 X_1 + c_2 X_2 + \cdots + c_n X_n$. Then $\mathrm{E}[X] = c_1 \mathrm{E}[X_1] + c_2 \mathrm{E}[X_2] + \cdots + c_n \mathrm{E}[X_n]$.

If A and B be two events in the probability space $(\Omega, \mathcal{A}, \Pr)$ with $\Pr[B] > 0$, then *conditional probability* $\Pr[A \mid B]$ of A, given B, is defined by the formula

$$\Pr[A \mid B] = \frac{\Pr[A \cap B]}{\Pr[B]}.$$

The above definition can be used repeatedly to obtain the following theorems.

Theorem 2.5. (The Intersection Rule) Let A_i with $1 \leq i \leq n$ be events on the probability space $(\Omega, \mathcal{A}, \Pr)$. Then

$$\Pr\left[\bigcap_{i=1}^{n} A_i\right] = \Pr[A_1] \Pr[A_2 \mid A_1] \Pr[A_3 \mid A_1 \cap A_2] \cdots \Pr[A_n \mid A_1 \cap \cdots \cap A_{n-1}].$$

Theorem 2.6. (The Total Probability Rule) If the events B_j, $j \geq 1$, are such that $\Pr[B_j] > 0$, $B_i \cap B_j = \varnothing$ if $i \neq j$, and $\Pr[\bigcup_j B_j] = 1$, then for an arbitrary event A,

$$\Pr[A] = \sum \Pr[B_j] \Pr[A \mid B_j].$$

We say that events A and B are *independent* if either (i) one of them is of zero probability, or (ii) if and only if $\Pr[B] > 0$, then $\Pr[A \mid B] = \Pr[A]$. Equivalently, A and B are independent if $\Pr[A \cap B] = \Pr[A] \Pr[B]$. Now suppose that n independent trials, each of which results in a "success" with probability p and in a "failure" with probability $1-p$, are to be performed. If the random variable X represents the number of successes that occur in the n trials, then X is said to have a *binomial distribution* with parameters n and p [29]. Then it can be shown that

$$\Pr[X = i] = \binom{n}{i} p^i (1-p)^{n-i}.$$

We will need the following technical lemma.

Lemma 2.7. (A useful version of Chernoff Bound for the Binomial Distribution) Let random variable X have binomial distribution with parameters n and p. Then we have

$$\Pr[X \leq k] \leq \exp\left(-\frac{(pn-k)^2}{2pn}\right).$$

Proof. By [30], for every $0 < \varepsilon < 1$, the Chernoff bound will give us

$$\Pr[X \leq (1-\varepsilon)pn] \leq \exp\left(-\varepsilon^2 pn/2\right).$$

Now take

$$\varepsilon = 1 - \frac{k}{pn},$$

then we will have

$$\begin{aligned}\Pr[X \leq k] &\leq \exp\left(-\left(1-\frac{k}{pn}\right)^2 \frac{pn}{2}\right) \\ &= \exp\left(-\frac{(pn-k)^2}{2pn}\right).\end{aligned}$$

■

2.2 Graph Theory

2.2.1 Basics

A *graph* G is composed of two sets, a finite set of elements $\mathcal{V}(G) = \mathcal{V}$ called vertices, and a finite set $\mathcal{E}(G) = \mathcal{E}$ of unordered pairs of elements of $\mathcal{V}(G)$ called edges. We denote G by $G = (\mathcal{V}, \mathcal{E})$ and refer to $|\mathcal{V}|$ and $|\mathcal{E}|$ as the *order* and *size* of G respectively. In the present thesis an edge $\{u, v\}$ is simplified as uv, also u and v are called *adjacent* vertices. Any vertex which is adjacent to u is called a neighbor of u. Let $\mathcal{U} \subseteq \mathcal{V}$, define $\mathcal{N}(\mathcal{U})$ as the set of vertices outside of $\mathcal{U}$ with at least one neighbor in $\mathcal{U}$.

Now let $G = (\mathcal{V}, \mathcal{E})$ be a graph. The number of edges incident to a vertex v of a graph G is the degree of v, denoted by $\deg(v)$. We also define the minimum and maximum degree of G by $\delta(G) = \min\{\deg(v) \,|\, v \in \mathcal{V}\}$ and $\Delta(G) = \max\{\deg(v) \,|\, v \in \mathcal{V}\}$ respectively. A set $\mathcal{I} \subseteq \mathcal{V}$ is called an *independent set* if and only if $uv \notin \mathcal{E}$ for every $u, v \in \mathcal{I}$. The size of the largest independent set of G is the *independence number* of G, denoted $\alpha(G)$. Furthermore a *maximal independent set* is an independent set

which is not a subset of any other independent set. The following theorem provides a lower bound on the independence number of a general graph [4].

Theorem 2.8. $\alpha(G) \geq \sum_{v \in \mathcal{V}} \frac{1}{\deg(v) + 1}$.

The sequence $p = v_1 v_2 \cdots v_l$ with $v_i \in \mathcal{V}$, $1 \leq i \leq l$ is a *path* from v_1 to v_l provided $v_i v_{i+1} \in \mathcal{E}$ and v_is are distinct. Define $\mathcal{V}(p) = \{v_1, v_2, \cdots, v_l\}$ and $\mathcal{N}(p) = \mathcal{N}(\mathcal{V}(p))$. If $v_1 = v_l$ and no other vertices are repeated, then the sequence is called a *cycle*. The length of the sequence p is denoted by $|p|$. The path p is a *shortest path* from v_1 to v_l if and only if $|p| \leq |p'|$ for every path $p' = v_1 \cdots v_l$. Let $p = v_1 v_2 \cdots v_l$ be a shortest path from v_1 to v_l, then the *distance* from v_1 to v_l is $\mathrm{dist}(v_1, v_l) = |p| - 1$. If there is no path from v_1 to v_l then the shortest path from v_1 to v_l is not defined. The *diameter* of G is $\mathrm{d}(G) = \max\{\mathrm{dist}(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$. Two vertices $v_i, v_j \in \mathcal{V}$ are *connected* if and only if there exists at least one path from v_i to v_j. A graph G is a *connected graph* if and only if every pair of vertices is connected. The *girth* of a graph is the length of the shortest cycle contained in the graph. Girth is not defined for the graphs with no cycles.

In the current study we will use a rough notion of density of a graph. A graph G of order n and size m is called a *dense graph* if m is large relative to n, else it is called a *sparse graph*.

Definition 2.9. Let $\mathcal{V}' \subseteq \mathcal{V}$. Then the *induced subgraph* of G on $\mathcal{V}'$, denoted by $G[\mathcal{V}']$, is $G' = (\mathcal{V}', \mathcal{E}')$ where $uv \in \mathcal{E}'$ if and only if $uv \in \mathcal{E}$ and $u, v \in \mathcal{V}'$.

2.2.2 Some Families of Graphs

We define a few graphs which will be useful later.

Definition 2.10. A graph of order n in which every pair of vertices is adjacent is called a *complete graph* of order n and denoted by K_n.

Definition 2.11. A *Kneser graph* $\mathrm{K}(n,k)$ is the graph whose vertices correspond to the k-element subsets of a n-element set, where two vertices are adjacent if and only if the two corresponding subsets are disjoint. As an example we build $\mathrm{K}(5,2)$. Take a five element set, say $\{1,2,4,5\}$ and let $w_1 = \{3,5\}$, $w_2 = \{2,3\}$, $w_3 = \{2,4\}$, $w_4 = \{1,4\}$, $w_5 = \{1,5\}$, $w_6 = \{1,2\}$, $w_7 = \{4,5\}$, $w_8 = \{1,3\}$, $w_9 = \{2,5\}$, and $w_{10} = \{3,4\}$. Then $\mathrm{K}(5,2)$ can be easily constructed as in Figure 2.1.

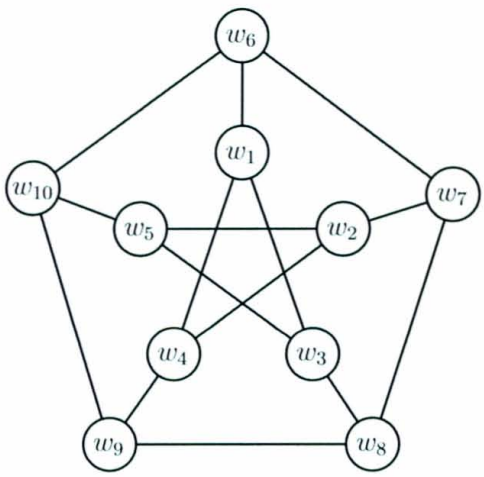

Figure 2.1: The Kneser graph $\mathrm{K}(5,2)$ (Petersen graph)

Definition 2.12. A *well-covered graph* is a graph in which every maximal independent set has the same cardinality. Figure 2.2 gives an example of a well covered graph

in which every maximal independent set is of cardinality two. .

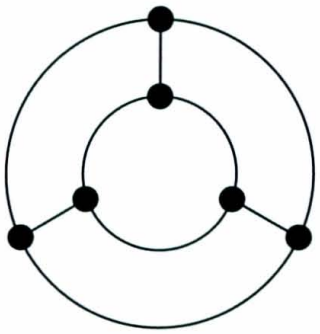

Figure 2.2: A well-covered graph

Definition 2.13. A K_r-free graph is a graph in which no r-subset of vertices induces a K_r. In particular a *triangle-free graph* is a K_3-free graph. In general, K_r-free graphs are also referred to as *clique-free graphs.*

2.2.3 Random Graphs

The theory of random graphs enables us to verify if all graphs but a small family (a family with measure zero in the underlying probability space) share a certain characteristic. This notion will be a key tool in our investigation of fast search number.

Given a real number p, $0 \leq p \leq 1$, the *binomial random graph*, denoted by $\mathcal{G}(n,p)$, is defined by taking Ω as the set of all graphs on n vertices and setting

$$\Pr\left[G \in \mathcal{G}(n,p)\right] = p^{|\mathcal{E}|}\,(1-p)^{\binom{n}{2}-|\mathcal{E}|}$$

for $G = (\mathcal{V}, \mathcal{E})$. It can be viewed as a result of $\binom{n}{2}$ independent coin flippings, one for each pair of vertices, with the probability of success (i.e., drawing an edge) equal to p [19].

Let an isomorphism from a graph G to a graph H be a bijection f from $\mathcal{V}(G)$ to $\mathcal{V}(H)$ such that any two vertices u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. A *graph property* is a class of graphs that is closed under graph isomorphism. If $p = p(n)$ is a fixed function (possibly constant), and $\mathcal{P}$ is a graph property, we may ask how the probability $\Pr[G \in \mathcal{P}]$ behaves for $G \in \mathcal{G}(n, p)$ as $n \to \infty$. If this probability tends to 1, then we say that *almost all graphs* satisfy $\mathcal{P}$ [9].

2.2.4 Searching Models

Many different searching models exist. In this paper we will work with two of them, namely the *node search* model and the *fast search* model which are introduced in [22] and [10] respectively. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. Initially all the edges of G are *contaminated*. We *search* (decontaminate) the edges of G by means of *searchers*. A vertex with a searcher is called a *guarded vertex*. A *search strategy* is a sequence of movements of searchers on the vertices of a graph which searches all the edges of the graph. The characteristics of the movements are identified separately in each searching model.

2.2.4.1 Node Search

A move that searches G belongs to one of the following types:

1. Placing a searcher on a vertex as a guard;

2. Removing a searcher from a vertex u that contains a searcher.

An edge is searched if both of its vertices are simultaneously guarded. A path that does not contain any searcher is called an *unguarded path*. A searched edge remains searched as long as it is not incident to a vertex that is connected by an unguarded path to a contaminated edge. If such a path ever occurs, the edge is said to be *recontaminated*. The graph is (node) searched when there are no contaminated edges.

Definition 2.14. Let G be a graph. The *node search number number* of G denoted by $\mathrm{s}_{\mathrm{node}}(G)$ is the minimum number of searchers which are needed to node search G.

In the context of node searching, two other related notions are also needed to be defined for the purposes of the current study. These are *pathwidth* and *vertex separation number*. The following definitions are from [5] and [21].

A *path decomposition* of a graph $G = (\mathcal{V}, \mathcal{E})$ is a sequence of subsets of vertices $(X_1, X_2, \ldots, X_r)$, such that

1. $\bigcup_{1 \leq i \leq r} X_i = \mathcal{V}$.
2. For all edges $uv \in \mathcal{E}$, there exists an i, with $u, v \in X_i$.
3. For every three indices i, j, k, if $i \leq j \leq k$, then $X_i \cap X_k \subseteq X_j$.

The *width* of a path decomposition $(X_1, X_2, \ldots, X_r)$ is $\max_{1 \leq i \leq r} |X_i| - 1$. The pathwidth of a graph G, denoted by $\mathrm{pw}(G)$, is the minimum width over all possible path decompositions of G.

A (linear) *layout* of G is a bijection $L : \mathcal{V} \to \{1, 2, \ldots, |\mathcal{V}|\}$. Thus L is a permutation of the vertices of G. For any layout L, define $\mathcal{V}_L(i) = \{u \in \mathcal{V} \mid L(u) \leq i$ and there is

some $v \in \mathcal{V}$ such that $uv \in \mathcal{E}$ and $L(v) > i\}$. Then $\mathcal{V}_L(i)$ is the number of vertices of G mapped to integers less than or equal to i that are adjacent to vertices mapped to integers greater that i. The vertex separation number of G with respect to L is defined as $vs_L(G) = \max_{1 \leq i \leq |\mathcal{V}|}\{|\mathcal{V}_L(i)|\}$. Then the vertex separation number of G would be defined with $vs(G) = \min\{vs_L(G) \mid \text{L is a linear layout of } G\}$.

The following theorems are from [22] and [21].

Theorem 2.15. [22] If G is a graph, then $\mathrm{s}_{\mathrm{node}}(G) = \mathrm{vs}(G) + 1$.

Theorem 2.16. [21] If G is a graph, then $\mathrm{pw}(G) = \mathrm{vs}(G)$.

The above theorems imply the following.

Corollary 2.17. If G is a graph, then $\mathrm{s}_{\mathrm{node}}(G) = \mathrm{pw}(G) + 1$.

2.2.4.2 Fast Search

In the fast searching model, a searching action can be one of the following types:

1. Placing a searcher on a vertex as a guard
2. Sliding a searcher from one vertex to another along an edge

An edge is searched when a searcher is slid along it. Sliding the only searcher guarding a vertex is not allowed when there are more than one contaminated edges incident to that vertex. Each edge can be traversed once and thus searched only once, so recontamination is not allowed to occur in the fast search model.

Now assume that we are to search an edge uv by sliding a searcher from u to v. Then two cases might happen. If there are contaminated edges incident to u other than uv, then we need a searcher as a guard on u, and another searcher to slide along uv. This case has been demonstrated in Figure 2.3 (i) in which dotted lines stand for searched edges and stars represent the searchers. The other case that might happen is when all the edges incident to u are searched except for uv. Then one searcher can slide along uv as demonstrated in 2.3 (ii).

(i) (ii)

Figure 2.3: Fast searching actions

The graph is (fast) searched when it contains no contaminated edges.

Definition 2.18. Let G be a graph. The *fast search number number* of G, denoted by $\mathrm{s}_{\mathrm{fast}}(G)$, is the minimum number of searchers needed to fast search G.

2.2.5 Brushing Model

The brushing model is a recently-introduced model on graphs with close relations to the fast search model. We will specifically compare these models. Initially, every edge and vertex of a graph is *dirty* and a fixed number of brushes start on a set of vertices.

At each step, a vertex v and all its incident edges which are dirty may be *cleaned* if there are at least as many brushes on v as there are incident dirty edges. When a vertex is cleaned, every incident dirty edge is traversed (i.e. cleaned) by one and only one brush, and brushes cannot traverse a clean edge. Let ξ be a sequence of vertices. Call the time in which a vertex of ξ is cleaned, a time step. We need $|\mathcal{V}(G)|$ time steps to clean a graph G (a time step can be composed of no movements). A graph is cleaned when every vertex has been cleaned [3].

Definition 2.19. Let G be a graph. The brush number of G denoted by $\mathrm{b}(G)$, is the minimum number of brushes needed to clean G.

If we take a brush strategy and interchange the roles of brushes, dirty edges, and cleaned edges with searchers, contaminated edges, and searched edges respectively, then we would have a fast search strategy (however not every fast search strategy can be translated to a brush strategy). This implies the following lemma.

Lemma 2.20. If G is a graph, then $\mathrm{s}_{\mathrm{fast}}(G) \leq \mathrm{b}(G)$.

2.2.6 Examples and comparisons

In this section we will see a few examples of searching and brushing. Our aim is to demonstrate typical strategies and differences between the fast search number, the node search number, and the brush number.

We define graphs G_1 and G_2 as illustrated in Figure 2.4. First we prove that $\mathrm{s}_{\mathrm{node}}(G_1) = 2$ by showing that we can (node) search G_1 with two searchers and then we will prove

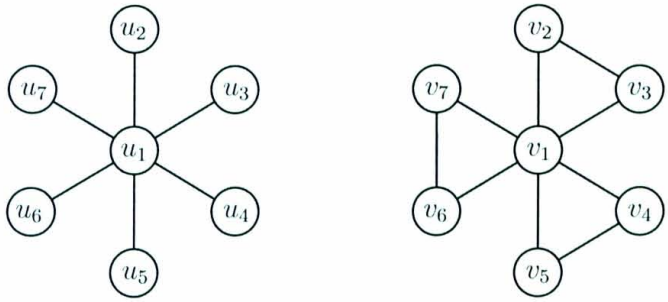

Figure 2.4: Examples for searching and brushing

that G_1 cannot be node-searched with only one searcher.

Place one searcher on u_1 and place a second searcher on u_2. This action searches u_1u_2. Remove the searcher from u_2. Note that u_1u_2 is not recontaminated as it is not incident to a vertex that is connected by an unguarded path to a contaminated edge. The removed searcher can search the remaining edges similarly, so $\mathrm{s}_{\mathrm{node}}(G_1) \leq 2$. But $\mathrm{s}_{\mathrm{node}}(G_1) > 1$, as according to the definition, at least two searchers are needed to search a single edge. Henceforth it follows that $\mathrm{s}_{\mathrm{node}}(G_1) = 2$.

Next we calculate the brush number of G_1. Place the first brush on u_2; since there is only one dirty edge incident to u_2, we can clean u_2. Clean u_3 and u_4 in the same fashion by means of two extra brushes. Now the three brushes are at u_1 and there are three dirty edges incident with u_1. These dirty edges can be traversed and hence cleaned by means of the three brushes. We conclude that $\mathrm{b}(G_1) \leq 3$. We proceed by showing that $\mathrm{b}(G_1) > 2$. Assume we first place two brushes at u_1, the number of dirty edges incident to u_1 is more than the number of brushes present. So the first

brush must be placed on some other vertex, say u_2. Now u_1u_2 can be cleaned with that brush. Now there is one brush present at u_1 but even if we add a second brush to u_1, again the number of dirty edges incident to u_1 would be greater than the number of the brushes present. So we must clean another vertex, say u_3, in the same fashion. Now there are two brushes present at u_1 and four dirty edges incident to u_1, at which point no further vertices can be cleaned. So we conclude that $\mathrm{b}(G_1) = 3$.

It is easy to see that $\mathrm{s}_{\text{fast}}(G_1) \leq 3$. Just take the brush strategy above as a fast search strategy. Thus G_1 can be searched by means of three searchers. Now we show that $\mathrm{s}_{\text{fast}}(G_1) > 2$. Assume we place the first searcher on u_1. Then whatever vertex we place the second searcher on, we will only be able to search a single edge. So assume we place the first searcher on say u_2. To search u_1u_2 slide the searcher to u_1. Now assume that we place the second searcher on u_1. Now there are two searchers at u_1 and one extra edge can be searched with one of these searchers. So assume we place the second searcher on say u_3 and slide it to search u_1u_3. Again there will be two searchers present at u_1, and at most one extra edge can be searched with three contaminated edge. It follows that $\mathrm{s}_{\text{fast}}(G_1) = 3$.

Next we show that $\mathrm{s}_{\text{node}}(G_2) = 3$. Place three searchers, one on v_1, one on v_2, and the last one on v_3. Consequently v_1v_2, v_2v_3, and v_1v_3 would be searched. Now remove the searchers from v_2 and v_3. Note that the searched edges would not be contaminated as they are connected to the contaminated edges by means of v_1, which is guarded. Now place the first removed searcher on v_4 and then the second one on v_5. Consequently v_1v_4, v_4v_5, and v_1v_5 would be searched. Next remove the searchers from v_4 and v_5 and place them on v_6 and v_7. Evidently all the edges are now searched and we will have $\mathrm{s}_{\text{node}}(G_2) \leq 3$. But we also have $\mathrm{s}_{\text{node}}(G_2) > 2$. To show this, assume

there is a strategy to search G_2 with only two searchers. Take a null sequence and at each step of the strategy add an edge to the sequence if it is searched. Note that the last element of the sequence would be the last dirty edge being searched, and also an edge might show up several times in the sequence as it might allow recontamination. Consider the last occurrence of v_2v_3 in the sequence. Then as two searchers are needed to search an edge, the two searchers would be at v_2 and v_3 right after v_2v_3 is searched. But v_2v_3 is either the last edge of the sequence or not. If it is not, then by the construction, there should be at least one contaminated edge after v_2v_3 is searched. Therefore at least one of the searchers at v_2 or v_3 must be removed to search the remaining contaminated edge(s). But no matter which one of the searchers are removed, v_2v_3 become recontaminated, since there is an unguarded path. But this cannot be the case since we considered the last occurrence of v_2v_3 in the sequence. So it must the case that v_2v_3 is the last element of the sequence. But by symmetry, the same argument applies to v_4v_5 and v_6v_7. And this is a contradiction as the sequence can only have one last element. It follows that $\mathrm{s}_{\mathrm{node}}(G_2) = 3$.

We now show that $\mathrm{b}(G_2) \leq 4$. Place two brushes on v_2 and clean it. Now there is a brush at v_3 and one dirty edge incident to it, so we can clean v_3. Note that v_1 cannot be cleaned at this time since there are two brushes at this vertex but the number of incident dirty edges is four. Nevertheless we can clean v_4 and v_5 in the same fashion by placing two brushes at v_4. Now there are four brushes at v_1 and two dirty edges. Therefore v_1 can be cleaned and without adding a new brush we can clean v_6 and then v_7. We proceed by showing that G_2 cannot be cleaned with three brushes. Having three brushes, we cannot begin with v_1, as the number of dirty edges incident with it is greater than three. Without loss of generality start off with v_2. Two brushes are needed to clean v_2. Next without adding a new brush v_3 can be cleaned so that we

end up with a configuration in which there are two brushes are present at v_1 with v_1, v_4, v_5, v_6, and v_7 to be cleaned. Even if we add an extra brush to v_1 we cannot clean it as the number of incident dirty edges is four. Furthermore with only one brush none of the other dirty vertices can be cleaned. We conclude that $\mathrm{b}(G_2) > 3$ and since $\mathrm{b}(G_2) \leq 4$, it follows that $\mathrm{b}(G_2) = 4$.

Finally we prove $\mathrm{s}_{\text{fast}}(G_2) = 2$. Note that G_2 cannot be searched with one searcher as the minimum degree of the graph is two and hence at least two searchers are needed to search the very first edge. Thus it suffices to show that it is possible to search G_2 with two searchers. Place two searchers on v_1. We can search v_1v_2 with one searcher while the other one is guarding v_1. After sliding a searcher, there would be a searcher at v_2. Using this searcher we can search v_2v_3, as this edge is the only contaminated edge incident to v_2. Next, v_1v_3 can be searched accordingly. Now there are two searchers located at v_1 again. Obviously applying the same procedure we can also search the remaining contaminated edges.

We see $\mathrm{s}_{\text{node}}(G_1) < \mathrm{b}(G_1) = \mathrm{s}_{\text{fast}}(G_1)$ but $\mathrm{s}_{\text{fast}}(G_2) < \mathrm{s}_{\text{node}}(G_2) < \mathrm{b}(G_2)$, exemplifying the fundamental differences between these parameters.

2.3 Asymptotics

The present research mostly applies asymptotics to analyze the order of growth. Let $g(x)$ and $f(x)$ be functions. We write $f(x) = \mathcal{O}\big(g(x)\big)$ if and only if there exists a constant $C > 0$ such that $\limsup_{x\to\infty} \left|\frac{f(x)}{g(x)}\right| < C$. We write $f(x) = \Omega\big(g(x)\big)$ if and only if $g(x) = \mathcal{O}\big((f(x)\big)$, or equivalently $\liminf_{x\to\infty} \left|\frac{f(x)}{g(x)}\right| > 0$. Also $f(x) = \Theta\big(g(x)\big)$ if and only if $f(x) = \mathcal{O}\big(g(x)\big)$ and $f(x) = \Omega\big(g(x)\big)$. Finally we write $f(x) = o\big(g(x)\big)$ if and

only if $\lim_{x\to\infty} \dfrac{f(x)}{g(x)} = 0$. In this context Theorem 2.21 will be useful [17].

Theorem 2.21. Let $g(x)$ and $f(x)$ be functions. If $\ln f(x) = o\big(\ln g(x)\big)$, then $f(x) = o\big(g(x)\big)$.

Chapter 3

Bounds for the Fast Search Number

Not very much is known about the order of magnitude of the fast search number of general graphs. This chapter aims to bound the fast search number. First we concentrate on an upper bound. We point out the existing bounds and then improve them. The nature of the improved upper bound will let us derive a result for almost all graphs. In the Section 3.1 we investigate the existing lower bounds and based on an established result, an asymptotic lower bound will be developed for almost all graphs. This result combined with a certain upper bound will reveal an interesting fact about the fast search number of almost all graphs which is presented in the final section.

3.1 Fast Searching vs Brushing

The brush number and fast search number of a graph have interconnections as formulated in Lemma 2.20. Obviously according to the lemma, any upper bound for brush

number would also serve as an upper bound for fast search number. But a general upper bound for brush number has already been established in [3]. Here we give a detailed proof of the bound.

Theorem 3.1. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. We have the following.

$$\mathrm{b}(G) \leq \frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{1}{4} \sum_{\substack{v \in \mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{\deg(v) + 1}$$

To prove this, we first need some preliminaries. For a graph G, let $\mathrm{b}_\xi(G)$ be the minimum number of brushes needed to clean G according to ξ, a sequence of vertices of G. Clearly $\mathrm{b}(G) \leq \mathrm{b}_\xi(G)$, for a sequence ξ of vertices. Now for every $v \in \mathcal{V}(G)$ let $\omega_t(v)$ denote the (minimum) number of brushes at vertex v, and $D_t(v)$ denotes the number of dirty edges incident to v, at time step t. Note that $\omega_0(\xi_{t+1})$ is the number of brushes initially needed for the vertex ξ_{t+1}. Thus the value of $\deg(\xi_{t+1}) - D_t(\xi_{t+1})$ simply refers to the number of cleaned edges incident to ξ_{t+1} at time step t. So one might suspect that the number of brushes initially needed would be $D_t(\xi_{t+1}) - (\deg(\xi_{t+1}) - D_t(\xi_{t+1}))$. But one must also note that this value might be negative at times when the number of present brushes at a vertex is greater than the number of dirty edges in a given time step, at which point obviously no further brushes would be needed. It follows that

$$\begin{aligned} \omega_0(\xi_{t+1}) &= \max\{D_t(\xi_{t+1}) - (\deg(\xi_{t+1}) - D_t(\xi_{t+1})), 0\} \\ &= \max\{2D_t(\xi_{t+1}) - \deg(\xi_{t+1}), 0\}. \end{aligned}$$

Now we can prove Theorem 3.1.

Proof. Let π be a random permutation of the vertices of G taken with uniform distribution. We clean G according to this permutation to get the value of $\mathrm{b}_\pi(G)$. Note that $\mathrm{b}_\pi(G)$ is now a random variable. For a vertex $v \in \mathcal{V}$, we already saw that the number of brushes which should be assigned to v initially is determined by the random variable $X(v) = \max\{0, 2N^+(v) - \deg(v)\}$, where (random variable) $N^+(v)$ is the number of neighbors of v that follow it in the permutation. Thus $N^+(v)$ is the number of dirty neighbors of v at the time when v is cleaned. Note that $N^+(v)$ belongs to the space of random permutations over the set $\{v\} \cup \mathrm{N}(v)$ and the random permutation π induces a uniform, random permutation on $\{v\} \cup \mathrm{N}(v)$. Now we calculate the probability that $N^+(v)$ attains each of the values of $0, 1, \ldots, \deg(v)$. There are $\deg(v)!$ permutations on the set $\{v\} \cup \mathrm{N}(v)$, with v at a certain position; as we can permute $\mathrm{N}(v)$ in $\deg(v)!$ different ways and add v in just one way. Next we juxtapose the rest of the vertices of $\mathcal{V} \setminus \big(\{v\} \cup \mathrm{N}(v)\big)$ in $\big(\deg(v)+2\big)\big(\deg(v)+3\big)\cdots|\mathcal{V}|$ different ways. Now since the space is uniform, for $i = 0, 1, \ldots \deg(v)$ we calculate the probability.

$$\begin{aligned} \Pr\left[N^+(v) = i\right] &= \frac{\deg(v)!\big(\deg(v)+2\big)\big(\deg(v)+3\big)\cdots|\mathcal{V}|}{|\mathcal{V}|!} \\ &= \frac{1}{\deg(v)+1} \end{aligned}$$

Also note that when $\deg(v)$ is even, we will have the following.

$$X(v) = \begin{cases} 0 & \text{if } N^+(v) \leq \deg(v)/2, \\ 2N^+(v) - \deg(v) & \text{if } N^+(v) > \deg(v)/2. \end{cases}$$

We conclude that

$$\begin{aligned}
\mathrm{E}\left[X(v)\right] &= 0 \quad + \sum_{i=\deg(v)/2+1}^{\deg(v)} \left(2i - \deg(v)\right) \Pr\left[N^+(v) = i\right] \\
&= \left(\frac{1}{\deg(v)+1}\right)\Bigg(\left[2\left(\frac{\deg(v)}{2}+1\right) - \deg(v)\right] + \cdots \\
&\quad + \left[2\left(\frac{\deg(v)}{2}+\frac{\deg(v)}{2}-1\right) - \deg(v)\right] \\
&\quad + \left[2\left(\frac{\deg(v)}{2}+\frac{\deg(v)}{2}\right) - \deg(v)\right]\Bigg) \\
&= \frac{2 + \ldots + (\deg(v) - 2) + \deg(v)}{\deg(v)+1} \\
&= \frac{1}{\deg(v)+1}\left(\frac{(\deg(v))^2 - 1}{4}\right) \\
&= \frac{\deg(v)+1}{4} - \frac{1}{4(\deg(v)+1)}\,.
\end{aligned}$$

Following the same argument, when $\deg(v)$ is odd we have

$$\mathrm{E}\left[X\left(v\right)\right] = \frac{1 + \ldots + (\deg(v) - 2) + \deg(v)}{\deg(v)+1} = \frac{\deg(v)+1}{4}\,.$$

Now since $\mathrm{b}_{\pi}(G) = \sum_{v} X(v)$, by the Linearity of Expectation

$$\begin{aligned}
\mathrm{E}[\mathrm{b}_{\pi}(G)] &= \mathrm{E}\left[\sum_{v \in V} X(v)\right] \\
&= \sum_{v \in \mathcal{V}} \mathrm{E}[X(v)] \\
&= \sum_{v \in \mathcal{V}} \frac{\deg(\mathrm{v}) + 1}{4} - \sum_{\substack{v \in \mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{4(\deg(v) + 1)} \\
&= \frac{1}{2} \sum_{v \in \mathcal{V}} \frac{\deg(v)}{2} + \sum_{v \in \mathcal{V}} \frac{1}{4} - \frac{1}{4} \sum_{\substack{v \in \mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{\deg(v) + 1} \\
&= \frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{1}{4} \sum_{\substack{v \in \mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{\deg(v) + 1}.
\end{aligned}$$

Then by the expectation principle, there is a permutation π_0 such that $\mathrm{b}(G) \leq \mathrm{b}_{\pi_0}(G) \leq \mathrm{E}[\mathrm{b}_{\pi}(G)]$ and the assertion holds. ■

As a result, along with Lemma 2.20, we will have the following.

Corollary 3.2. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. Then

$$\mathrm{s}_{\mathrm{fast}}(G) \leq \frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{1}{4} \sum_{\substack{v \in \mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{\deg(v) + 1}.$$

In the other hand, from [26] we know the brush number of a complete graph on n vertices for even n is $n^2/4$ and for odd n is $(n^2 - 1)/4$, while by [10], the fast search number of a complete graph on n vertices is n for $n \geq 4$. Hence for big enough n the

fast search number of a complete graph on n vertices is *arbitrarily* less than its brush number. Other families of graphs with fast search number less than brush number could be easily constructed. This fact suggests that Corollary 3.2 might be tightened.

3.2 A General Upper Bound for Fast Search Number

Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. Fix arbitrary permutations $\sigma_{\mathcal{E}}$ and $\sigma_{\mathcal{V}}$ on $\mathcal{E}$ and $\mathcal{V}$ respectively. We introduce the searching strategy $\mathcal{S}_{(\sigma_{\mathcal{E}}, \sigma_{\mathcal{V}})}$. Let $uv = e \in \mathcal{E}$. Search the edges of G according to the order induced by $\sigma_{\mathcal{E}}$. When we go to search e in $\sigma_{\mathcal{E}}$, we start from that vertex of e which comes first in $\sigma_{\mathcal{V}}$. Without loss of generality assume u comes before v in $\sigma_{\mathcal{V}}$; that is $\sigma_{\mathcal{V}}(u) < \sigma_{\mathcal{V}}(v)$. Then there are four distinct ways that e could be searched.

(i) - If there is no searcher at u and $\deg(u) = 1$, then we place once searcher at u and slide it to v. Else if there os no searcher at u and $\deg(u) > 1$, then we place two searchers at u and slide one of them from u to v and keep it at v.

(ii) - If there is exactly one searcher guarding u and not all the edges incident to u are searched other than e, then we keep that searcher at u. Place a new searcher at u, then slide the new searcher from u to v.

(iii) - If there is exactly one searcher guarding u and all the edges adjacent to u are searched except for e, slide that searcher from u to v.

(iv) - Otherwise, there are at least two searchers at u. Pick one of them and slide it from u to v.

The four possibilities for searching an edge will be referred to as (i), (ii), (iii), and (iv) henceforth whenever needed. Now follow the above procedure for all the edges according to the order defined by $\sigma_{\mathcal{E}}$. Certainly this gives a fast search strategy on G, as the only actions taken are placing searchers and sliding them. Furthermore the first and the second parts of the above procedure guarantees that a searcher does not slide when it is solely guarding a vertex with more than one contaminated edge incident to it, forbidding recontamination. Also since $\sigma_{\mathcal{E}}$ is a permutation on $\mathcal{E}$, we have that each edge is traversed (which is consistent with the fast search model), and more importantly, each edge is traversed exactly once resulting in G being searched according to the definition.

We make a quick example to demonstrate the above strategy. Consider the graph H as shown in the left up left corner of Figure 3.1. We have $\mathcal{V}(H) = \{v_1, v_2, v_3, v_4, v_5\}$ and $\mathcal{E}(H) = \{v_1v_2, v_1v_3, v_1v_4, v_2v_4, v_2v_5, v_3v_4, v_4v_5\}$. Take arbitrary permutations say $\sigma_{\mathcal{V}(H)} = \{v_2, v_3, v_4, v_5, v_1\}$ and $\sigma_{\mathcal{E}(H)} = \{v_3v_4, v_2v_4, v_1v_3, v_2v_5, v_4v_5, v_1v_4, v_1v_2\}$. Now we search H according to these permutations. In each step one edge will be searched as demonstrated in Figure 3.1 from left to right and top down. In the figure, solid lines and dotted lines refer to contaminated and searched edges respectively. Also searchers have been distinguished with asterisks. The first edge to be searched according to $\sigma_{\mathcal{E}(H)}$ would be v_3v_4. But v_3 comes before v_4 in $\sigma_{\mathcal{V}(H)}$ and (i) applies, so we place two searchers on v_3 and slide one of them to v_4. Next we have v_2v_4 with v_2 coming before v_4. Again (i) applies, so we place two searchers on v_2 and one of them would be slid to v_4. Then v_1v_3, and we see v_3 comes before v_1. The only searcher on v_3 will be slid to v_1 based on (iii). Next we need to search v_2v_5 with v_2 coming before v_5. We place a searcher on v_2 and slide it to v_3 as (iii) applies in this case. The next edge to be searched is v_4v_5. Note that v_4 comes before v_5, so (iv) applies and we slide one of the

searchers located at v_4 to v_5. Then turn comes to v_1v_4 with v_4 coming before v_1. We slide the only searcher located at v_4 to v_1 as (iii) applies in this case. The last edge to be searched is v_1v_2 and v_2 comes before v_1. Again (iii) applies and we slide the only searcher at v_2 to v_1. At this point all the edges of H are searched and by definition H is searched by means of five searchers. Note, however that $\mathrm{s}_{\mathrm{fast}}(H) = 3$.

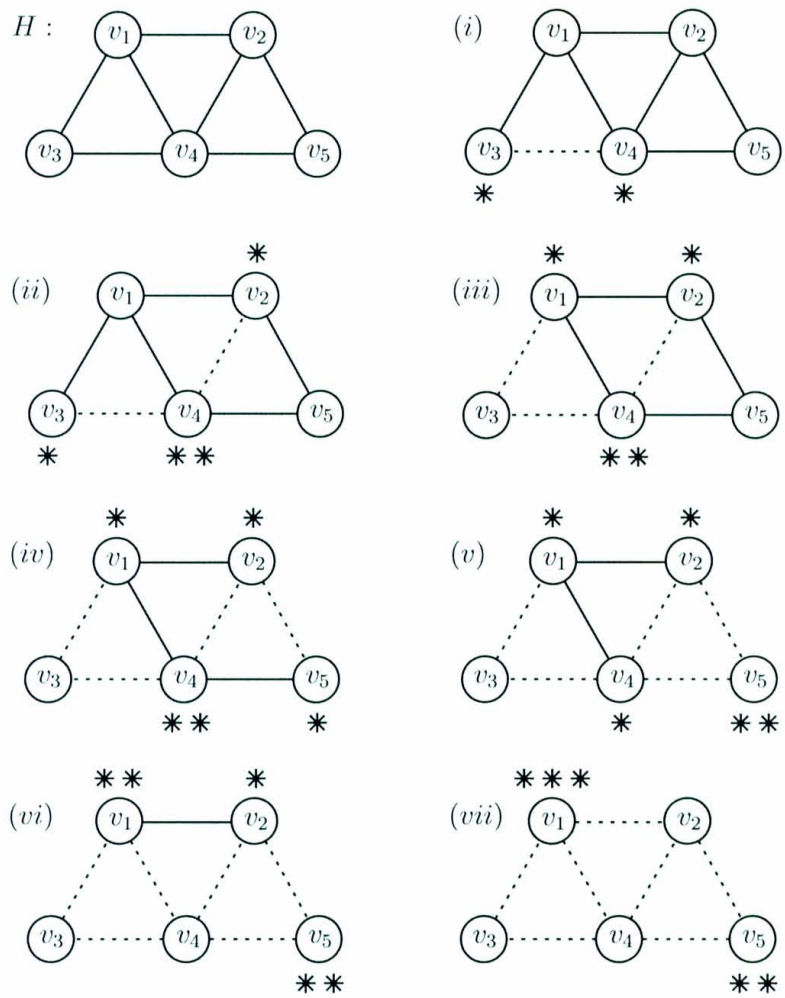

Figure 3.1: Fast searching H according to specified permutations

Based on this idea, the following theorem can be proved giving a better bound than that of Corollary 3.2 for big enough $\delta(G)$.

Theorem 3.3. Let $G = (\mathcal{V}, \mathcal{E})$ be a graph with minimum degree $\delta(G) = \delta$. Then $\mathrm{s}_{\mathrm{fast}}(G) \leq (1/3 + 3/\delta)|\mathcal{E}|$.

Proof. Let $\sigma_{\mathcal{E}}$ and $\sigma_{\mathcal{V}}$ be independent random permutations on $\mathcal{E}$ and $\mathcal{V}$ respectively. Construct the probability space $(\Omega, 2^{\Omega}, \Pr)$, where the sample set Ω consists of all strategies $\mathcal{S}_{(\sigma_{\mathcal{E}}, \sigma_{\mathcal{V}})}$ induced by $\sigma_{\mathcal{E}}$ and $\sigma_{\mathcal{V}}$. Having picked $\sigma_{\mathcal{E}}$ and $\sigma_{\mathcal{V}}$ uniformly at random, the probability space is uniform as simple events are equiprobable. Let $\omega \in \Omega$ be a strategy, define the random variable $X(\omega) = X \in \mathbb{N} \cup \{0\}$ on Ω which indicates the number of searchers needed to search G applying ω. Furthermore for $e \in \mathcal{E}$, define the random variable $X_e(\omega) = X_e \in \{0, 1, 2\}$ indicating the number of searchers we need to assign to e to search it when it is being considered. Then $X = \sum_{e \in \mathcal{E}} X_e$ and hence by the Linearity of Expectation

$$\begin{aligned} \mathrm{E}[X] &= \mathrm{E}\left[\sum_{e \in \mathcal{E}} X_e\right] \\ &= \sum_{e \in \mathcal{E}} \mathrm{E}[X_e]. \end{aligned} \tag{3.1}$$

Next we determine $\mathrm{E}[X_e]$. Let $e = uv$, define another random variable $Y_e^u(\omega) = Y_e^u \in \{0, 1, \ldots, \deg(u) - 1\}$ where $Y_e^u = \left|\{e' \in \mathcal{E} \mid e' = uw, w \in \mathcal{V}, \sigma_{\mathcal{E}}(e') < \sigma_{\mathcal{E}}(e)\}\right|$, the number of edges incident to u which are searched before e. Assume that we assumed $\sigma_{\mathcal{V}}(u) < \sigma_{\mathcal{V}}(v)$. Then by the Total Probability Rule we have

$$\Pr[X_e = 2] = \sum_{i=0}^{\deg(u)-1} \Pr[Y_e^u = i] \Pr[X_e = 2 \mid Y_e^u = i]. \tag{3.2}$$

Now if there is no edge incident to u searched before e, there will be no guard at u when e is being considered and hence exactly two searchers must be assigned according to (i). That is

$$\Pr\left[X_e = 2 \,|\, Y_e^u = 0\right] = 1. \tag{3.3}$$

We next consider the case where at least one edge incident to u has been searched before e. Let $e' = uw$ be the first searched edge in the set of all edges incident to u searched before e. Then $\sigma_{\mathcal{E}}(e') < \sigma_{\mathcal{E}}(e)$. Then by (i), independent of $\sigma_{\mathcal{V}}(u) < \sigma_{\mathcal{V}}(w)$ or $\sigma_{\mathcal{V}}(u) > \sigma_{\mathcal{V}}(w)$, after e' is searched, exactly one searcher would be guarding u. Afterwards by (ii) and (iii), no matter how many adjacent edges are searched, at the time that we consider e, at least one searcher is guaranteed to guard u. Hence as e' was arbitrary, when there is at least one searched incident edge at the time e is considered, we will either need an extra searcher or no extra searchers. That is,

$$\Pr\left[X_e = 2 \,|\, Y_e^u = i\right] = 0 \quad \text{for} \quad i \geq 1 \tag{3.4}$$

which along with (3.2) and (3.3) give

$$\Pr\left[X_e = 2\right] = \Pr\left[Y_e^u = 0\right]. \tag{3.5}$$

Similarly, by the total probability rule we have

$$\Pr\left[X_e = 1\right] \;=\; \sum_{i=0}^{\deg(u)-1} \Pr\left[Y_e^u = i\right] \Pr\left[X_e = 1 \,|\, Y_e^u = i\right]. \tag{3.6}$$

Now if all the $\deg(u) - 1$ edges incident to u are searched before e, as discussed earlier, at least one searcher is guarding u once e is considered; then this would be case (iii)

and no extra searcher would be assigned. That is

$$\Pr[X_e = 1 \mid Y_e^u = \deg(u) - 1] = 0\,. \tag{3.7}$$

On the other hand (3.3) implies

$$\Pr[X_e = 1 \mid Y_e^u = 0] = 0\,,$$

which along with (3.6) and (3.7) gives

$$\Pr[X_e = 1] = \sum_{i=1}^{\deg(u)-2} \Pr[Y_e^u = i] \Pr[X_e = 1 \mid Y_e^u = i]\,. \tag{3.8}$$

Now combining (3.8) with (3.5), by the definition of the expectation we get

$$\begin{aligned} \mathrm{E}[X_e] &= 2\Pr[X_e = 2] + \Pr[X_e = 1] + 0 \\ &= 2\Pr[Y_e^u = 0] + \sum_{i=1}^{\deg(u)-2} \Pr[Y_e^u = i] \Pr[X_e = 1 \mid Y_e^u = i]\,. \end{aligned} \tag{3.9}$$

Similar to the proof of Theorem 3.1, for $i = 0, 1, \ldots, \deg(u) - 1$ we have the following.

$$\Pr[Y_e^u = i] = \frac{1}{\deg(u)} \tag{3.10}$$

Returning to (3.9), we will estimate $\Pr[X_e = 1 \mid Y_e^u = i]$ for $i = 1, 2, \ldots, \deg(u) - 2$. It is easy to see that $\Pr[X_e = 1 \mid Y_e^u = i] = 1$. So assume $i \geq 2$, that is at least two edges are searched before e. Suppose e_k has been searched before e_{k+1} for $k = 1, 2, \ldots, i - 1$.

Observe that by the Total Probability Rule,

$$\Pr\left[X_e = 0 \mid Y_e^u = i\right] = \frac{1}{2}\Pr\left[X_e = 0 \mid Y_e^u = i \text{ and } \sigma_{\mathcal{V}}(u_i) < \sigma_{\mathcal{V}}(u)\right] + \frac{1}{2}\Pr\left[X_e = 0 \mid Y_e^u = i \text{ and } \sigma_{\mathcal{V}}(u_i) > \sigma_{\mathcal{V}}(u)\right]. \quad (3.11)$$

This is because in exactly half of the permutations on $\mathcal{V}$, $\sigma_{\mathcal{V}}(u_i) < \sigma_{\mathcal{V}}(u)$ and in the other half $\mathcal{V}$, $\sigma_{\mathcal{V}}(u_i) > \sigma_{\mathcal{V}}(u)$. Consider the permutation σ' on $\{u, u_1, u_2, \ldots, u_i\}$ induced by $\sigma_{\mathcal{V}}$. Assume $\sigma'_{\mathcal{V}}(u_i) < \sigma'_{\mathcal{V}}(u)$. When e_i is to be searched, since $i \geq 2$,there is at least one searcher guarding u. Note that another searcher will be slid from u_i to u and thus once e is considered, at least two searchers will be available at u. It follows that $\Pr\left[X_e = 0 \mid Y_e^u = i \text{ and } \sigma'(u_i) < \sigma'(u)\right] = 1$. But since σ' is induced by $\sigma_{\mathcal{V}}$ we conclude

$$\Pr\left[X_e = 0 \mid Y_e^u = i \text{ and } \sigma_{\mathcal{V}}(u_i) < \sigma_{\mathcal{V}}(u)\right] = 1 \quad (3.12)$$

Assume $\sigma'_{\mathcal{V}}(u_i) > \sigma'_{\mathcal{V}}(u)$ and let u_k satisfy $\sigma'_{\mathcal{V}}(u_k) < \sigma'_{\mathcal{V}}(u)$. Then once turn comes to e_k, if there is more than one searcher guarding u, one of them will be taken to search e_k. Else there would be exactly one searcher in which case, the searcher would not be taken. We conclude that for every vertex coming after u in σ', at most one searcher will be removed from u. This fact implies that if in σ' the number of vertices coming before u is at least one more than the number of vertices coming after u, then at least two searchers will be available once e is considered. We count the number of the permutations on $\{u, u_1, u_2, \ldots u_i\}$ with u_i coming after u in each of which the number of vertices coming before u is strictly greater that the number of vertices coming after u. We know that in all of these induced permutations, once e is considered, there would be at least two searchers available at u.

Take a permutation on $\{u_1, u_2, \ldots, u_{i-1}\}$ in $(i-1)!$ ways. We can place u in the permutation in $\lfloor (i-1)/2 \rfloor$ ways such that there are at least three (or two when $i-1$ is even) more vertices coming before u than after. If we place u at the end of the permutation, then we can place u_i after u in just one way. If we place u just before the last element of the permutation, then u_i can be placed after u in exactly two ways; immediately after u or after the last element of the permutation. Similarly if we place u in the jth position before the last element of the permutation, then there are j options for the position of u_i. Hence by the uniformity of th probability space we can write

$$\begin{aligned}
\Pr\left[X_e = 0 \,|\, Y_e^u = i \text{ and } \sigma'(u_i) > \sigma'(u)\right] &\geq \frac{(i-1)! \sum_{j=1}^{\lfloor (i-1)/2 \rfloor} j}{\frac{1}{2}(i+1)!} \qquad (3.13)\\
&= \frac{\left\lfloor \frac{i-1}{2} \right\rfloor \left(\left\lfloor \frac{i-1}{2} \right\rfloor + 1 \right)}{i(i+1)}\\
&\geq \frac{\left(\frac{i-1}{2} - 1 \right) \left(\frac{i-1}{2} + 1 - 1 \right)}{i(i+1)}\\
&= \frac{(i-3)(i-1)}{4i(i+1)}
\end{aligned}$$

The last expression is greater than $1/10$ for all $i \geq (1/3)(11 + 2\sqrt{19}) \approx 6.57$. Also one can directly verify that the right hand side of (3.13) is zero for $i = 2$ and greater than $1/10$ for all $3 \leq i \leq 6$. But σ' is induced by $\sigma_{\mathcal{V}}$, hence we conclude

$$\Pr\left[X_e = 0 \,|\, Y_e^u = i \text{ and } \sigma_{\mathcal{V}}(u_i) > \sigma_{\mathcal{V}}(u)\right] \geq 1/10 \text{ for all } i \geq 3, \qquad (3.14)$$

and

$$\Pr\left[X_e = 0 \,|\, Y_e^u = 2 \text{ and } \sigma_{\mathcal{V}}(u_2) > \sigma_{\mathcal{V}}(u)\right] = 0 \tag{3.15}$$

From (3.4) we have $\Pr\left[X_e = 1 \,|\, Y_e^u = i\right] + \Pr\left[X_e = 0 \,|\, Y_e^u = i\right] = 1$. Hence by (3.11), (3.12), (3.14), and (3.15)

$$\Pr\left[X_e = 1 \,|\, Y_e^u = 2\right] \leq \frac{1}{2} \text{ and } \Pr\left[X_e = 1 \,|\, Y_e^u = i\right] \leq \frac{9}{20} \text{ for all } i \geq 3.$$

Now from (3.9) and (3.10)

$$\begin{aligned}
\mathrm{E}[X_e] &= \frac{2}{\deg(u)} + \frac{1}{\deg(u)} \sum_{i=1}^{\deg(u)-2} \Pr\left[X_e = 1 \,|\, Y_e^u = i\right] \\
&= \frac{2}{\deg(u)} + \frac{1}{\deg(u)} \left(\Pr\left[X_e = 1 \,|\, Y_e^u = 1\right] + \sum_{i=2}^{\deg(u)-2} \Pr\left[X_e = 1 \,|\, Y_e^u = i\right] \right) \\
&= \frac{2}{\deg(u)} + \frac{1}{\deg(u)} \left(1 + \frac{1}{2} + \frac{9}{20}\big(\deg(u) - 4\big) \right) \\
&= \frac{2}{\deg(u)} - \frac{6}{20\deg(u)} + \frac{9}{20} \\
&\leq \frac{2}{\delta} + \frac{9}{20}
\end{aligned}$$

It follows from (3.1) that $\mathrm{E}[X] \leq (9/20 + 2/\delta)\,|\mathcal{E}|$ which implies $\mathrm{s}_{\text{fast}}(G) \leq (9/20 + 2/\delta)\,|\mathcal{E}|$ by the Expectation Principle. ■

We claimed that the above theorem establishes a better bound than the brushing bound for graphs with large enough minimum degree. Consider the following. If

$G = (\mathcal{V}, \mathcal{E})$ is a graph with $\delta(G) = \delta > 40$, then $2|\mathcal{E}|/\delta < |\mathcal{E}|/20$ implying $9|\mathcal{E}|/20 + 2|\mathcal{E}|/\delta < |\mathcal{E}|/2 + |\mathcal{V}|(1 - 1/\delta)/4$. But

$$\begin{aligned}
\frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{|\mathcal{V}|}{4\delta} &< \frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{1}{4}\sum_{v\in\mathcal{V}} \frac{1}{\delta + 1} \\
&\leq \frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{1}{4}\sum_{v\in\mathcal{V}} \frac{1}{\deg(v) + 1} \\
&\leq \frac{|\mathcal{E}|}{2} + \frac{|\mathcal{V}|}{4} - \frac{1}{4}\sum_{\substack{v\in\mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{\deg(v) + 1},
\end{aligned}$$

Therefore for all graphs with minimum degree greater than forty, Theorem 3.3 gives a tighter bound than Corollary 3.2.

3.3 An Upper Bound for Fast Search Number of Almost All Graphs

Based on Theorem 3.3, we may derive a result for fast search number of almost all graphs.

Definition 3.4. Let G be a graph of size m. For a fixed $\varepsilon > 0$, we say that G satisfies property $\mathfrak{B}_\varepsilon$ if and only if $\mathrm{s}_{\mathrm{fast}}(G) \leq (9/20 + \varepsilon)\, m$.

We will show that almost every graph satisfies $\mathfrak{B}_\varepsilon$. To do so we also introduce the following graph property.

Definition 3.5. For a fixed $d > 1$, we say that a graph G satisfies property $\mathfrak{D}_d$ if and only if $\delta(G) \geq d$.

Lemma 3.6. For a fixed $d > 1$, almost every graph (in $\mathcal{G}(n,p)$) satisfies $\mathfrak{D}_d$.

Proof. Fix p with $0 < p < 1$. We will show that $\Pr[G \in \mathfrak{D}_d] \to 1$ as $n \to \infty$, where $G \in \mathcal{G}(n,p)$. Without loss of generality, let $\{v_1, v_2, \ldots, v_n\}$ be the set of vertices of G. Note that random variable $\deg(v_i)$ has binomial distribution with parameters $n-1$ and p. Denote the event $\deg(v_i) < d$ by V_i, then by Theorem 2.7 for $1 \leq i \leq n$ we get

$$\begin{aligned} \Pr[V_i] &\leq \exp\left(- \frac{\big[(n-1)p - (d-1)\big]^2}{2pn} \right) \\ &= \exp\big(- \Theta(n) \big). \end{aligned}$$

But we have

$$\begin{aligned} \Pr\left[\bigcup_{i=1}^{n} V_i \right] &\leq \sum_{i=1}^{n} \Pr[V_i] \\ &\leq n \exp\big(- \Theta(n) \big), \end{aligned}$$

tending to zero as $n \to \infty$. This implies the probability of the complement event, that is $\Pr[G \in \mathfrak{D}_d]$ tends to one as $n \to \infty$. ∎

We now state the following.

Corollary 3.7. For a fixed $\varepsilon > 0$, almost every graph satisfies $\mathfrak{B}_\varepsilon$.

Proof. Fix $p > 0$. Let $G \in \mathcal{G}(n,p)$ be a graph of size m and minimum degree δ. Then $\mathrm{s}_{\mathrm{fast}}(G) \leq (9/20 + 2/\delta)\, m$ by Theorem 3.3. Therefore if we let δ be a value greater than d with $d \geq \varepsilon/2$, then $\mathrm{s}_{\mathrm{fast}}(G) \leq (9/20 + \varepsilon)\, m$. Now by the definition, $G \in \mathfrak{D}_d$ implies

$G \in \mathfrak{B}_\varepsilon$ and consequently $\mathfrak{D}_d \subseteq \mathfrak{B}_\varepsilon$. It follows that $\Pr[G \in \mathfrak{B}_\varepsilon] \geq \Pr[G \in \mathfrak{D}_d]$. But by Lemma 3.6, $\lim_{n\to\infty} \Pr[G \in \mathfrak{D}_d] = 1$ so we conclude that $\lim_{n\to\infty} \Pr[G \in \mathfrak{B}_\varepsilon] = 1$. ■

3.4 A Lower Bound for the Fast Search Number of Almost All Graphs

Definition 3.8. For a graph G denote the set of all vertices with odd degree by $\mathrm{V}_o(G)$, and the set all vertices with even degree by $\mathrm{V}_e(G)$.

From [10] we know the following bound on the fast search number.

Theorem 3.9. [10] Let $G = (\mathcal{V}, \mathcal{E})$ be a graph. Then $\mathrm{s}_{\mathrm{fast}}(G) \geq |\mathrm{V}_o(G)| / 2$.

Based on Theorem 3.9 we can establish the following.

Theorem 3.10. Fix $c \in \mathbb{R}$ with $0 < c < 1$. Then for almost all graphs $G \in \mathcal{G}(n, p)$, $\mathrm{s}_{\mathrm{fast}}(G) = \Omega(n^c)$.

In order to show this, we will need a lemma.

Lemma 3.11. Let random variable X_n have binomial distribution with parameters n and $p \in (0, 1)$. Then $\Pr[X_n \text{ is odd}] = \frac{1}{2} \pm o(1)$ and $\Pr[X_n \text{ is even}] = \frac{1}{2} \pm o(1)$.

Proof. Let

$$P(n) = \Pr[X_n \text{ is even}] = \sum_{\substack{1 \le i \le n \\ i \text{ is even}}} \binom{n}{i} p^{n-i}(1-p)^i,$$

and

$$Q(n) = \Pr[X_n \text{ is odd}] = \sum_{\substack{1 \le i \le n \\ i \text{ is odd}}} \binom{n}{i} p^{n-i}(1-p)^i.$$

From the binomial theorem we know

$$\begin{aligned} |P(n) - Q(n)| &= \left| \sum_{\substack{1 \le i \le n \\ i \text{ is even}}} \binom{n}{i} p^{n-i}(1-p)^i - \sum_{\substack{1 \le i \le n \\ i \text{ is odd}}} \binom{n}{i} p^{n-i}(1-p)^i \right| \\ &= \left| (-1)^0 \binom{n}{0} p^n (1-p)^0 + \cdots + (-1)^n \binom{n}{n} p^0 (1-p)^n \right| \\ &= |2p-1|^n . \end{aligned}$$

But $0 < p < 1$ implies $|2p-1| < 1$ and thus

$$|P(n) - Q(n)| = |2p-1|^n = o(1). \tag{3.16}$$

But $P(n)+Q(n) = 1$. Thus by (3.16) we see $P(n) = 1/2 \pm o(1)$ and $Q(n) = 1/2 \pm o(1)$, as required. ■

Now we prove Theorem 3.10.

Proof. Fix $0 < p < 1$ and let $G = (\mathcal{V}, \mathcal{E}) \in \mathcal{G}(n,p)$. Since by Theorem 3.9, $\mathrm{s}_{\text{fast}}(G) \ge |\mathrm{V}_o(G)|/2$, it suffices to show that for almost all graphs $G \in \mathcal{G}(n,p)$, we have $|\mathrm{V}_o(G)| = \Omega(n^c)$.

Consider the event $|V_e(G)| \geq k$ where $k \in \mathbb{N}$. Then the probability of this event is at most the probability that there exists a set $\mathcal{K} \in \mathcal{V}$ with $\deg_G(v)$ even, for all $v \in \mathcal{K}$.

$$\begin{aligned}
\Pr\left[\, |\mathrm{V}_e(G)| \geq k \,\right] &\leq \Pr\left[\, A_{\hat{\mathcal{K}}}\right] \\
&\leq \Pr\left[\bigcup_{\substack{\mathcal{K}\subset\mathcal{V} \\ |\mathcal{K}|=k}} A_{\mathcal{K}}\right] \\
&\leq \sum_{\substack{\mathcal{K}\subset\mathcal{V} \\ |\mathcal{K}|=k}} \Pr\left[A_{\mathcal{K}}\right] \\
&= \binom{n}{k} \Pr\left[A_{\mathcal{K}}\right]. \qquad (3.17)
\end{aligned}$$

Now let $\mathcal{K} = \{u_1, u_2, \ldots, u_k\}$ and B_v be the event in which $\deg_G(v)$ is even. Then

$$\Pr\left[A_{\mathcal{K}}\right] = \Pr\left[B_{u_1} \cap B_{u_2} \cap \cdots \cap B_{u_k}\right]. \qquad (3.18)$$

But by the intersection rule we know the following.

$$\Pr\left[\bigcap_{i=1}^{n} B_{u_i}\right] = \Pr\left[B_{u_1}\right] \Pr\left[B_{u_2}|B_{u_1}\right] \Pr\left[B_{u_3}|B_{u_1} \cap B_{u_2}\right] \cdots \Pr\left[B_{u_k}|B_{u_1} \cap \cdots \cap B_{u_{k-1}}\right] \qquad (3.19)$$

Denote the complement of B_v by B'_v (the event in which $\deg_G(v)$ is odd). Then by the total probability rule we have $\Pr\left[B_{u_2}\right] = \Pr\left[B_{u_2}|B_{u_1}\right] + \Pr\left[B_{u_2}|B'_{u_1}\right]$, and consequently $\Pr\left[B_{u_2}|B_{u_1}\right] \leq \Pr\left[B_{u_2}\right]$. The same argument applies to show $\Pr\left[B_{u_i}|B_{u_1} \cap \cdots \cap B_{u_{i-1}}\right] \leq$

$\Pr[B_{u_i}]$. Therefore by (3.18) and (3.19)

$$\Pr[A_{\mathcal{K}}] \leq \Pr[B_{u_1}] \Pr[B_{u_2}] \cdots \Pr[B_{u_k}]. \tag{3.20}$$

Note that $\deg_G(v)$ (which is a random variable) has binomial distribution with parameters $n-1$ and p, as we have $n-1$ independent trials of being an edge between v and other $n-1$ vertices, each with the probability p for success. Now we apply Lemma 3.11 to get $\Pr[B_{u_i}] = 1/2 \pm o(1)$. Then by (3.17) and (3.20),

$$\Pr\big[\, |\mathrm{V}_e(G)| \geq k \,\big] \leq \binom{n}{k} \left(\frac{1}{2} \pm o(1)\right)^k.$$

For some n_0 and $n > n_0$ we would have $\big(1/2 \pm o(1)\big) \leq \varepsilon$, where $0 < \varepsilon < 1$. We also know from [4] Theorem 1.2.1, that $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$. Since $\binom{n}{k} = \binom{n}{n-k}$, it is also true that $\binom{n}{k} \leq \left(\frac{en}{n-k}\right)^{(n-k)}$. Therefore the following is concluded.

$$\Pr\big[\, |\mathrm{V}_e(G)| \geq k \,\big] \leq \left(\frac{en}{n-k}\right)^{n-k} \varepsilon^k \leq (en)^{n-k} \varepsilon^k.$$

Now set $k = k(n) = n - n^c$, where c is the constant of the hypothesis. Then we have

$$\Pr\big[\, |\mathrm{V}_e(G)| \geq n - n^c \,\big] \leq \left(\frac{en}{\varepsilon}\right)^{n^c} \varepsilon^n.$$

This will imply

$$\lim_{n\to\infty} \Pr\big[\, |\mathrm{V}_e(G)| \geq n - n^c \,\big] \leq \lim_{n\to\infty} \left(\frac{en}{\varepsilon}\right)^{n^c} \varepsilon^n = 0.$$

Note that the limit equals zero as $n^c \ln\left(\frac{en}{\varepsilon}\right) = o\left(n \ln \frac{1}{\varepsilon}\right)$, and hence by Theorem 2.21 we have $\left(\frac{en}{\varepsilon}\right)^{n^c} = o\left(\left(\frac{1}{\varepsilon}\right)^n\right)$. Now applying the Squeeze Theorem we get $\lim_{n\to\infty} \Pr\big[\, |\mathrm{V}_e(G)| \geq n - n^c \,\big] = 0$. But $|\mathrm{V}_o(G)| + |\mathrm{V}_e(G)| = |V(G)|$, so we conclude $\lim_{n\to\infty} \Pr\big[\, |\mathrm{V}_o(G)|$

$\leq n^c \big] = 0$, or equivalently $\lim_{n\to\infty} \Pr\big[\,|\mathrm{V}_o(G)| > n^c\,\big] = 1$. Therefore $\lim_{n\to\infty} \Pr\big[\,|\mathrm{V}_o(G)|\,/\,n^c > 0\,\big] = 1$. By Theorem 2.2 it follows that $\Pr\big[\,\lim_{n\to\infty} |\mathrm{V}_o(G)|\,/\,n^c > 0\,\big] = 1$ and consequently $\Pr\big[\,\liminf_{n\to\infty} |\mathrm{V}_o(G)|\,/\,n^c > 0\,\big] = 1$. Thus with $|\mathrm{V}_o(G)|\,/\,n^c$ being non-negative, according to definition $\Pr\big[\,|\mathrm{V}_o(G)| = \Omega(n^c)\,\big] = 1$ as $n \to \infty$. This completes the proof. ■

Chapter 4

Upper Bounds for the Node Search Number

In this chapter we will find a general upper bound for node search number. Firstly we will review the current literature through the next section.

4.1 Existing Bounds for Node Search Number

To the knowledge of the author, no exclusive upper bound has been developed for node search number. But there are quite a few for pathwidth. Considering the tight connection between node search number and pathwidth, we must consider these results.

Theorem 4.1. [5] If G is a planar graph of order n, then $\mathrm{pw}(G) = \mathcal{O}(n)$.

Theorem 4.2. [12] If G is a cubic or sub-cubic graph (a graph with maximum degree three) of order n, then for every $\varepsilon > 0$, there exists some n_ε such that $\mathrm{pw}(G) \leq (1/6 + \varepsilon)\, n$ for $n > n_\varepsilon$.

But the only known general upper bound on pathwidth is due to [23].

Theorem 4.3. [23] If G is a graph of order n and size m, then $\mathrm{pw}(G) \leq m/5.769 + \mathcal{O}(\log n)$.

By Corollary 2.17, pathwidth and node search number are asymptotically the same. Thus we will have the following immediate corollary.

Corollary 4.4. Let G be a graph of order n and size m. Then $\mathrm{s}_{\mathrm{node}}(G) \leq m/5.769 + \mathcal{O}(\log n)$.

Next we will improve the above bound.

4.2 Bounding the Node Search Number

In this section, we develop two upper bounds. The first one is in terms of the independence number of the graph, and the second one is in terms of the maximum degree of the graph.

Theorem 4.5. If $G = (\mathcal{V}, \mathcal{E})$ is a graph, then $\mathrm{s}_{\mathrm{node}}(G) \leq n - \alpha(G) + 1$.

Proof. Let $\mathcal{I}$ be a maximal independent set of G. One can search G through the following procedure. Place exactly one searcher on every vertex of $\mathcal{V} \setminus \mathcal{I}$. Place one extra searcher on the first vertex of an arbitrary ordering of $\mathcal{I}$, say u_1. Now all the neighbors of u_1 are guarded, as $\mathcal{I}$ is an independent set and hence there is no edge

between u_1 and $\mathcal{I} \setminus \{u_1\}$. Thus all the edges incident with u_1 become searched. But if we remove the searcher from u_1 none of these edges get recontaminated as they are not incident to a vertex that is connected by an unguarded path to a contaminated edge. Now we may remove the searcher from u_1 and place it on the next vertex of the ordering of $\mathcal{I}$ and follow the same procedure until all the edges in $G[\mathcal{I}]$ is completely searched. Note that all the edges in $G[\mathcal{V} \setminus \mathcal{I}]$ are also searched, as all the vertices in $\mathcal{V} \setminus \mathcal{I}$ have been guarded beforehand. It follows that G is searched without any further movements of searchers. Thus $\mathrm{s}_{\mathrm{node}}(G) \leq |\mathcal{V} \setminus \mathcal{I}| + 1 = n - \alpha(G) + 1$. ■

Now bounding the independence number we will have the following.

Corollary 4.6. If G is a d-regular graph of order n, then

$$\mathrm{s}_{\mathrm{node}}(G) \leq \left(\frac{d}{d+1}\right) n + 1.$$

Proof. By Theorem 2.8 we have $\alpha(G) \geq n/(d+1)$. Now the result follows from Theorem 4.5. ■

For graphs close to regular, we can do asymptotically (in terms of the order of the graph) better than Corollary 4.6. But first we need prove some useful lemmas.

Lemma 4.7. If $G = (\mathcal{V}, \mathcal{E})$ is a graph of order n with $\mathcal{U} \subseteq \mathcal{V}$ such that $G[\mathcal{U}]$ is a collection of vertex vertex disjoint paths, then $\mathrm{s}_{\mathrm{node}}(G) \leq n - |\mathcal{U}| + 2$.

Proof. Let $\mathcal{P} = (p_1, p_2, \ldots, p_k)$ be an arbitrary ordering of paths in $G[\mathcal{U}]$. In Figure 4.1 a conceptual illustration of G has been presented, where the dotted lines between

$G[\mathcal{V}\backslash\mathcal{U}]$ and $G[\mathcal{V}(p_i)]$ stand for possible edges between these two components. Note that paths p_i can be as short as one vertex. Search G through the following procedure. Place exactly one searcher on each vertex of $\mathcal{V} \setminus \mathcal{U}$. Without loss of generality let $p_1 = u_1u_2\cdots u_l$, with $l \geq 1$. We place two extra searchers, one searcher on u_1 and one on u_2. But based on the assumption, u_1 has no neighbor in $\mathcal{U}$ except for u_2 which has a searcher on it. So all other possible edges of u_1 are in $\mathcal{U} \setminus \mathcal{V}$, being guarded. It follows that all the edges incident to u_1 in G are searched. But if we remove the searcher from u_1 none of these edges become recontaminated, as they are not incident to a vertex that is connected by an unguarded path to a contaminated edge. Place the removed searcher on u_3. Applying the same argument, all the neighbors of u_2 excluding u_1 now are guarded. But u_1u_2 is already searched so all the edges incident to u_2 are searched now. Besides, we can remove the searcher from u_2 without any recontamination as none of the searched edges are incident to a vertex that is connected by an unguarded path to a contaminated edge. Remove the searcher from u_2 and place it on u_4 and follow similar steps until the edges of p_1 are completely searched. We can also use the two searchers assigned for p_1 to search p_2 through p_k in the same fashion. Note that all the edges in $G[\mathcal{V} \setminus \mathcal{U}]$ are also searched, as all the vertices in $\mathcal{V} \setminus \mathcal{U}$ have been simultaneously guarded firstly. It follows that G is searched without any further movements of searchers. Thus $\mathrm{s}_{\mathrm{node}}(G) \leq |\mathcal{V} \setminus \mathcal{U}| + 2 = n - |\mathcal{U}| + 2$. ■

Lemma 4.8. If G is a graph and $p = u_1u_2\cdots u_k$ is a shortest path from u_1 to u_k in G with $\mathcal{U} = \{u_1, u_2, \ldots, u_k\}$, then $G[\mathcal{U}]$ is exactly p.

Proof. The graph $G[\mathcal{U}]$ consists of p and other possible edges. There exist other edges if and only if $G[\mathcal{U}]$ has a cycle, contradicting the fact that p is a shortest

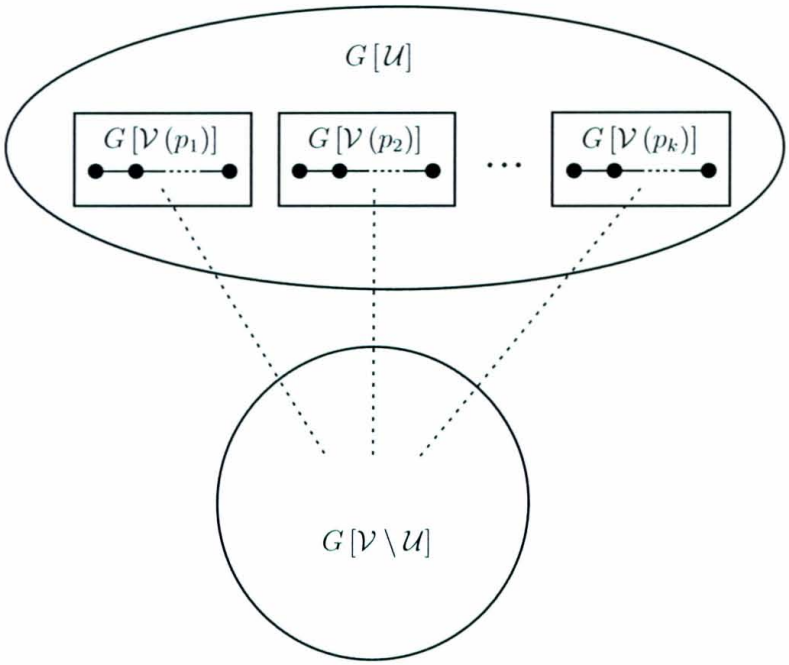

Figure 4.1: A picture of G

path. ■

Lemma 4.9. If $G = (\mathcal{V}, \mathcal{E})$ is a graph with $\mathrm{d}(G) = k$, then for every $k' < k$, there exist $u', v' \in \mathcal{V}$ such that $\mathrm{dist}(u', v') = k'$.

Proof. Since $\mathrm{d}(G) = k$ there exists a shortest path of length k in G say $p = uw_1 \cdots w_{k-1}v$. Take $p' = uw_1 \cdots w_{k'}$, then p' is a shortest path from from u to $w_{k'}$, as if not, there would be some p'' with length less than p' by replacing which we can construct a path from u to v with length less than k, a contradiction. Thus $\mathrm{dist}(u, w_{k'}) = k'$, as required. ■

Lemma 4.10. If $G = (\mathcal{V}, \mathcal{E})$ is a connected graph of order $n > 2$ and maximum degree $\Delta = \Delta(G) > 2$, then $\mathrm{d}(G) \geq \left(\log_{\Delta} n\right) - 1$.

Proof. Let $v \in \mathcal{V}$. Then at most Δ vertices are distance 1 from v, and at most $\Delta(\Delta-1)^{i-1}$ vertices are distance i from v for $i \geq 2$. It follows that

$$\begin{aligned} n &\leq 1 + \Delta + \Delta(\Delta-1) + \cdots + \Delta(\Delta-1)^{\mathrm{d}(G)-1} \\ &= \frac{\Delta(\Delta-1)^{\mathrm{d}(G)} - 2}{\Delta - 2} \\ &\leq \Delta^{\mathrm{d}(G)+1}. \end{aligned}$$

Hence $\mathrm{d}(G) \geq \left(\log_{\Delta} n\right) - 1$. ■

Lemma 4.11. If $G = (\mathcal{V}, \mathcal{E})$ is a connected graph of order n with fixed maximum degree $\Delta = \Delta(G) > 2$, then for every fixed $\varepsilon > 0$, there exists a set $\mathcal{U}_{\varepsilon} \subseteq \mathcal{V}$ such that $G[\mathcal{U}_{\varepsilon}]$ is a collection of vertex disjoint paths and

$$|\mathcal{U}_{\varepsilon}| \geq \left(\frac{1}{\Delta - 1} - \varepsilon\right) n - \mathcal{O}(1).$$

Proof. Having fixed ε, define the following algorithm on G:

Algorithm.

Set $\mathcal{U} := \varnothing$, $\mathcal{W} := \mathcal{V}$

Fix $c := \left\lceil \left(\frac{2\Delta}{\varepsilon(\Delta-1)} - 2\Delta \right) / (\Delta - 1) \right\rceil + 1$

While $|\mathcal{W}| \geq \Delta^{c+1}$

1. Pick a shortest path p in $G[\mathcal{W}]$ with $|p| = c$
2. $\mathcal{U} := \mathcal{U} \cup \mathcal{V}(p)$
3. $\mathcal{W} := \mathcal{W} \setminus \left(\mathcal{V}(p) \cup \mathcal{N}(p) \right)$
4. Maintaining the maximum degree Δ, keep $G[\mathcal{W}]$ connected by adding as many edges as needed.

End

Note that by Lemma 4.10, $|\mathcal{W}| \geq \Delta^{c+1}$ implies $\mathrm{d}(G[\mathcal{W}]) \geq c$. Also line four of the loop keep the graph connected (we will verify the condition later). Then by Lemma 4.9, in each iteration of the loop we are guaranteed to have a shortest path p in $G[\mathcal{W}]$ with $|p| = c$. In each iteration we simply remove a shortest path of a fixed length and its neighbors from the graph. After the termination of the algorithm, $\mathcal{U}$ is the set of all removed shortest paths. Having removed all the neighbors of paths, $G[\mathcal{U}]$ is a collection of vertex disjoint paths by Lemma 4.8.

Before determining the size of $|\mathcal{U}|$, note that after the execution of line three of the loop, the graph may get disconnected say into components $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$. We know there exists a vertex u_i in $\mathcal{C}_i$ with $\deg_{\mathcal{C}_i}(u_i) \leq \Delta - 1$ since the component had been connected to the ret of the components before the execution of line three. Thus $\mathcal{C}_i$ is not

a complete graph and there should be another vertex v_i in $\mathcal{C}_i$ with $\deg_{\mathcal{C}_i}(v_i) \leq \Delta - 1$. With this fact being said, it is easy to see that the condition of line four of the loop can be satisfied; it suffices to add $\{u_1v_2, u_2v_3, \ldots, u_{k-1}v_k\}$ to $G[\mathcal{W}]$.

Now we determine $|\mathcal{U}|$ when the algorithm terminates. At each iteration, $|\mathcal{U}|$ grows by $c+1$. On the other hand, consider Figure 4.2. Take the path $p = w_1w_2 \cdots w_{c+1}$ and cluster the vertices as shown in the figure. There are at most Δ vertices in the first bag, and at most Δ vertices in the last bag. Also each of the $c-1$ middle bags contains at most $\Delta - 1$ vertices. So $|\mathcal{V}(p) \cup \mathcal{N}(p)| \leq (\Delta-1)(c-1)+2\Delta$. This implies at each iteration $|\mathcal{W}|$ is reduced by at most $(\Delta-1)(c-1)+2\Delta$. Thus it follows that

$$\begin{aligned} |\mathcal{U}| &\geq \left\lfloor \frac{n - \Delta^{c+1} + 1}{(\Delta-1)(c-1)+2\Delta} \right\rfloor (c+1) \\ &= \left(\frac{c+1}{(\Delta-1)(c-1)+2\Delta} \right) n - \mathcal{O}(1). \end{aligned} \tag{4.1}$$

Note that the bracket term is a lower bound for the number of iterations of the loop and the asymptotic term $\mathcal{O}(1)$ replaces a function of fixed Δ and c.

$$\begin{aligned} \frac{c+1}{(\Delta-1)(c-1)+2\Delta} &\geq \frac{c-1}{(\Delta-1)(c-1)+2\Delta} \\ &= \frac{1}{\Delta-1} - \frac{2\Delta}{(\Delta-1)\big((\Delta-1)(c-1)+2\Delta\big)} \\ &\geq \frac{1}{\Delta-1} - \varepsilon, \end{aligned}$$

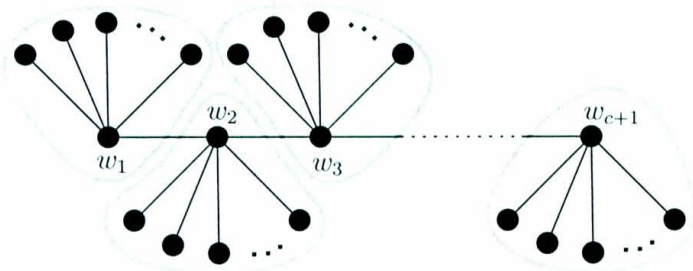

Figure 4.2: Removing a path

as

$$c \geq \left[\left(\frac{2\Delta}{\varepsilon(\Delta - 1)} - 2\Delta \right) / (\Delta - 1) \right] + 1.$$

Hence we conclude by (4.1) that

$$|\mathcal{U}| \geq \left(\frac{1}{\Delta - 1} - \varepsilon \right) n - \mathcal{O}(1).$$

Recall that to keep the graph connected we added some edges. Thus $\mathcal{U}$ is a collection of vertex-disjoint paths in $\hat{G}$, composed of the same vertex and edge sets as G, plus possibly some extra edges. If we remove the set of the extra edges from $\mathcal{U}$, it would be still a collection of vertex disjoint paths. So we can set $\mathcal{U}_\varepsilon$ with $\mathcal{U}$ excluding the extra edges. ■

Now we can state the following theorem.

Theorem 4.12. Let G be a graph of order n with fixed maximum degree $\Delta =$

$\Delta(G) > 2$. Then for every $\varepsilon > 0$

$$\mathrm{s}_{\mathrm{node}}(G) \leq \left(\frac{\Delta - 2}{\Delta - 1} + \varepsilon\right) n + \mathcal{O}(1).$$

Proof. From Lemma 4.11 we know that there exists a set $\mathcal{U}_\varepsilon \subseteq \mathcal{V}$ such that $G[\mathcal{U}_\varepsilon]$ is a collection of vertex disjoint paths such that

$$|\mathcal{U}_\varepsilon| \geq \left(\frac{1}{\Delta - 1} - \varepsilon\right) n - \mathcal{O}(1).$$

But by Lemma 4.7,

$$\begin{aligned} \mathrm{s}_{\mathrm{node}}(G) &\leq n - \left[\left(\frac{1}{\Delta - 1} - \varepsilon\right) n - \mathcal{O}(1)\right] + 2 \\ &\leq \left(\frac{\Delta - 2}{\Delta - 1}\right) n + \varepsilon n + \mathcal{O}(1) \\ &= \left(\frac{\Delta - 2}{\Delta - 1} + \varepsilon\right) n + \mathcal{O}(1). \end{aligned}$$

■

It remains to show that we have made an improvement. Let G be a graph of order n and size m and fixed maximum degree Δ. First take n big enough such that the asymptotic term in Theorem 4.12 is less than the asymptotic term of Theorem 4.3. Now let $\delta > 11$. Then for every $\Delta > 11$ with ε small enough, we have

$$5.769 \left(\frac{\Delta - 2}{\Delta - 1} + \varepsilon\right) n < \frac{\delta n}{2} \leq m.$$

So the bound given in Theorem 4.12 is tighter than that of Theorem 4.3 for all graphs with $\delta > 11$.

For a d-regular graph G of order n with $d > 2$ Theorem 4.12 tells us $\mathrm{s}_{\mathrm{node}}(G) \leq \left(\frac{d-2}{d-1} + \varepsilon\right) n + \mathcal{O}(1)$ which asymptotically is a tighter bound compared to that of Corollary 4.6. Nonetheless, for the cases in which the graph is far from regular or we have a good knowledge of the independence number of the graph we can apply Theorem 4.5. Our first example is the family of Kneser graphs. Kneser graphs are not necessarily sparse, cubic or planar. So existing results do not tell us much about their pathwidth. In contrast we know the following.

Lemma 4.13. [1] For the Kneser graph $\mathrm{K}(n,k)$, we have $\alpha\big(\mathrm{K}(n,k)\big) = \binom{n-1}{k-1}$.

This piece of information leads us to the following result.

Corollary 4.14. For the Kneser graph $\mathrm{K}(n,k)$, we have

$$\mathrm{s}_{\mathrm{node}}\big(\mathrm{K}(n,k)\big) \leq \frac{n-k}{n}\left|\mathrm{V}\big(\mathrm{K}(n,k)\big)\right| + 1\,.$$

Proof. Since $\left|\mathrm{V}\big(\mathrm{K}(n,k)\big)\right| = \binom{n}{k}$, and by Lemma 4.13, $\alpha\big(\mathrm{K}(n,k)\big) = \binom{n-1}{k-1}$, the result

would be a straightforward consequence of theorem 4.5 and Pascal's identity as follows.

$$\begin{aligned}
\mathrm{s}_{\mathrm{node}}\big(\mathrm{K}\,(n,k)\big) &\leq \big|\mathrm{V}\big(\mathrm{K}\,(n,k)\big)\big| - \alpha\big(\mathrm{K}\,(n,k)\big) + 1 \\
&= \binom{n}{k} - \binom{n-1}{k-1} + 1 \\
&= \binom{n-1}{k} + 1 \\
&= \frac{n-k}{n}\binom{n}{k} + 1 \\
&= \frac{n-k}{n}\big|\mathrm{V}\big(\mathrm{K}\,(n,k)\big)\big| + 1\,.
\end{aligned}$$

■

We proceed with a few other examples.

Corollary 4.15. If G is a well-covered graph of order n, then $\mathrm{s}_{\mathrm{node}}\,(G) \leq n/2 + 1$.

Proof. From [28] we know $\alpha\,(G) \geq n/2$. The result follows from Theorem 4.5. ■

Corollary 4.16. If G is a graph of order n with odd girth $2k+1$, and $\delta\,(G) > n/(k+1)$ where $k \geq 4$, then $\mathrm{s}_{\mathrm{node}}\,(G) \leq (k+1)n/(2k+1)$.

Proof. From [2] we know $\alpha(G) \geq kn/(2k+1)$. The result follows from Theorem 4.5. ■

Corollary 4.16 indicates that node search number of a dense graph need not be *very* large. (A construction of dense graphs with large independence numbers is discussed in [20].) Further examples of such dense graphs can be found among clique-free graphs.

Corollary 4.17. Let G_1, G_2 and G_3 be graphs of order n, where G_1 is a triangle-free graph with $\delta(G_1) > 2n/5$, G_2 is a K_4-free graph with $\delta(G_2) > 3n/5$ and G_3 is a K_r-free graph, with $\delta(G_3) > (2r-5)n/(2r-3)$, $r \geq 5$, then $s_{node}(G_1) \leq n/2 + 1$, $s_{node}(G_2) \leq 2n/3 + 1$ and $s_{node}(G_3) \leq (r-2)n/(r-1) + 1$.

Proof. By [16] we know G_1, G_2, and G_3 exist and we have $\alpha(G_1) \geq n/2$, $\alpha(G_2) \geq n/3$ and $\alpha(G_3) \geq n/(r-1)$. Now the result would follow from Theorem 4.5. ■

Chapter 5

Conclusion and Further Works

In this thesis, we studied two edge searching models, node searching and fast searching. We concentrated on the magnitude of the node search number and the fast search number of general graphs in the context of classical graph theory and also using the theory of random graphs. Consider a graph $G = (\mathcal{V}, \mathcal{E})$ of order n, size m, maximum degree Δ, and minimum degree δ. We first showed that the upper bound on the brush number introduced in [3] is in fact also an upper bound for the fast search number. Thus,

$$\mathrm{s}_{\mathrm{fast}}(G) \leq \frac{m}{2} + \frac{n}{4} - \frac{1}{4} \sum_{\substack{v \in \mathcal{V} \\ \deg(v) \text{ is even}}} \frac{1}{\deg(v) + 1}.$$

But applying the probabilistic method we improved the result to $\mathrm{s}_{\mathrm{fast}}(G) \leq (9/20 + 2/\delta)\, m$ for $\delta > 18$. Next we saw that for almost all graphs the fast search number is less than one third of the size of graph. To shed more light on the order of magnitude of the fast search number, fast searching was asymptotically investigated. We discovered that for every rational $c < 1$, the fast search number of almost all graphs is $\Omega(n^c)$. This is actually an asymptotic lower bound for almost all graphs. Next we studied the node search number which is essentially almost pathwidth. Here we worked on the

problem of improving the only existing upper bound on the node search number of a general graph which has been discovered in the context of pathwidth. This bound on pathwidth manifests itself in node searching as $\mathrm{s}_{\mathrm{node}}(G) \leq m/5.769 + \mathcal{O}(\log n)$. For fixed Δ, we proved that

$$\mathrm{s}_{\mathrm{node}}(G) \leq \left(\frac{\Delta - 2}{\Delta - 1} + \varepsilon\right) n + \mathcal{O}(1),$$

and showed that this later bound is tighter than the former when $\delta > 11$.

All of the results in the node search section can be easily used to get new results for pathwidth, using Corollary 2.17. But there is also one more algorithmic consideration. Clearly Theorems 4.5 and 4.12 are *constructive* proofs as we have deterministically constructed a search strategy to prove each theorem. A basic question here would be; is it possible to derive path decompositions from these search strategies for a graph? The answer is yes. Fortunately, the approaches taken in [22] and [21] are both constructive. So for any connected graph, it would be natural to take the node search strategies in Theorems 4.5 and 4.12, *transform* them to vertex separations applying the algorithm introduced in [22], and then transform the vertex separations to path decompositions using the algorithm introduced in [21]. It must be noted though, the hidden constant in Theorem 4.12 is very large which from an algorithmic point of view makes the whole procedure quite inapplicable for graphs of small order.

In studying the fast search number of graphs we concentrated on the size m of the graph. But there are different search numbers which are bounded by means of the order of graph. Certain observations reveal the fast search number can be properly bounded in terms of size. In the other hand we saw that the fast search number of

almost all graphs grows at least sub-linearly with the order. The author strongly suspects the growth is most probably linear.

The only lower bound on the fast search number given in this thesis is an asymptotic lower bound for almost all graphs. Other lower bounds are available, but most of them bound the fast search number of certain classes of graphs. While separate investigation of different classes of graphs is valuable on its own merit, the question still remains whether we are able to give a lower bound on the fast search number of a general graph in terms of elementary graph parameters.

For the node search number, we applied a maximum degree analysis. But the author believes that a tighter bound can and should be in terms of the average degree of graph. Nonetheless, a better bound in terms of maximum degree is also reasonable; Theorem 4.12 implies the asymptotic pathwidth of a cubic or sub-cubic graph of order n is less than almost $n/2$ while from Theorem 4.2 we know it is less than almost $n/6$. The trade-off made to obtain a general bound in the current study, could be shrunk for a class of graphs with fixed maximum degree. The first step could be the investigation of quartic and sub-quartic graphs. No general upper bound has been found for this class of graphs yet.

Bibliography

[1] M. Aigner, G. M. Ziegler, *Proofs from The Book.* Fourth edition. Springer-Verlag, Berlin, 2010. viii+274 pp.

[2] M. O. Albertson, L. Chan, R. Haas, *Independence and graph homomorphisms.* J. Graph Theory 17 (1993), no. 5, 581-588.

[3] N. Alon, P. Prałat, N. Wormald, *Cleaning regular graphs with brushes.* SIAM J. Discrete Math. 23 (2008/09), no. 1, 233-250.

[4] N. Alon, J. H. Spencer, *The probabilistic method.* Third edition. With an appendix on the life and work of Paul Erdős. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Inc., Hoboken, NJ, 2008. xviii+352 pp.

[5] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth.* Theoret. Comput. Sci. 209 (1998), no. 1-2, 1-45.

[6] H. L. Bodlaender, K. Hans; T. Kloks, D. Kratsch, *Treewidth and pathwidth of permutation graphs.* Automata, languages and programming (Lund, 1993), 114–125, Lecture Notes in Comput. Sci., 700, Springer, Berlin, 1993.

[7] H. L. Bodlaender, R. H. Möhring, *The pathwidth and treewidth of cographs.* SWAT 90 (Bergen, 1990), 301–309, Lecture Notes in Comput. Sci., 447, Springer, Berlin, 1990.

[8] R. L. Breisch, *Southwestern Cavers*, 6 (1967), 72-78.

[9] R. Diestel, *Graph theory.* Translated from the 1996 German original. Graduate Texts in Mathematics, 173. Springer-Verlag, New York (1997), xiv+289 pp.

[10] D. Dyer, B. Yang, and Ö. Yaşar, *On the Fast Searching Problem*, Lecture Notes in Computer Science, Volume 5034 (2008), 143-154.

[11] M. R. Fellows, M. A. Langston, *On search, decision, and the efficiency of polynomial-time algorithms.* J. Comput. System Sci. 49 (1994), no. 3, 769-779.

[12] F. V. Fomin, K. Høie, *Pathwidth of cubic graphs and exact algorithms.* Inform. Process. Lett. 97 (2006), no. 5, 191-196.

[13] M. Franklin, Z. Galil, M. Yung, *Eavesdropping games: a graph-theoretic approach to privacy in distributed systems.* J. ACM 47 (2000), no. 2, 225-243.

[14] J. Galambos, *Advanced probability theory.* Second edition. Probability: Pure and Applied, 10. Marcel Dekker, Inc., New York, 1995. xii+461 pp. ISBN: 0-8247-9332-3 (Reviewer: Paul Embrechts), 60-01.

[15] R. Garbe, *Tree-width and path-width of comparability graphs of interval orders.* Graph-theoretic concepts in computer science (Herrsching, 1994), 26-37, Lecture Notes in Comput. Sci., 903, Springer, Berlin, 1995.

[16] W. Goddard, J. Lyle, *Dense graphs with small clique number.* J. Graph Theory 66 (2011), no. 4, 319-331.

[17] D. H. Greene, D. E. Knuth, *Mathematics for the analysis of algorithms.* Third edition. Progress in Computer Science and Applied Logic, 1. Birkhäuser Boston, Inc., Boston, MA, 1990. viii+132 pp.

[18] M. Habib, R. H. Möhring, *Treewidth of cocomparability graphs and a new order-theoretic parameter.* Order 11 (1994), no. 1, 47–60.

[19] S. Janson, T. Łuczak, A. Rucinski, *Random graphs.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000. xii+333 pp.

[20] P. K. Jha, S. Klavžar, *Independence in direct-product graphs.* Ars Combin. 50 (1998), 53-63.

[21] N. Kinnersley, *The vertex separation number of a graph equals its path-width.* Inform. Process. Lett. 42 (1992), no. 6, 345-350.

[22] L. M. Kirousis, C. H. Papadimitriou, *Searching and pebbling.* Theoret. Comput. Sci. 47 (1986), no. 2, 205-218.

[23] J. Kneis, D. Mölle, S. Richter, P. Rossmanith, *A bound on the pathwidth of sparse graphs with applications to exact algorithms.* (English summary) SIAM J. Discrete Math. 23 (2008/09), no. 1, 407-427.

[24] E. Korach, N. Solel, *Tree-width, path-width, and cutwidth.* Discrete Appl. Math. 43 (1993), no. 1, 97-101.

[25] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, C. H. Papadimitriou, *The complexity of searching a graph.* J. Assoc. Comput. Mach. 35 (1988), no. 1, 18-44.

[26] M. E. Messinger, R. J. Nowakowski, P. Prałat, *Cleaning a network with brushes.* Theoret. Comput. Sci. 399 (2008), no. 3, 191-205.

[27] T. D. Parsons, *Pursuit-evasion in a graph.* Theory and applications of graphs (Proc. Internat. Conf., Western Mich. Univ., Kalamazoo, Mich., 1976), pp. 426–441. Lecture Notes in Math., Vol. 642, Springer, Berlin, 1978.

[28] B. Randerath, P. D. Vestergaard, *Well-covered graphs and factors.* Discrete Appl. Math. 154 (2006), no. 9, 1416-1428.

[29] S. M. Ross, *Introduction to probability models.* Seventh edition. Harcourt/Academic Press, Burlington, MA, 2000. xvi+693 pp.

[30] Edited by J. R. Sack, J. Urrutia, *Handbook of computational geometry.* North-Holland, Amsterdam, 2000. x+1027+I48 pp.

[31] D. Stanley, B. Yang, *Fast searching games on graphs.* J. Comb. Optim. 22 (2011), no. 4, 763-777.

[32] B. Yang, *Fast edge-searching and related problems.* Combinatorial optimization and applications. Part II, 228–242, Lecture Notes in Comput. Sci., 6509, Springer, Berlin, 2010.