# On The Paranoid Watchman Problem

by

Dominik W. Brugger

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in

Mathematics

at

The University of Wisconsin–Milwaukee

May 2009

# On The Paranoid Watchman Problem

by

Dominik W. Brugger

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in

Mathematics

at

The University of Wisconsin–Milwaukee

May 2009

---

Major Professor Date

---

Co-Chair Date

---

Graduate School Approval Date

# Abstract

# On The Paranoid Watchman Problem

by

Dominik W. Brugger

The University of Wisconsin–Milwaukee, 2009
Under the Supervision of Professor Christine Cheng and Professor Jeb Willenbring

The *Paranoid Watchman Problem* was introduced by Dalzell et al. as a search problem on a graph $G$. The *watchman number* $w(G)$ is the minimum number of watchmen with a visibility radius of 1 that is required to guarantee the capture of an invisible, arbitrarily fast and clairvoyant intruder who is trying to avoid capture. By discretizing the problem, we show that $w(G)$ and the *domination search number* of $G$, $ds(G)$, differ by at most 1. This implies that calculating and approximating the watchman number is NP-hard, and various lower and upper bounds on the domination search number are inherited. We then present several counterexamples to a previous characterization by Dalzell et al. of graphs $G$ with $w(G) = 1$ but prove that *interval graphs* and certain graphs that are spanned by a *caterpillar* tree are a subset thereof. Then we show that circular-arc graphs have a watchman number of at most 2. Further, we give an upper bound on the watchman number of a tree that has the same order as its radius.

Major Professor Date

# Table of Contents

# List of Figures

# Acknowledgements

First, I express my sincere gratitude to my advisor Professor Christine Cheng for her excellent guidance and assistance during my whole year at the University of Wisconsin-Milwaukee and especially throughout this thesis project.

I also thank Professors Jeb Willenbring and Richard Stockbridge for serving on my committee and Professor Matt Walsh for several discussions.

Finally, I am very grateful for the support of my family and friends.

Seit man begonnen hat, die einfachsten Behauptungen zu beweisen, erwiesen sich viele von ihnen als falsch.

*Bertrand Russell*

# Chapter 1

# Introduction

The theory of graph searching evolved over the last four decades. In 1967, Breisch [4], a spelunker, was the first to study the problem of finding a lost person in a system of dark caves. The goal is to locate the person using the minimum number of searchers independent of how the person is moving, even if he wants to *avoid* being rescued.

In 1976, Parsons [13] gave the first mathematical formulation of the problem. He described the *pursuit-evasion* problem as a game between searchers and an *evader* (the lost man) on a finite connected graph $G = (V, E)$. The movement of each person, searcher or evader, is modeled as a function $f : [0, \infty) \to V \cup E$. The evader is assumed to be capable of moving arbitrarily fast and has full information about the searchers' movements. The evader is caught by one of the searchers if they are at the same vertex or edge at the same time; i.e., their corresponding functions have the same value for some $t_0 \in [0, \infty)$.

Some time later, Golovach [9] described the *edge search* problem, a one-person game on a graph $G$ that is discrete in nature. Initially, all the edges of the graph are contaminated by some poisonous gas. At each time step, one of the following movements is made: a searcher is placed on a vertex, a searcher is removed from a vertex, or a searcher is moved across an edge. An edge $e = \{u, v\}$ is *cleared* either by (i) placing two searchers at vertex $u$, and then moving one of them across $e$ to vertex $v$, or by (ii) sliding a searcher from $u$ to $v$ provided all the edges incident to $u$ are already cleared. The goal is to clear all the edges of the graph. Golovach showed that the edge search problem is equivalent to Parsons' pursuit-evasion problem.

By now, many graph search problems have been considered in the literature. The above models capture how these problems are typically presented. Either they are *continuous* two-person games where one party is in charge of a set of searchers and the other is in charge of an evader or several evaders, and they are trying to outwit each other; and/or they are *discrete* one-person games where the graph is initially contaminated and the goal is to clear the entire graph using a set of allowed moves. The differences in these problems are often due to the capabilities of the searchers and the evaders, and the movements that can clear a vertex or an edge of a graph. A main parameter of interest is the *graph search number*[1], the minimum number of searchers needed to catch an evader or clear the graph. We refer readers to Fomin and Thilikos' annotated bibliography [7] and Alspach's paper [2] for comprehensive surveys on the field of graph searching.

In this thesis, we consider the *paranoid watchman problem* that was first introduced by Dalzell et al. [5]. A group of watchmen are patrolling a graph $G = (V, E)$ to catch an intruder. The movement of each person, watchman or intruder, is described by a function $f : [0, \infty) \to V$. He behaves in such a way that the amount of time he spends on an edge is instantaneous, and the sequence of vertices he visits forms a walk on $G$. Additionally, a watchman has a vision radius of one; he can see the vertex he is currently occupying and the vertex's immediate neighborhood. On the other hand, the intruder is assumed to be powerful; he sees all the watchmen and knows about their future movements. An intruder is caught if at some time $t_0 \in [0, \infty)$, there is a watchman that can see the intruder because the vertices the intruder and the watchman are occupying are the same or adjacent.

We begin Chapter 2 with the *domination search problem* that is due to Fomin et al. [6]. It is a discrete one-person game where each searcher, like a watchman, also has a vision radius of one. Next, we formally describe the paranoid watchman

[1]Whenever the graph search problem has a name, we shall refer to this number as the said problem's number – e.g., the edge search number, etc.

problem and discretize it using the domination search problem's formulation as a template.

The discrete formulation of the paranoid watchman problem enables us to directly compare the watchman number and the domination search number of a graph. In Chapter 3, we prove that these numbers differ by at most 1. This result has several direct implications. We therefore list several results about the domination search number of a graph. Various upper bounds on the domination search number from [6] immediately translate to the watchman number. More importantly, it allows us to show that approximating the watchman number of a graph to within a logarithmic factor is NP-hard.

In Chapter 4, we consider the problem of characterizing 1-watchable graphs, which are graphs with watchman number equal to 1. In their paper [5], Dalzell et al. introduced the notion of the *dominant subgraph* $G^\star$ of a graph $G$. They noted that if a single watchman can clear $G$ then that watchman's walk can be restricted to the vertices of $G^\star$. They then went on to argue that $G$ is 1-watchable if and only if $G^\star$ is a connected *interval graph*. We show that this characterization is false in both directions. To argue the necessity direction, Dalzell et al. claimed that the dominant subgraph of a 1-watchable graph must be *chordal* and *asteroidal triple-free*; that is, the largest induced cycle of the graph is a 3-cycle and it contains no three vertices such that any two of them are connected by a path that avoids the neighborhood of the third. To disprove this claim, we show that for every positive integer $k \geq 4$, there is a graph with an induced cycle of length $k \geq 4$ that is 1-watchable. In other words, there are 1-watchable graphs with very large cycles. On the other hand, when $G^\star$ is a connected interval graph, they presented an algorithm that constructed a watchman's walk which they claim clears the whole graph. We show that there is a graph which satisfies the input of their algorithm, and yet the watchman's walk that is created from their algorithm does not clear the graph.

Worse, this graph's watchman number is greater than 1. Hence, our counterexample shows that the property that $G^\star$ is a connected interval graph does not a guarantee that $G$ is 1-watchable. On the bright side, we are able to salvage a few results from their characterization – for example, we prove that if $G$ is a connected interval graph, then $G$ is indeed 1-watchable.

In Chapter 5, we present some results on the watchman number of some graph classes. A natural generalization of interval graphs are *circular-arc graphs*. We show that the watchman number of such graphs, provided they are connected, are at most 2. Another interesting class we consider are graphs with spanning trees that are *caterpillars*. We provide upper bounds on the watchman numbers of such graphs. Finally, we also show that the watchman number of a tree is bounded by the order of its radius. We conclude in Chapter 6.

# Chapter 2

# Search Programs

In this chapter we first give the definition of *domination search* according to [6] and the continuous two-person game formulation of the paranoid watchman problem as introduced in [5]. We then give a discrete one-person game formulation and show why it is appropriate. To illustrate the concept we work through an example graph and then discuss the notion of *monotonicity*.

## 2.1 Domination Search

Domination search was first introduced by Fomin et al. in [6]. In this discrete graph searching variant, the goal of the game is to clear, using multiple searchers, a graph $G = (V, E)$ whose vertices were all declared *contaminated* in the beginning. Movements alternate between *placing searchers* on some vertices of $G$ or *removing searchers* from them. When placing searchers on a vertex $v \in V$, the vertex itself and its neighborhood $N(v)$ get *cleared*. On the other hand, when removing a searcher from a vertex $v$, previously cleared vertices might get *recontaminated* if the intruder has the chance to get there from a contaminated vertex.

Let us now define domination search formally. To simplify notation we write $N[V'] = \bigcup_{v \in V'} N[v]$ for a subset of vertices $V' \subseteq V$.

**Definition 1** *A domination search program* $\Pi_{ds}$ *on a graph* $G = (V, E)$ *is a sequence* *of pairs (also considered as the steps of* $\Pi_{ds}$*)*

$$\Pi_{ds} = (D_0, A_0), (D_1, A_1), \ldots, (D_{2m-1}, A_{2m-1})$$

*such that*

1. $D_i \subseteq V$ *and* $A_i \subseteq V$ *for all* $i = 0, 1, \ldots 2m-1$ *where* $D_i$ *is the multiset of vertices containing a searcher, and* $A_i$ *is the set of cleared vertices at step* $i$

2. $D_0 = \emptyset$, $A_0 = \emptyset$

3. *At the* $(2i-1)$*-th step we place new searchers and clear vertices:* $D_{2i-2} \subset D_{2i-1}$ *and* $A_{2i-1} = A_{2i-2} \cup N[D_{2i-1}]$ *for every* $i = 1, 2, \ldots m$.

4. *At the* $2i$*-th step we remove searchers and possibly recontaminate vertices:* $D_{2i-1} \supset D_{2i}$ *and* $A_{2i}$ *is the subset of* $A_{2i-1}$ *such that whenever* $v \in A_{2i}$ *every path from* $v$ *to a vertex from* $V - A_{2i-1}$ *contains a vertex in* $N[D_{2i}]$.

*A domination search program is winning if all vertices are cleared in the end, i.e. if* $A_{2m-1} = V$. *The domination search number* $ds(G)$ *is the minimum number* $k$ *such that there exists a winning domination search program using at most* $k$ *searchers at any given time.*

## 2.2 The Watchman Search Program

### 2.2.1 Continuous Formulation

*The Paranoid Watchman Problem* was first introduced by Dalzell et al. [5] as a continuous two-person game on a graph $G = (V, E)$. The game is played by two parties, a group of $k$ watchmen and an intruder. The goal of the watchmen is to catch an intruder who is trying to escape detection.

To describe the movement of each person, watchman or intruder, we first define a general *movement function* as a function $f$ of the form

$$f : [0, \infty) \to V \cup E$$

with $f(t)$ identifying the location of the person at time $t \in [0, \infty)$. The movement of a person from one vertex to an adjacent one is assumed to occur instantaneously, i.e. for all edges $e \in E$ we have $f^{-1}(e) = \{t_j\}$ for some $t_j \in [0, \infty)$ while he resides on a vertex over a period of time of positive length, i.e. for all vertices $v \in V$ we have $f^{-1}(v) = \bigcup_j (a_j, b_j)$ for some $a_j < b_j \in [0, \infty)$. Further, jumps within the graph are not allowed, thus for every $t_0 \in [0, \infty)$ such that $f(t_0) = \{u, v\} \in E$ there exists an $\varepsilon > 0$ with $f(t) = u$ for $t \in (t_0 - \varepsilon, t_0)$ and $f(t) = v$ for $t \in (t_0, t_0 + \varepsilon)$ or vice versa.

We shall refer to the movement functions of the $k$ watchmen as $f_1, f_2, \ldots f_k$ and that of the intruder as $g$.

The two parties posses the following capabilities: Every watchman has a radius of visibility of one, i.e. while residing at a vertex $v \in V$ he is also able to monitor the immediate neighborhood $N(v)$ of $v$. The intruder has full information about the watchmen, i.e. he knows $f_1, f_2, \ldots f_k$ so that at any time he knows their positions and where and when they plan to move next.

The intruder is *caught* if he ends up in the closed neighborhood of at least one watchman, i.e. if

$$\underbrace{g(t)}_{\in V} \in N[\underbrace{f_i(t)}_{\in V}] \quad \text{for some } t \in [0, \infty) \text{ and some } i = 1, \ldots, k.$$

The graph $G$ is said to be *$k$-watchable* if there exist watchman functions $f_1, \ldots, f_k$ such that for any intruder function $g$ the intruder is caught by one of the watchmen. The *watchman number* of $G$ is defined as the minimum number $k$ such that $G$ is

$k$-watchable.

### 2.2.2 Discrete Model

We now give an alternate formulation of the watchman problem as a discrete model. This has several advantages: First, it is easier to compare the discrete model with other well-established graph search models that are also discrete. Second, this version allows us to run some computer simulations that would be more difficult for a continuous model.

**Definition 2** *A $k$-watchmen search program $\Pi$ on graph $G = (V, E)$ is a sequence of pairs*

$$\Pi = (W_0, A_0), (W_1, A_1), \ldots, (W_r, A_r)$$

*such that*

*(i) for $i = 0$ to $r$, $W_i = (v_1^i, v_2^i, \ldots, v_k^i)$ where $v_j^i \in V$ is the location of watchman $j$ at step $i$ for $j = 1, \ldots, k$*

*(ii) for $j = 1, \ldots, k$, $i = 0, \ldots, r-1$, either $v_j^i = v_j^{i+1}$ or $\{v_j^i, v_j^{i+1}\} \in E$ (i.e., at each step, a watchman either stays put or moves to an adjacent vertex)*

*(iii) $A_0 = \bigcup_{j=1}^{k} N[v_j^0]$ (i.e., the set of cleared vertices at the end of step $0$ is the union of the closed neighborhoods of the vertices containing a watchman at step $0$)*

*(iv) for $i = 1$ to $r$, $A_i = \tilde{A}_{i-1} \cup \bigcup_{j=1}^{k} N[v_j^i]$ where $\tilde{A}_{i-1} \subseteq A_{i-1}$ such that every path from a vertex in $V - A_{i-1}$ to a vertex in $\tilde{A}_{i-1}$ contains a non-starting vertex in $\bigcup_{j=1}^{k} N[v_j^i]$ (i.e., $\tilde{A}_{i-1}$ is the set of cleared vertices at the end of step $i-1$ that are "protected" by the $k$ watchmen at step $i$. A vertex $u \in \tilde{A}_{i-1}$ is protected because any*

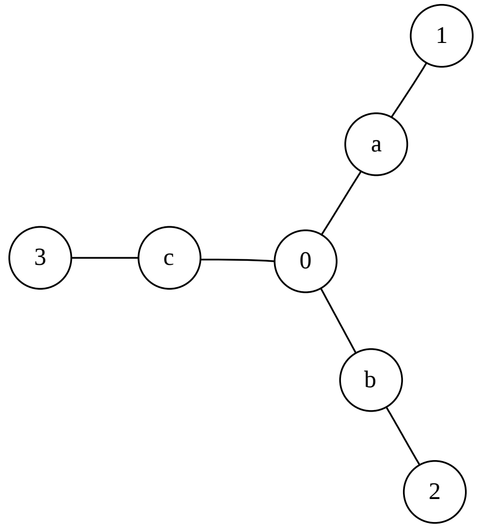


Figure 2.1: Example of a 1-watchable graph

*path from a contaminated node $w$ at the end of step $i-1$ to $u$ has to pass through the neighborhood of some $v_j^i$.)*

*We say that $\Pi$ is a winning $k$-watchmen search program if $A_r = V$.*

In the next subsection, we shall show that this discrete model captures the continuous paranoid watchman model.

In order to internalize the definition of the watchman search program, let us consider the following example.

**Example 1** *Let $G = (V, E)$ denote the graph as illustrated in Figure 2.1. Then we can define a winning 1-watchman search program by*

$$\begin{aligned}\Pi \ := \ & ((a), \{0, a, 1\}), ((0), \{0, a, 1, b, c\}), ((b), \{0, a, 1, b, 2\}), \\ & ((0), \{0, a, 1, b, 2, c\}), ((c), \underbrace{\{0, a, 1, b, 2, c, 3\}}_{=V}).\end{aligned}$$

*The movement of the watchman is illustrated in Figure 2.2. At step* $0$ *the watchman is placed at vertex* $a$ *and its closed neighborhood* $\{0, a, 1\}$ *is cleared. When moving to* $0$ *at step* $1$*, the neighborhood of* $0$ *is cleared so* $b$ *and* $c$ *are added to* $A_1$*. At this step no recontamination occurs. At step* $2$ *we move the watchman to* $b$ *and clear* $2$ *but* $c$ *gets recontaminated as there is a path from the contaminated vertex* $3$ *to* $c$ *that does not intersect with* $N[b] = \{0, b, 0\}$*. Then we move the watchman back to* $0$ *at step* $4$*, clearing* $c$ *while protecting all of the previously cleared vertices. Finally, at step* $4$*, the watchman moves to* $c$*, clearing vertex* $3$ *that was the only contaminated vertex left.*

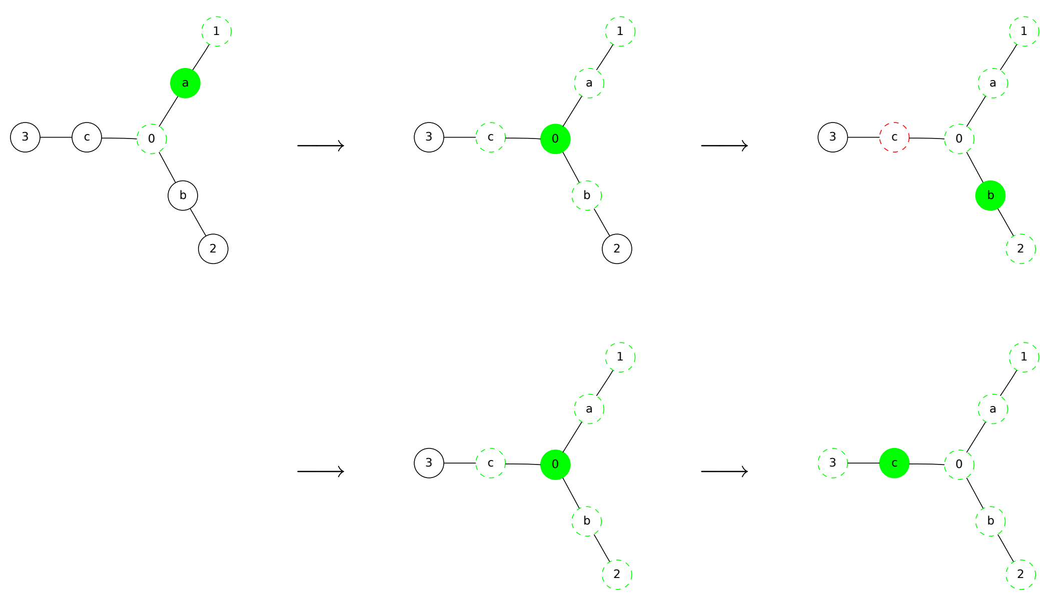


Figure 2.2: A winning 1-watchman program

## 2.3 Discretizing the Paranoid Watchman Problem

Let $f_1, f_2, \ldots, f_k$ be the movement functions of $k$ watchmen in graph $G = (V, E)$. Set $t_0 = 0$, and recursively define $t_i$ as the first time after $t_{i-1}$ that some watchman moved. Hence, for watchman $j$, $f_j(t_i)$ is either a vertex or an edge and $f_j(t)$ is fixed throughout the interval $(t_i, t_{i+1})$ for $i \in \{0\} \cup \mathbf{Z}^+$. Thus, we can think of the time line as being partitioned into intervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{i-1}, t_i), [t_i, t_{i+1}), \ldots$ as illustrated in Figure 2.3.

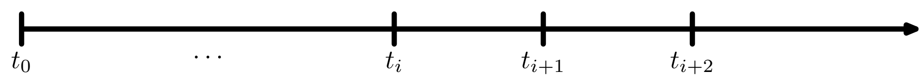


Figure 2.3: Time line

For each $j$, create another movement function $f'_j$ from $f_j$ so that the watchman's movement in the interval $[i, i+1)$ is a time-scaled version of his movement under $f_j$ in the interval $[t_i, t_{i+1})$, for $i \in \{0\} \cup \mathbf{Z}^+$. In other words,

$$f'_j(t') = f_j(t) \text{ where } t' - i = \frac{t - t_i}{t_{i+1} - t_i}.$$

**Claim 1** *The $k$ watchmen using $f_1, \ldots, f_k$ can catch any intruder in $G$ if and only if they can do so using $f'_1, \ldots, f'_k$.*

**Proof** Let $g$ be the movement function of an intruder in $G$. Define another intruder movement function $g'$ so that his movement in the interval $[i, i+1)$ is also a time-scaled version of his movement under $g$ in the interval $[t_i, t_{i+1})$, $i \in \{0\} \cup \mathbf{Z}^+$. Notice that $k$ watchmen using $f_1, \ldots, f_k$ cannot catch an intruder using $g$ if and only if $k$ watchmen using $f'_1, \ldots, f'_k$ cannot catch an intruder using $g'$. The lemma follows immediately. $\square$

Given $f_1, f_2, \ldots, f_k$, let $A(t)$ denote the set of vertices in $G$ that the watchmen are sure to be intruder-free at time $t$. That is, if $v \notin A(t)$, then $v$ is a potential hiding place for an intruder because there is some intruder function $g$ so that an intruder using $g$ is not caught by any of the watchmen by time $t$ and $g(t) = v$. Similarly, define $A(t^-)$ and $A(t^+)$ as the set of vertices in $G$ that the watchmen are sure to be intruder-free just prior to time $t$ and just after time $t$ respectively.

Suppose there exists $t_i$ and $t_{i'}$, $i < i'$ so that the positions of each watchman is the same in the intervals $(t_i, t_{i+1})$ and $(t_{i'}, t_{i'+1})$ and $A(t_{i+1}^-) = A(t_{i'+1}^-) \neq V$. For each $j$ create another movement function $f_j'$ from $f_j$ by omitting the portion from $[t_{i+1}, t_{i'+1})$. That is, set $f_j'(t)$ to $f_j(t)$ when $t \in [0, t_{i+1})$ and to $f_j(t + t_{i'+1} - t_{i+1})$ when $t \in [t_{i+1}, \infty)$.

**Claim 2** *The $k$ watchmen using $f_1, f_2, \ldots, f_k$ can catch any intruder in $G$ if and only if they can do so using $f_1', f_2', \ldots, f_k'$.*

**Proof**

Suppose an intruder in $G$ is moving according to $g$, and $k$ watchmen using $f_1, \ldots, f_k$ cannot capture this intruder. Let $v$ denote the location of the intruder just prior to $t_{i'+1}$; that is, if $g(t_{i'+1})$ is a vertex then $v = g(t_{i'+1})$ and if $g(t_{i'+1})$ is an edge then $v$ is the vertex which the intruder is moving from. In either case, $v \notin A(t_{i'+1}^-)$ because the watchmen cannot catch the intruder. Since $A(t_{i+1}^-) = A(t_{i'+1}^-)$, there is another intruder movement function $g'$ so that the intruder eludes capture by the watchmen using $f_1, \ldots, f_k$, and the intruder is at vertex $v$ just prior to $t_{i+1}$. Define another intruder function $g''$ as follows: let $g''(t) = g'(t)$ in the interval $[0, t_{i+1})$, and let $g''(t) = g(t + t_{i'+1} - t_{i+1})$ in the interval $[t_{i+1}, \infty)$. Since $f_1', \ldots, f_k'$ was obtained from $f_1, \ldots, f_k$ by simply omitting the $[t_{i+1}, t_{i'+1})$-portion of these functions, it is now easy to see that $k$ watchmen using $f_1', \ldots, f_k'$ cannot capture an intruder using $g''$.

On the other hand, suppose some intruder is moving according to some function

$h$, and $k$ watchmen using $f'_1, \ldots, f'_k$ cannot capture this intruder. As in the previous paragraph, let $u$ denote the location of the intruder just prior to $t_{i+1}$. Since the functions $f'_1, \ldots, f'_k$ behave exactly like $f_1, \ldots, f_k$ in the interval $[0, t_{i+1})$, $u \notin A(t^-_{i+1})$. But $A(t^-_{i+1}) = A(t^-_{i'+1})$ so there is an intruder function $h'$ so that the intruder using $h'$ eludes capture by the watchmen using $f_1, \ldots, f_k$, and the intruder is at vertex $u$ just prior to $t_{i'+1}$. Then consider the intruder function $h''$ so that $h''(t) = h'(t)$ in the interval $[0, t_{i'+1})$ and $h''(t) = h(t - (t_{i'+1} - t_{i'+1}))$ in the interval $[t_{i'+1}, \infty)$. Again, it is easy to verify that an intruder using $h''$ cannot be captured by $k$ watchmen using $f_1, f_2, \ldots, f_k$. $\square$

Let us denote a movement function on $G$ as *integer-stepped* if and only if a watchman moves at integer time steps.

**Theorem 1** *Let $G = (V, E)$ be a graph with $n$ vertices. Then $G$ is $k$-watchable if and only if there exists $k$ integer-stepped movement functions on $G$ so that by time $T = n^k 2^n + 1$, any intruder in $G$ is captured by $k$ watchmen using these functions.*

**Proof** If $k$ watchmen can catch any intruder in $G$ by time $T$, $G$ is clearly $k$-watchable. So let us assume the converse – $G$ is $k$-watchable. Hence, there exists $k$ movement functions $f_1, f_2, \ldots, f_k$ so that they can capture any intruder in $G$. According to Claim 1, these functions can be made integer-stepped without compromising the watchmen's ability to capture intruders in $G$. Furthermore, according to Claim 2, they can also be trimmed so that whenever each watchman is occupying exactly the same position at intervals $(i, i+1)$ and $(i', i'+1)$ with $i' > i$, then $A((i+1)^-) \neq A((i'+1)^-)$. Now, there are $n^k$ different sequences of vertices that the watchmen can occupy in an interval $(i, i+1)$. There are also at most $2^n$ possible sets for $A((i+1)^-)$. So there are at most $n^k 2^n$ distinct $(i, i+1) - A((i+1)^-)$ pairs. Hence, once the $k$ watchmen have gone through $n^k 2^n$ different intervals, they must surely have captured the intruder. $\square$

**Theorem 2** *$w(G) \leq k$ if and only if there exist $k$ watchman movement functions $f_1, f_2, \ldots, f_k$ such that for some time $t < \infty$, the set of vertices the watchmen know for certain are intruder free at time $t$ is $V$; i.e. $A(t) = V$.*

**Proof** '⇒': According to Theorem 1, $A(n^k 2^n + 1) = V$.
'⇐': Suppose $\exists$ $k$ watchman movement functions $f_1, f_2, \ldots, f_k$ such that for some time $t < \infty$, $A(t) = V$. Let $g$ by some intruder function. If the intruder has not been caught by time $t$, then the vertices occupied by the intruder at time $t' \geq t$ are not intruder-free. But this contradicts the assumption that $A(t) = V$. □

Let us now convert the continuous model to the discrete model.

**Theorem 3** *Let $f_1, f_2, \ldots, f_k$ be the movement functions of $k$ watchmen on graph $G$. If there exists $t < \infty$ such that $A(t) = V$ then there is a winning discrete $k$-watchman search program* $\Pi$.

**Proof** Again, set $t_0 = 0$ and let $t_{i+1}$ be the first time after $t_i$ that one of the $k$ watchmen moves. Hence, as before, we think of the time line as being partitioned into intervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{i-1}, t_i), [t_i, t_{i+1}), \ldots$ Once the watchmen have settled into their respective vertices at $t_i$, the intruder is free to roam the graph.

First, let us investigate what happens in the continuous model when the watchmen move from their vertices $v_j^i$ to $v_j^{i+1}$ at time $t_{i+1}$. Then the vertices $\bigcup_j N\left[v_j^{i+1}\right]$ are guaranteed to be intruder-free. As the intruder can only move from one vertex to an adjacent one at time $t_i$, vertices on the border to the not intruder-free area that are not in the neighborhood of the watchmen, i.e. vertices in

$$\begin{aligned} R_{\{t_{i+1}\}} &:= \left\{ v \in A(t_{i+1}^-) - \bigcup_j N\left[v_j^{i+1}\right] \middle| \exists \{u, v\} \in E, u \in V - A(t_{i+1}^-) \right\} \\ &= \left\{ v \in A(t_{i+1}^-) \middle| \exists \{u, v\} \in E, u \in V - A(t_{i+1}^-) \right\} - \bigcup_j N\left[v_j^{i+1}\right] \end{aligned}$$

are also no longer guaranteed to be intruder-free.

Now let us also consider the time frame $(t_{i+1}, t_{i+2})$, i.e. after the watchmen have settled at the vertices $v_j^{i+1}$ but before they move again. During this time, no new vertices are identified as intruder-free, but every vertex in the set

$$R_{(t_{i+1},t_{i+2})} := \left\{ v \in A(t_{i+1}^+) \,\middle|\, \exists \text{path from } u \in R_{\{t_{i+1}\}} \text{ to } v \text{ not intersecting with } \bigcup_j N\left[v_j^{i+1}\right] \right\}$$

is no longer considered intruder-free.
Therefore $A(t_{i+1}^+) = \left(A(t_{i+1}^-) \cup \bigcup_j N\left[v_j^{i+1}\right]\right) - R_{\{t_{i+1}\}}$ and $A(t_{i+2}^-) = A(t_{i+1}^+) - R_{(t_{i+1},t_{i+2})}$.

For $i \in \{0\} \cup \mathbf{Z}^+$ define $W_i = (v_1^i, v_2^i, \ldots, v_k^i) = (f_1(t_i), f_2(t_i), \ldots, f_k(t_i))$ as illustrated in Figure 2.4. Recall that in the discrete model $A_i$ is the set of cleared vertices at step $i$.

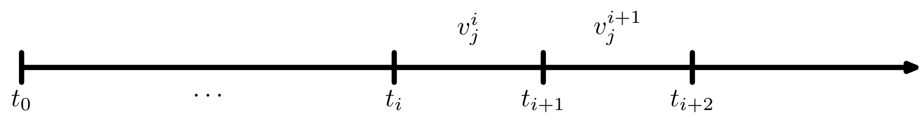


Figure 2.4: Location of watchman $j$ on the time line

**Claim 3** *For every* $i \in \{0\} \cup \mathbf{Z}^+$, $A_i = A(t_{i+1}^-)$.

**Proof** For $i = 0$, $A_0 = \bigcup_j N\left[v_j^0\right] = A(t_1^-) = A(t_{0+1}^-)$ ✓
Now assume $A_i = A(t_{i+1}^-)$ for some $i \in \{0\} \cup \mathbf{Z}^+$. We have to show: $A_{i+1} = A(t_{i+2}^-)$
*Proof:* "$\subseteq$": Let $v \in A_{i+1} = \bigcup_j N\left[v_j^{i+1}\right] \cup \tilde{A}_i$.

We need to show: $v \in A(t_{i+2}^-)$, i.e. $v \in A(t_{i+1}^+) - R_{(t_{i+1},t_{i+2})}$. If $v \in \bigcup_j N\left[v_j^{i+1}\right]$ then the statement is true. Now let $v \in \tilde{A}_i - \bigcup_j N\left[v_j^{i+1}\right]$. Then $v \in A_i - \bigcup_j N\left[v_j^{i+1}\right]$ such that every path from $u \in V - A_i$ to $v$ contains a non-starting vertex in $\bigcup_j N\left[v_j^{i+1}\right]$.

Assume $v \in R_{\{t_{i+1}\}}$, then as $A_i = A(t_{i+1}^-)$, there exists $u \in V - A_i$ such that there is an edge $\{u, v\} \in E$.

Therefore $v \in \bigcup_j N\left[v_j^{i+1}\right]$ as $v$ is the only non-starting vertex in this $(u,v)$-path. This is a contradiction to $v \in \tilde{A}_i - \bigcup_j N\left[v_j^{i+1}\right]$. Thus $v \notin R_{\{t_{i+1}\}}$ and therefore $v \in A(t_{i+1}^+)$.

We still need to show that $v \notin R_{(t_{i+1},t_{i+2})}$. Therefore, assume to the contrary that there exists a path $P$ from $u \in R_{\{t_{i+1}\}}$ to $v$ not intersecting with $\bigcup_j N\left[v_j^{i+1}\right]$. Then there exists $\tilde{u} \in V - A_i$ with $\{u, \tilde{u}\} \in E$ (otherwise $u$ would not be in $R_{\{t_{i+1}\}}$). Then the path $\tilde{P} = \tilde{u}P$ from $\tilde{u}$ to $v$ does not contain a non-starting vertex in $\bigcup_j N\left[v_j^{i+1}\right]$ which is again a contradiction to the assumption that $v \in \tilde{A}_i - \bigcup_j N\left[v_j^{i+1}\right]$. Therefore $v$ must *not* be in $R_{(t_{i+1},t_{i+2})}$ and we proved that $A_{i+1} \subseteq A(t_{i+2}^-)$.

Now let us argue the other inclusion.

"$\supseteq$": Let $v \in A(t_{i+2}^-)$. Then $v \in A(t_{i+1}^+) - R_{(t_{i+1},t_{i+2})}$. We need to show that $v \in A_{i+1} = \bigcup_j N\left[v_j^{i+1}\right] \cup \tilde{A}_i$. Let us assume the contrary, then $v \notin \bigcup_j N\left[v_j^{i+1}\right]$ and $v \notin \tilde{A}_i$. Thus $v \in A(t_{i+1}^-) = A_i$ and $v$ was recontaminated, i.e. there exists $u \in V - A_i$ and a path $P$ from $u$ to $v$ that does not contain a non-starting vertex in $\bigcup_j N\left[v_j^{i+1}\right]$. Then there exists $\tilde{v}_2 \in P \cap (V - A_i)$ with $d_G(\tilde{v}_2, A_i - \tilde{A}_i) = 1$ and there also exists $\tilde{v}_1 \in P \cap \left(A_i - \tilde{A}_i\right)$ such that $\{\tilde{v}_1, \tilde{v}_2\} \in E$. Then $\tilde{v}_1 \in R_{\{t_{i+1}\}}$ as $A_i = A(t_{i+1}^i)$ and there exists a path from $v$ to $\tilde{v}_1$ not intersecting with $\bigcup_j N\left[v_j^{i+1}\right]$, in contradiction to $v \notin R_{(t_{i+1},t_{i+2})}$. Therefore, $v$ must be in $A(t_{i+2}^-)$ and we also proved $A_i \supseteq A(t_{i+1}^-)$. Together with $A_i \subseteq A(t_{i+1}^-)$ we have the equality $A_i = A(t_{i+1}^-)$ and the claim follows by induction. $\square$

On the other hand, we can also convert the discrete model to a continuous version.

**Theorem 4** *Suppose* $\Pi$ *is a winning* $k$*-watchman search program. Then there exist movement functions* $f_1, f_2, \ldots f_k$ *such that for some time* $t < \infty$, $A(t) = V$.

**Proof** Assume we are given $\Pi$ by $\Pi = (W_0, A_0), (W_1, A_1), \ldots, (W_r, A_r)$ with $W_i = (v_1^i, v_2^i, \ldots, v_k^i)$ for $i = 0, 1, \ldots r$. The corresponding continuous model can be constructed as follows.

Define the $k$ watchmen's movement functions $f_1, f_2, \ldots, f_k : [0, \infty) \to V \cup E$ by

$$f_j(t) = \begin{cases} v_j^0 & t = 0 \\ v_j^i & t \in (i, i+1), \quad i = 0, 1, \ldots r-1 \\ v_j^r & t > r \\ \{v_j^{i-1}, v_j^i\} & t = i, \quad i = 1, 2, \ldots r \end{cases}$$

Then as above, it can be proved similarly that for every $i \in \{0\} \cup \mathbf{Z}^+$, $A_i = A((i+1)^-)$. As $\Pi$ is winning, $A_r = V$ and so is $A(r+1) = V$. $\square$

Hence, according to Theorem 2, 3 and 4, $w(G) \leq k$ if and only if there exists a winning discrete $k$-watchman search program. We can now restrict our discussion to the discrete version and, in line with the continuous one, we denote the *watchman number of* $G$ by $w(G)$, as the smallest integer $k$ so that $G$ has a winning $k$-watchman search program. A graph $G$ is said to be $k$-watchable if there is a winning $k$-watchman search program for $G$.

## 2.4 Monotonicity

**Definition 3** *A discrete graph search program* $\Pi$ *is called monotone if the sequence of cleared vertices* $A_0, A_1, \ldots, A_r$ *is monotone, i.e. if* $A_i \subseteq A_{i+1}$ *for* $i = 0, \ldots, r-1$. *In other words, recontamination does not occur.*

For the edge search problem, LaPaugh [10], and later Bienstock and Seymour [3], showed that for every graph $G$ with edge search number $k$, there is always a *monotone* strategy that clears the graph using $k$ searchers. As every edge needs to be cleared only once, this implies that such a strategy requires $\mathcal{O}(n)$ steps. This allowed Megiddo et al.[12] to prove that determining if the edge search number of a graph is at most a certain integer is NP-complete.

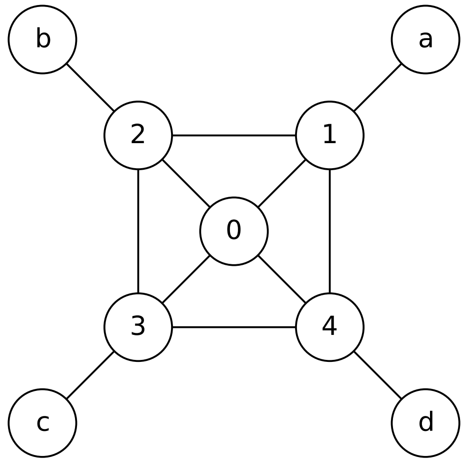


Figure 2.5: 1-watchable graph $G$ where recontamination cannot be avoided

| i | $W_i$ | $A_i$ | recontaminated |
|---|---|---|---|
| 0 | (1) | $\{0, 1, 2, 4, a\}$ | {} |
| 1 | (0) | $\{0, 1, 2, 3, 4, a\}$ | {} |
| 2 | (2) | $\{0, 1, 2, 3, a, b\}$ | {4} |
| 3 | (0) | $\{0, 1, 2, 3, 4, a, b\}$ | {} |
| 4 | (3) | $\{0, 1, 2, 3, 4, a, b, c\}$ | {} |
| 5 | (4) | $\{0, 1, 2, 3, 4, a, b, c, d\}$ | {} |

Table 2.1: Winning watchman program for Figure 2.5

We can ask the same question for the paranoid watchman problem. Assume that $G = (V, E)$ is $k$-watchable. *Is there a winning monotone $k$-watchmen search program?* The answer turns out to be *No*.

We demonstrate it using the example in Figure 2.1. In order to clear this graph using only one watchman, it is unavoidable to visit the central vertex 0, as all of the vertices $a$, $b$ and $c$ need to be visited to clear the other vertices 1, 2 and 3. Afterwards we need to enter one of the branches, let's say, without loss of generality, we visit $a$ after 0. Then the vertices $b$ and $c$ get recontaminated.

Figure 2.5 shows another example, where a winning 1-watchman search program

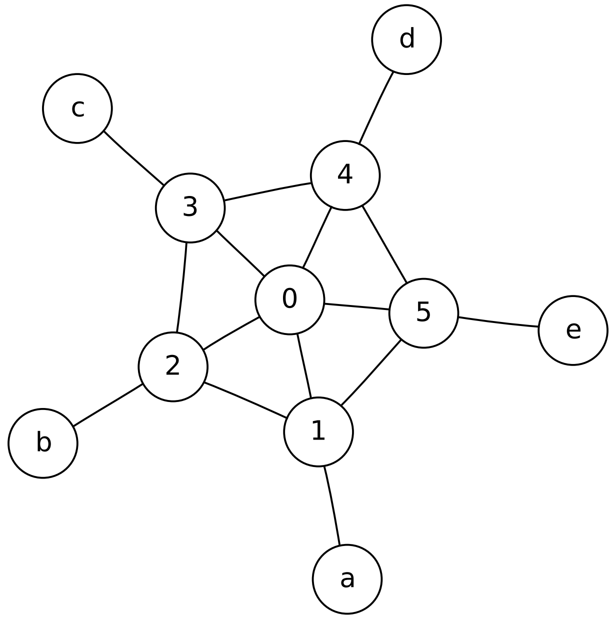


Figure 2.6: 1-watchable graph $G$ where recontamination cannot be avoided

is described in Table 2.1. In both cases, the recontamination of the vertices can be thought of as being "accidental"; at time $t$ when they were cleared, the intent of the programs were to clear other nodes. The programs were going to clear them at some later time $t' > t$.

This is no longer the case if we use a 5-cycle instead of the 4-cycle as seen in Figure 2.6. All vertices on the cycle need to be visited since they all have adjacent leaves. Exploiting symmetry, we can assume that vertex 1 is visited before its non-adjacent vertices 3 and 4. Suppose 3 is visited after 1. At that point, 1 gets recontaminated because there is a path from 4 to 1 that does not intersect the neighborhood of 3. A possible winning 1-watchman search program is described in Table 2.2.

In domination search, all of the three examples above *can* be cleared using 2 searchers without causing any vertex to get recontaminated. Place a searcher on the central vertex, and use another one to clear the leaves. This does not mean that

| i | $W_i$ | $A_i$ | recontaminated |
|---|---|---|---|
| 0 | (5) | $\{0, 1, 4, 5, e\}$ | $\{\}$ |
| 1 | (1) | $\{0, 1, 2, 5, a, e\}$ | $\{4\}$ |
| 2 | (0) | $\{0, 1, 2, 3, 4, 5, a, e\}$ | $\{\}$ |
| 3 | (3) | $\{0, 1, 2, 3, 4, 5, a, c, e\}$ | $\{\}$ |
| 4 | (4) | $\{0, 3, 4, 5, c, d, e\}$ | $\{1, 2, a\}$ |
| 5 | (0) | $\{0, 1, 2, 3, 4, 5, c, d, e\}$ | $\{\}$ |
| 6 | (2) | $\{0, 1, 2, 3, 4, 5, b, c, d, e\}$ | $\{\}$ |
| 7 | (1) | $\{0, 1, 2, 3, 4, 5, a, b, c, d, e\}$ | $\{\}$ |

Table 2.2: Winning 1-watchman search program for Figure 2.6

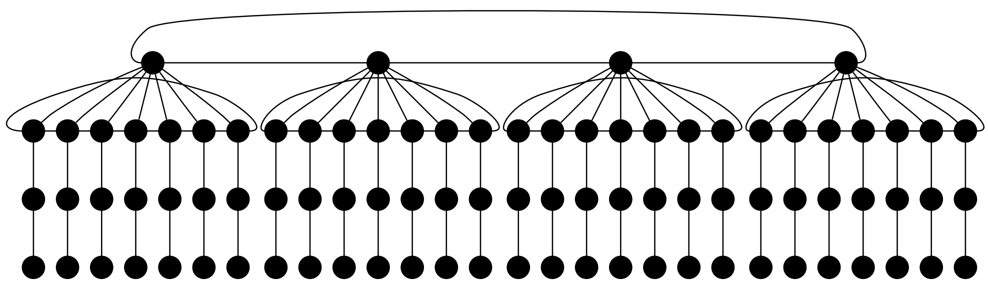

Figure 2.7: The Dobrev graph

there is always a winning monotone $ds(G)$-domination search program as Fomin et al. [6] pointed out the example of the *Dobrev* graph as seen in Figure 2.7. This graph has domination search number 2 but it is not possible to use a winning *monotone* 2-domination search program.

# Chapter 3

# Domination Search and Watchman Search

In this chapter we relate the domination search number and watchman number and show that they differ by at most 1. This allows us to directly transfer bounds on the domination search number for some graph families to the watchman number. It also implies that determining the watchman number as well as approximating it is NP-hard.

## 3.1 The Relationship between Watchman Search and Domination Search

There is a very close relationship between the watchman number and the domination search number as the following two theorems will show. The first one was shown to be true by Dalzell et al. in [5]. We reprove it here for completeness.

**Theorem 5 ([5])** *For every connected graph $G$, $w(G) \leq ds(G)$.*

To simplify notation we treat a sequence $W = (v_1, v_2, \ldots, v_l)$ just like a multi-set $\{v_1, v_2, \ldots, v_l\}$ with multiplicity function $\mathbf{1}_W$ given by $\mathbf{1}_W(x) = \sum_{i=1}^{l} \mathbf{1}_{\{v_i\}}$.

**Proof** Assume $ds(G) = k$ and let $\Pi_{ds} = (D_0, A_0), (D_1, A_1), \ldots, (D_{2m-1}, A_{2m-1})$ denote a winning $k$-domination search program for $G$. Now construct a $k$-watchman search program $\Pi = (W_0, C_0), (W_1, C_1), \ldots, (W_r, C_r)$.

| domination search | $D_0$ $\emptyset$ | $D_1$ $+$ | $D_2$ $-$ | $D_3$ $+$ | $D_4$ $-$ | $D_5$ $+$ | $\ldots$ | $D_{2m-1}$ $+$ |
|---|---|---|---|---|---|---|---|---|
| watchman search | | $W_{i_0}$ | $W_{i_0+1}W_{i_0+2}\ldots$ | $W_{i_1}$ | $W_{i_1+1}W_{i_1+2}\ldots$ | $W_{i_2}$ | $\ldots$ | $W_{i_{m-1}}$ |

Table 3.1: Watchman search program $\Pi$ for $\Pi_{ds}$

$$\begin{array}{ccccc} W_{i_{j-1}} & W_{i_{j-1}+1}, W_{i_{j-1}+2}, \ldots, W_{i_j-1} & & & W_{i_j} \\ \cup\!\mid & & \cup\!\mid & & \cup\!\mid \\ D_{2j-1} & \supseteq & D_{2j} & \subseteq & D_{2j+1} \end{array}$$

Figure 3.1: Relationship between $\Pi$ and $\Pi_{ds}$

Therefore we consider two consecutive steps $2j$ and $2j+1$ after step $2j-1$ in $\Pi_{ds}$. In the first step, searchers are removed from vertices $R := D_{2j} - D_{2j-1}$ and in the second step, searchers are placed on vertices $P := D_{2j+1} - D_{2j}$.

As illustrated in Table 3.1 and Figure 3.1, in the watchman search program these two moves are imitated in the steps $i_{j-1}+1, i_{j-1}+2, \ldots, i_j$ such that $D_{2j} \subseteq W_l$ for $l = i_{j-1}+1, i_{j-1}+2, \ldots, i_j - 1$ and $D_{2j} \subseteq D_{2j+1} \subseteq W_{i_j}$, i.e. the searchers that are not removed in step $2j$ in the domination search are always part of the watchmen group in the corresponding steps. Further, for every searcher $s$ in $D_{2j+1} - D_{2j-1}$ use an exclusive, “unused” watchman $t$ from $W_{i_{j-1}} - D_{2j+1}$ and move him along an (preferably short) $(v_t^{i_{j-1}}, s)$-path $(p_0, p_1, \ldots, p_r)$ to $s$. Thus the steps in between get specified by $\left(v_t^{i_{j-1}}, v_t^{i_{j-1}+2}, v_t^{i_{j-1}+1}, \ldots, v_t^r\right) := (p1, p2, \ldots, p_r)$ with $r \leq i_j$ and $v_t^l := v_t^r$ for $r < l \leq i_j$.

Then we need to show that $\Pi$ is a winning $k$-watchman search program:

**Claim 4** *The set of cleared vertices in the $k$-domination search after step $2j+1$ is a subset of the set of cleared vertices after step $i_j$ in the $k$-watchman search program, i.e. $A_{2j+1} \subseteq C_{i_j}$.*

**Proof** [By induction] First $A_1 = N[D_1] \subseteq N[W_0] = C_0 \checkmark$. Now assume the claim to be true for some $j$, i.e. $A_{2j+1} \subseteq C_{i_j}$. Then at step $2j+2$ in the domination search

| watchman search | | $W_0$ | $W_1$ | $\ldots$ | $W_r$ |
|---|---|---|---|---|---|
| domination search | $D_0$ | $D_1, D_2$ | $D_3, D_4$ | $\ldots$ | $D_{2r+1}, D_{2r+2}$ |

Table 3.2: Domination search program $\Pi_{ds}$ for $\Pi$

program exactly those searchers are removed that are moved during the next few steps of the watchman search, thus $A_{2j+2} \subseteq C_i$ for all $i_j \leq i < i_{j+1}$. Finally, as $W_{i_{j+1}} \supseteq D_{2j+3}$ we have $A_{2(j+1)+1} = A_{2j+3} \subseteq C_{j+1}$. $\square$

**Theorem 6** *For every connected graph $G$, $ds(G) \leq w(G) + 1$.*

**Proof** Let $G$ be a graph and $\Pi = (W_0, C_0), (W_1, C_1), \ldots, (W_r, C_r)$ be a winning $k$-watchman search program of length $r$. Without loss of generality let us assume that at every step, only one watchman is moving (if there were more than one watchmen moving, perform these movements in several consecutive steps individually). Consider the $i$-th step and assume that watchman $j(i)$ is moving from vertex $v_-^i := v_{j(i)}^{i-1}$ to $v_+^i := v_{j(i)}^i$, i.e. $W_i = (v_1^i, v_2^i, \ldots, v_k^i) = \left(v_1^{i-1}, v_2^{i-1}, \ldots, v_{j(i)}^i, \ldots, v_k^{i-1}\right)$ with $W_{i-1} = \left(v_1^{i-1}, v_2^{i-1}, \ldots, v_{j(i)}^{i-1}, \ldots, v_k^{i-1}\right)$ for $i = 1, \ldots, r$. Construct a domination search program $\Pi_{ds} = (D_0, A_0), (D_1, A_1), \ldots, (D_{2r+2}, A_{2r+2})$ of length $2r + 2$ that corresponds with $\Pi$ as shown in Table 3.2 with

1. $D_0 = \emptyset$, $A_0 = \emptyset$

2. $D_1 = D_2 = \{v_1^0, v_2^0, \ldots, v_k^0\}$

3. $D_{2i-1} = D_{2i-2} \cup \left\{v_+^{i-1}\right\}$ for $i = 2, \ldots, 2r$

   i.e. at every odd step add one searcher at the vertex that the corresponding watchman in $\Pi$ is moving to

4. $D_{2i} = D_{2i-1} - \left\{v_-^{i-1}\right\}$ for $i = 2, \ldots, 2r$

   i.e. at every even step remove one searcher from the vertex where the corresponding watchman in $\Pi$ is leaving from

Now all we have to show is that $\Pi_{ds}$ is a winning $(k+1)$-domination search program:

At step 1, $k$ searchers are placed on vertices in $G$. None is removed at step 2 and after that, one searcher is added and removed alternately, so $\max_i |D_i| = k+1$.

**Claim 5** *The set of cleared vertices in the k-watchmen search program after step i is a subset of the set of cleared vertices after step $2i+2$ in the $(k+1)$-domination search program, i.e. $C_i \subseteq A_{2i+2}$, for every $i = 0, \ldots, r$.*

**Proof** (By induction)

First, $C_0 = \bigcup_{j=1}^{k} N[v_j^0] = N[D_2] = A_2$ ✓. Now assume $C_i \subseteq A_{2i+2}$ for some $i$. Then

- $C_{i+1} = \bigcup_{j=1}^{k} N[v_j^{i+1}] \cup \tilde{C}_{i+1} = N[W_{i+1}] \cup \tilde{C}_{i+1}$ with $\tilde{C}_{i+1}$ being the set of protected vertices at step $i+1$
- $A_{2i+3} = A_{2i+2} \cup N[v_+^{i+1}]$
- $D_{2i+4} = D_{2(i+2)} = W_{i+1}$
- $A_{2i+4} \subseteq A_{2i+3}$ such that for every vertex $v \in A_{2i+4}$ every path containing $v$ and a vertex from $V - A_{2i+3}$ contains a vertex from $N[D_{2i+4}]$.

Now let $v \in C_{i+1}$. If $v \in N[W_{i+1}]$ then also $v \in A_{2i+4}$ as the same vertices are occupied in both the watchman search program and in the domination search program in step $i+1$ and $2i+4$ respectively. But if $v \notin N[W_{i+1}]$ then $v$ had been protected, i.e. $v \in C_i$ and every path from a contaminated vertex in $V - C_i$ had a non-starting vertex in $N[W_{i+1}]$. As $C_i \subseteq A_{2i+2}$, we also have $v \in A_{2i+2}$ and even $v \in A_{2i+3}$ because $v \notin N[W_{i+1}]$. Now consider a path $P$ from a contaminated vertex in $V - A_{2i+3}$ to $v$. As $V - A_{2i+3} \subseteq V - C_{i+1}$, $P$ has a (non-starting) vertex in $N[D_{2i+4}] = N[W_{i+1}]$. Thus $v \in A_{2i+4}$ and we have $C_{i+1} \subseteq A_{2(i+1)+2}$. □

Combining Theorem 5 and Theorem 6 leads to the following

**Corollary 7** *For every connected graph $G$, $w(G) \leq ds(G) \leq w(G) + 1$. In other words $w(G)$ is a 1-absolute approximation of $ds(G)$.*

Note that the inequalities in Corollary 7 are tight. To see why, consider $G$ as the simple path of length 3. Both $ds(G) = w(G) = 1$. For the latter, let $G$ be a simple path of length greater than 3. Then clearly $w(G) = 1$ but $ds(G) = 2$.

## 3.2 Implications

### 3.2.1 Complexity

When analyzing a computational problem, one fundamental question is whether it can be solved *efficiently*. Fomin et al. [6] also proved the following theorem.

**Theorem 8 ([6])** *There is a constant $c > 0$ such that there is no polynomial time algorithm to approximate the domination search number of a graph within a factor of $c \log n$ unless $\mathbf{P} = \mathbf{NP}$.*

**Theorem 9** *The problem* PARANOID WATCHMAN SEARCH*: 'Given a graph $G$ and an integer $k$, decide whether $w(G) \leq k$' is $\mathbf{NP}$-hard.*

**Proof** Let $c$ be the constant from Theorem 8 and $G$ be a graph on $n$ vertices with $\log n > \frac{2}{c}$ and let $k$ be an integer. Assume $\mathbf{P} \neq \mathbf{NP}$ and PARANOID WATCHMAN SEARCH can be answered in polynomial time. Using binary search starting at $n$ the watchman number $w(G)$ can be efficiently computed exactly. By Corollary 7, $w(G)$ is a 1-absolute approximation and hence, for large enough $n$, a ratio $c \log n > 2$ approximation of $ds(G)$ which contradicts Theorem 8. $\square$

**Theorem 10** *There is a constant $c > 0$ such that there is no polynomial time algorithm to approximate the watchman number of a graph within a factor of $c \log n$ unless $\mathbf{P} = \mathbf{NP}$.*

**Proof** Let $\tilde{c}$ be the constant from Theorem 8 and $G$ be a graph on $n$ vertices. Then assume to the contrary that for all $c > 0$ there is an *efficient* algorithm, approximating $w(G)$ by $k(c)$ within a factor of $c \log n$, i.e. $w(G) \leq k(c) \leq c \log n \; w(G)$. Then, by Corollary 7, $ds(G) - 1 \leq k(c) \leq c \log n \; ds(G)$ or equivalently $ds(G) \leq k(c) + 1 \leq c \log n \; ds(G) + 1$.

Now choose $c := \tilde{c}/2 \leq \tilde{c}\frac{\log n \; ds(G) - 1}{\log n \; ds(G)}$ then $k(c) + 1 \leq c \log n \; ds(G) + 1 \leq \tilde{c} \log n \; ds(G)$ and $k(c) + 1$ therefore is an approximation for $ds(G)$ within factor $c \log n$, contradicting Theorem 8. □

### 3.2.2 Bounds

Due to Theorem 5 the following upper bounds for on domination search number also apply to the watchman number.

#### Spanning Trees

**Definition 4** *Given a graph $G = (V, E)$, another graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$, written as $G' \subseteq G$.*

*An acyclic connected subgraph of $G$ that contains all of its vertices is called a spanning tree of $G$.*

**Definition 5** *A tree $T = (V, E)$ that contains a path $P = (v_1, v_2, \ldots, v_n)$ (called the backbone of $T$) such that any vertex $v \in V$ has distance smaller than or equal to 1 from $P$ is called a caterpillar.*

**Definition 6** *For a graph $G = (V, E)$ we define $G^k = (V, E^k)$ as the graph where two vertices are adjacent if and only if they have distance at most $k$ in $G$, i.e. $\{u, v\} \in E^k$ if and only if $d_G(u, v) \leq k$. We have $G = G^1 \subseteq G^2 \subseteq \ldots$.*

**Theorem 11 ([6])** *Let $T = (V, E(T))$ be a spanning tree of a graph $G = (V, E)$ such that $G \subseteq T^{k+1}$ and let $l$ be the number of leaves of the tree $T_1 = T - V_1(T)$ where $V_1(T)$ is the set of leaves of the tree $T$. Then $ds(G) \leq \left\lceil \frac{l}{2} \right\rceil (k + 1) + 1$.*

**Corollary 12** *Let* $T = (V, E(T))$ *be a spanning tree of a connected graph* $G = (V, E)$ *such that* $G \subseteq T^{k+1}$ *and let* $l$ *be the number of leaves of the tree* $T_1 = T - V_1(T)$ *where* $V_1(T)$ *is the set of leaves of the tree* $T$*. Then* $w(G) \leq \left\lceil \frac{l}{2} \right\rceil (k+1) + 1$.

**Theorem 13 ([6])** *Let* $T$ *be a spanning caterpillar of a graph* $G$ *and* $k$ *an integer such that* $G \subseteq T^{k+1}$*. Then* $ds(G) \leq \max\{2, k\}$.

**Corollary 14** *Let* $T$ *be a spanning caterpillar of a connected graph* $G$ *and* $k$ *an integer such that* $G \subseteq T^{k+1}$*. Then* $w(G) \leq \max\{2, k\}$.

### Co-Comparability Graphs

**Definition 7** *A graph* $G = (V, E)$ *is called co-comparability graph if there is a vertex ordering* $(v_1, v_2, \ldots, v_n)$ *of* $V$ *such that if* $i < j < k$ *and* $\{v_i, v_k\} \in E$ *then either* $\{v_i, v_j\} \in E$ *or* $\{v_j, v_k\} \in E$.

**Theorem 15 ([6])** *For every co-comparability graph* $G$ *we have* $ds(G) \leq 2$.

**Corollary 16** *For every connected co-comparability graph* $G$ *we have* $w(G) \leq 2$.

### AT-free Graphs

**Definition 8** *A set of three vertices of a graph* $G$ *is called an asteroidal triple if there exists a path between any two of them that avoids the neighborhood of the third.*

**Definition 9** *A graph is called asteroidal triple-free (AT-free) if it does not contain an asteroidal triple.*

The following Theorem was conjectured by Fomin et al. [6] and proved in [1].

**Theorem 17 ([6, 1])** *For every AT-free graph* $G$*,* $ds(G) \leq 2$.

**Corollary 18** *For every connected AT-free graph* $G$*,* $w(G) \leq 2$.

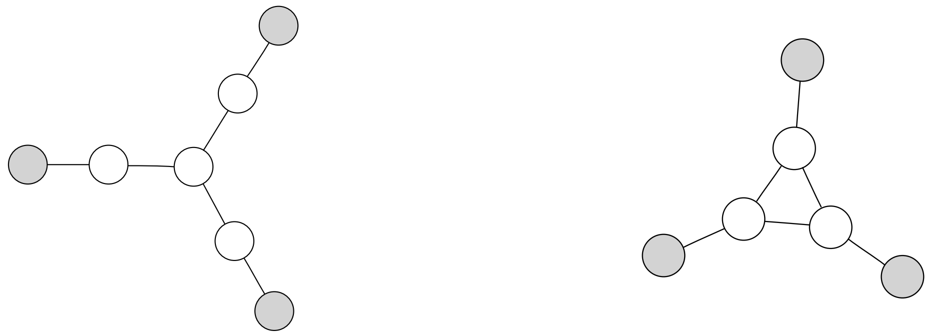

Figure 3.2: Examples of asteroidal triples

**Graphs with a Large Watchman Number**

Fomin et al. [6] showed that there are graphs with an arbitrary large domination search number.

**Theorem 19 ([6])** *For every $\varepsilon > 0$ and every integer $m$ there exists a graph $G$ on $n \geq m$ vertices such that $ds(G) = \Omega\left(n^{1-\varepsilon}\right)$.*

Note that the large graphs that are used here are *hypercube* graphs which are connected. Then by Theorem 6 the next corollary immediately follows.

**Corollary 20** *For every $\varepsilon > 0$ and every integer $m$ there exists a connected graph $G$ on $n \geq m$ vertices such that $w(G) = \Omega\left(n^{1-\varepsilon}\right)$.*

# Chapter 4

# 1-Watchable Graphs

In this chapter, we study the problem of characterizing 1-watchable graphs. In [5], Dalzell et al. introduced the *dominant subgraph* $G^\star$ of a graph $G$ and proved that if it is possible for a single watchman to clear $G$, then the range of its walk function can be restricted to $G^\star$.

**Definition 10 ([5])** *Given a graph* $G = (V, E)$ *the dominant subgraph* $G^\star$ *of* $G$ *is obtained as follows: Vertices* $v$ *that are subordinate to some other* $v' \in V$ *(i.e.* $N[v] \subseteq N[v']$*) are simultaneously removed from* $V$ *and edges* $\{u, v\}$ *with* $N[u] = N[v]$ *are contracted to a single node.*

For the graph $G$ in Figure 4.1 (that we already used in Chapter 2), $G^\star$ is the graph shown in Figure 4.2.

**Definition 11** *A graph* $G = (V, E)$ *is called interval graph if it has an interval representation, i.e. if for every vertex* $v_i \in V$ *there is an interval* $I_i = [l_i, u_i] = [l(I_i), r(I_i)]$ *such that the two intervals overlap if and only if their corresponding vertices are joined by an edge, i.e.*

$$I_j \cap I_k \neq \emptyset \quad \textit{if and only if} \quad \{v_j, v_k\} \in E.$$

Dalzell et al. characterized 1-watchable graphs in [5] as follows:

> *A graph* $G$ *is* 1*-watchable if and only if its dominant subgraph* $G^\star$ *is a connected interval graph.*

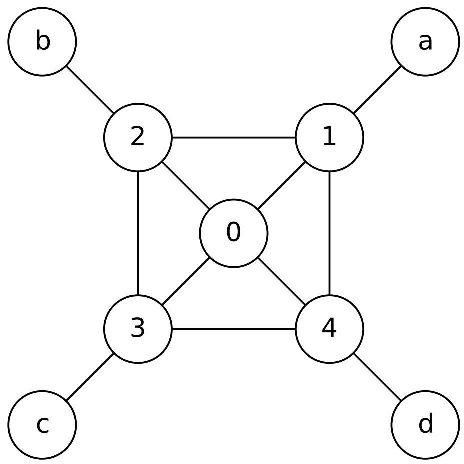


Figure 4.1: 1-watchable graph $G$

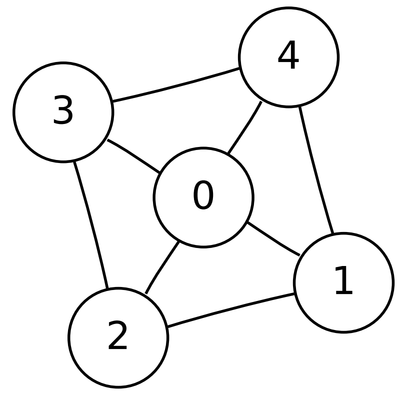


Figure 4.2: $G^\star$ of Figure 4.1

We shall show that their characterization is false in both directions, but that there are aspects of it that can be salvaged.

## 4.1 Counterexamples

A graph has an interval representation (i.e. it is an interval graph) if and only if it is chordal and asteroidal triple-free [11]. To argue the necessity direction of their characterization, Dalzell et al. claimed that

*If $G$ is 1-watchable, then $G^\star$ is chordal and AT-free.*

### 4.1.1 1-Watchable Graphs with Large Cycles and many Asteroidal triples

The first necessary condition of chordality is stated in Lemma 2.2 in [5]: *If $G$ is 1-watchable, then $G^\star$ is chordal.* Figure 4.1 is an example of a graph that is 1-watchable, yet its dominating subgraph $G^\star$ contains a 4-cycle and is not chordal, as shown in 4.2.

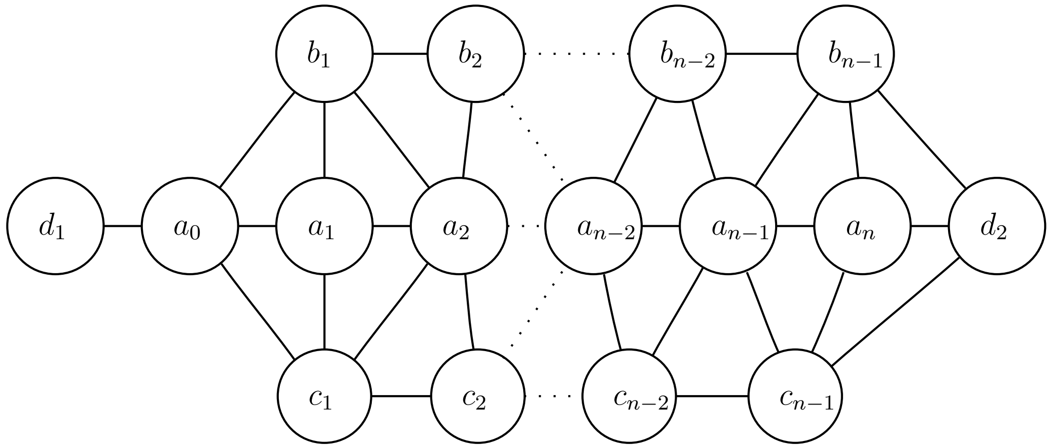


Figure 4.3: 1-watchable graph $G$ such that $G^\star$ contains a chordless $2n$-cycle

We also present a family of graphs that are 1-watchable although their dominant subgraph contains an induced chordless cycle of arbitrary length. In Figure 4.3 we show the construction of a graph $G_n$ whose dominant subgraph contains an induced chordless cycle of length $2n$ for every positive integer $n$. Formally, the graph $G_n = (V_n, E_n)$ is defined as follows:

- $V_n = \{a_0, a_1, \ldots, a_n, b_1, b_2, \ldots, b_{n-1}, c_1, c_2, \ldots, c_{n-1}, d_1, d_2\}$

- $E_n = \bigcup_{i=1}^{n-1} \{\{a_i, b_i\}, \{a_i, c_i\}, \{a_i, a_{i+1}\}, \{b_i, a_{i+1}\}, \{c_i, a_{i+1}\}\}$
  $\cup \{\{a_0, a_1\}, \{a_0, b_1\}, \{a_0, c_1\}\}$
  $\cup \{\{d_1, a_0\}, \{a_n, d_2\}, \{b_{n-1}, d_2\}, \{c_{n-1}, d_2\}\}$

It is easy to verify that $G_n^\star$ is the subgraph of $G_n$ induced by the vertices $V_n - \{d_1, d_2\}$. $G_n^\star$ contains the *chordless* cycle $(a_0, b_1, b_2, \ldots, b_{n-1}, a_n, c_{n-1}, c_{n-2}, \ldots, c_1, a_0)$ of length $2n$.

We can clear $G_n$ by using one watchman to walk along the path induced by $a_0, a_1, \ldots, a_n$. This way, at every step $k$ the vertices '*left*' of the watchman's position $a_k$, i.e. $\{a_i, b_i, c_i \mid i \leq k\}$, remain cleared as $\{a_{k-1}, b_{k-1}, c_{k-1}\} \in N[a_k]$ for $k = 2, \ldots, n$ and therefore $G$ is 1-watchable.

Furthermore, $G_n$ also serves as a counterexample for disproving the statement of Lemma 2.3 in [5]: *If $G$ is 1-watchable, then $G^\star$ is asteroidal triple-free.* For $n \geq 4$ we can, for example, choose non-adjacent vertices $b_r$, $b_s$ and $c_t$ for some $1 \leq r, s, t \leq n-1$ with $r < s$ as an asteroidal triple using paths $(b_r, b_{r+1}, \ldots, b_s)$, $(b_r, b_{r-1}, \ldots b_1, a_1, c_1, c_2, \ldots, c_t)$ and $(b_s, b_{s+1}, \ldots, b_{n-1}, a_{n-1}, c_{n-1}, c_{n-2}, \ldots, c_t)$ that connect any two of them avoiding the neighborhood of the third vertex.

**Definition 12** *Given a graph $G = (V, E)$, a vertex $v$ is called cut vertex if the removal of $v$ increases the number of connected components.*

**Lemma 21** *Given a connected graph $G = (V, E)$ with cut vertices $a$, $b$ and $c$ that form an asteroidal triple. Then $G$ is not 1-watchable.*

**Proof** Assume to the contrary that $G = (V, E)$ is a 1-watchable graph that contains an asteroidal triple $a$, $b$ and $c$ which are also cut vertices. First note that to effectively clear any of these vertices, let us say $a$, it needs to be visited. Otherwise, at the point the watchman leaves the neighborhood of $a$, $a$ gets recontaminated because the component of the subgraph induced by $V - \{a\}$ that does not contain $b$ or $c$ is still contaminated as $a$ was never visited.

Now let $\Pi$ be an arbitrary 1-watchman program. Without loss of generality, we can assume that $a$ is cleared first in $\Pi$. Afterwards, let us clear vertex $b$. Thus $b$ needs to be visited. At this point, $c$ is still contaminated as it was not visited before and there is a path from $c$ to $a$ that avoids the neighborhood of $b$ since $a$, $b$ and $c$ form an asteroidal triple. Therefore, $a$ gets recontaminated and at any time at most one of the vertices of $a$, $b$ and $c$ is cleared and $G$ cannot be 1-watchable. $\square$

### 4.1.2 Non-1-Watchable Graphs whose Dominant Subgraphs are Interval Graphs

On the other hand, we also disprove the sufficiency condition of Dalzell et al's characterization, that *every connected graph $G$ such that $G^\star$ is an interval graph is*

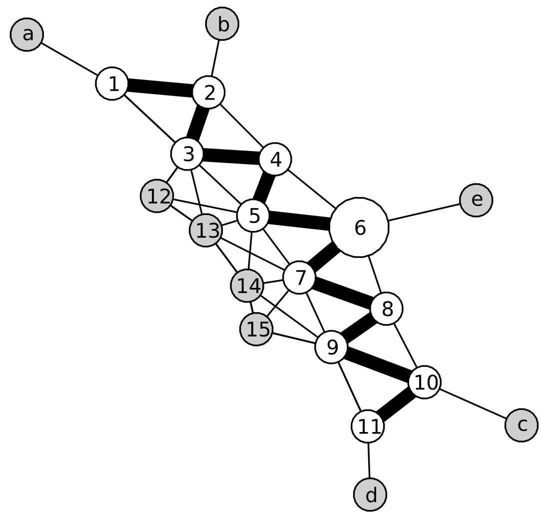


Figure 4.4: $G^{\star}$ is an interval graph, yet $G$ is not 1-watchable

1-*watchable.*

In Figure 4.4, we present a graph $G$ whose dominant subgraph $G^{\star}$ is an interval graph but $G$ is not 1-watchable. The dominant subgraph $G^{\star}$ of $G$ is the subgraph induced by the nodes $\{1, 2, \ldots, 11\}$ as the nodes 12, 13 are subordinate to 5 and the nodes 14, 15 to 9 and $a$, $b$, $c$, $d$, $e$ to 1, 2, 10, 11, 6. $G^{\star}$ can be represented by the following set of intervals. Define

$$I_k := [l_k, u_k] := [k, k + 2.5], \quad k = 1, \ldots, 11$$

then

$$I_i \cap I_j \neq \emptyset \Leftrightarrow \{i, j\} \in E(G^{\star}), \quad i \neq j$$

as shown in Figure 4.5.

Using the algorithm in [5], the watchman's walk for $G$ is the $(1, 11)$-path

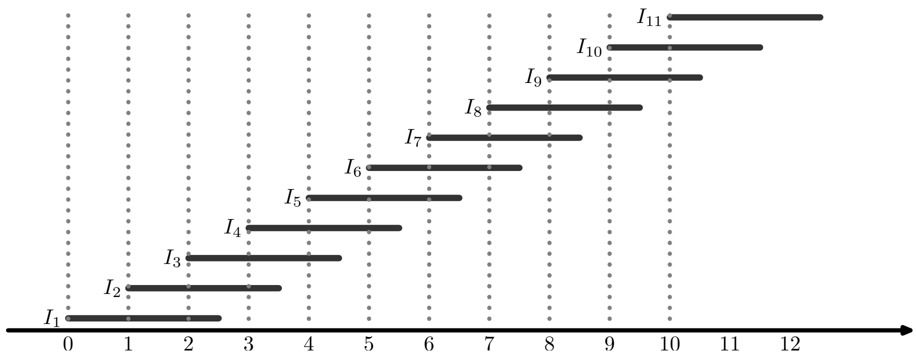


Figure 4.5: Interval representation of $G^{\star}$

$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)$. However, when the watchman is at vertex 6, there is a path from the contaminated vertex 11 to the already cleared vertex 1 described by $(11, 9, 15, 14, 13, 12, 3, 1)$ that does not contain a vertex in the closed neighborhood of 6. Therefore, vertex 1 gets recontaminated but never cleared again using the algorithm's watchman search program.

We also argue that this graph is not 1-watchable. There are three vertices 1, 6 and 11 that are cut vertices as they all have adjacent leaves. Furthermore, they form an asteroidal triple. Thus $G$ cannot be 1-watchable according to Lemma 21. Note that the graph in this example can be expanded by repeating the centerpiece, i.e. the subgraph induced by $\{4, 5, 6, 7, 13, 14\}$, in a natural way, so that there are many counterexamples to the original assertion that when $G^{\star}$ is a connected interval graph then $G$ is 1-watchable.

## 4.2 Interval Graphs

Although we showed that the original characterization is false, we can still salvage some of the results. Every connected interval graph $G$ is also a co-comparability graph[8]. According to Corollary 16, it has a watchman number of at most 2. This result can be improved:

**Theorem 22** *For every connected interval graph $G$ we have $w(G) = 1$.*

**Proof** Let $l^* = \min\{l(I_v), v \in V\}$; i.e., $l^*$ is the smallest left endpoint of the intervals representing the vertices of $G$. Similarly, let $r^* = \max\{r(I_v), v \in V\}$. Generate a sequence of nodes as follows.

1. Set $i = 0$ and $l_0 = l^*$.
2. While $l_i < r^*$, find the interval $I_v$ so that $I_v$ has the largest right endpoint among all the intervals that contain $l_i$. Set $w_i = v$, the vertex corresponding to $I_v$. Increment $i$ by 1 and update $l_i$ to $r(I_v)$.
3. Return the sequence $w_0, w_1, \ldots, w_{i-1}$.

Without loss of generality, let vertices $u^*$ and $v^*$ have $l(u^*) = l^*$ and $r(v^*) = r^*$. First, we note that $w_0$ exists since $u^*$ is a feasible candidate for $w_0$. Furthermore, if $w_{j-1}$ exists and $l_j = r(I_{w_{j-1}}) < r^*$, $w_j$ has to exist; otherwise, there is no path from $w_{j-1}$ to $v^*$, violating the assumption that $G$ is a connected graph. Thus, the algorithm we have described above will terminate and the very last vertex $w_{i-1}$ returned will have the property that the right endpoint of $I_{w_{i-1}}$ is $r^*$.

Second, since the point $l_j$ is contained in the intervals corresponding to $w_{j-1}$ and $w_j$ for $j = 1, \ldots, j-1$, the sequence $w_0, w_1, \ldots, w_{j-1}$ forms a path in $G$. Let us now argue that a watchman that walks through this path clears all the vertices of $G$.

**Claim 6** *For $j = 0, \ldots, i-1$, when the watchman is at $w_j$, all the vertices in the set $\{v : r(I_v) \leq r(I_{w_j})\}$ are cleared.*

*Proof of claim:* Suppose $j = 0$. If $u \in \{v : r(I_v) \leq r(I_{w_0})\}$ then $I_u \subseteq I_{w_0}$ since the left endpoint of $I_u$ cannot start earlier than $l^*$. That is, $u$ is adjacent to $v$ and so is cleared when the watchman is at $w_0$.

Assume the claim is true for an arbitrary $j$; let us now show that it remains true for $j+1$. When the watchman is at $w_j$, a contaminated node $c$ has to have $l(I_c) > r(I_{w_j})$ by our assumption. If a node $u$ in $\{v : r(I_v) \leq r(I_{w_j})\}$ is to be recontaminated via the node $c$, a path from $c$ to $u$ must go through a vertex whose corresponding interval contains the point $r(I_{w_j})$. But any such vertex is adjacent to $w_{j+1}$ since the interval $I_{w_{j+1}}$ contains $r(I_{w_j})$. Hence, the vertices in $\{v : r(I_v) \leq r(I_{w_j})\}$ are protected when the watchman moves from $w_j$ to $w_{j+1}$. Additionally, all the vertices $v$ with $r(I_{w_j}) < r(I_v) \leq r(I_{w_{j+1}})$ are also cleared by the watchman since these vertices are adjacent to $w_{j+1}$. It follows that all the vertices in $\{v : r(I_v) \leq r(I_{w_{j+1}})\}$ are cleared when the watchman is at $w_{j+1}$. By induction, the claim is true.

Finally, we note that since $r(I_{w_{i-1}}) = r^*$, $V = \{v : r(I_v) \leq r(I_{w_{i-1}})\}$; i.e., all vertices are cleared when the watchman walks through $w_0, w_1, \ldots, w_{i-1}$. $\square$

## 4.3 Spanning Caterpillars

For certain types of graphs that are induced by spanning caterpillars, we can also improve the upper bound on the watchman number given by Corollary 14.

**Theorem 23** *Let $T$ be a spanning caterpillar of a graph $G$ such that $G \subseteq T^2$. Then $w(G) = 1$.*

**Proof** (See Figure 4.6 for an example.)
Let $P = (v_0, v_1, \ldots, v_r)$ denote the backbone of $T$. We show that we can clear the graph $G$ by walking along the backbone $P$ using only one watchman, i.e. construct a watchman program $\Pi = (W_0, A_0), (W_1, A_1), \ldots, (W_r, A_r)$ with $W_l = (v_l)$ for $l = 0, 1, \ldots r$.

**Claim 7** *At every step $j$, $\bigcup_{l=0}^{j} N[v_l] \subseteq A_j$ for $j = 0, 1, \ldots, r$.*

*Proof of claim (by induction):* For $j = 0$ we have $A_0 = N[v_0]$. Suppose the claim is true for $j$. We now show that none of the vertices in $A_j$ get recontaminated when the watchman moves from $v_j$ to $v_{j+1}$. Therefore, we assume to the contrary, that a

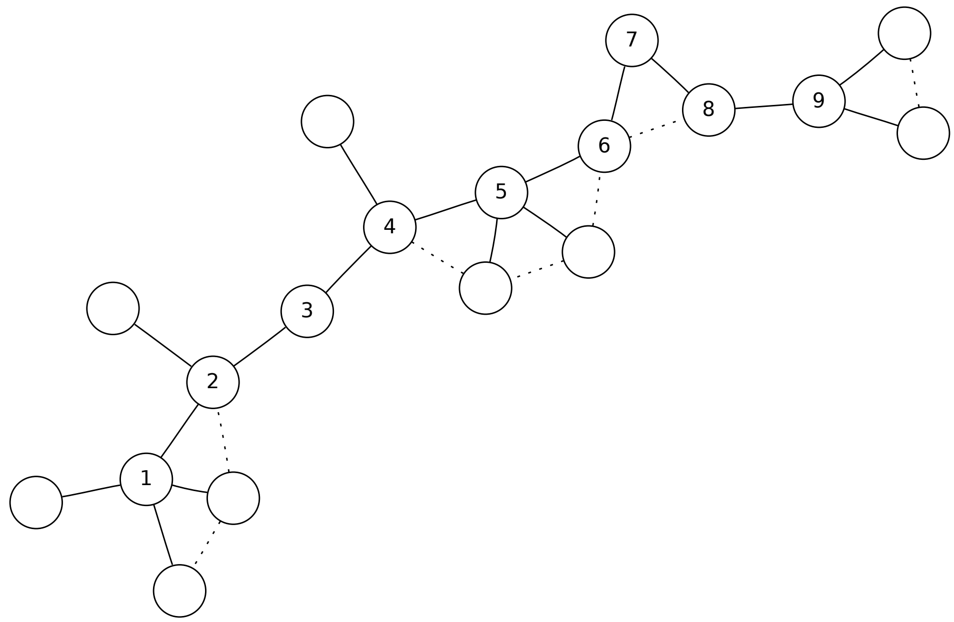


Figure 4.6: Example of a spanning caterpillar $T$ of a graph $G$ with $G \subseteq T^2$

vertex $v^r \in N[v_s]$ for some $s \leq j$ is recontaminated due to a path from $v^c \in N[v_t]$ for some $t \geq j+1$ to $v^r$. Such a path *has* to contain a non-starting vertex that lies in $N[v_{j+1}]$ because $G \subseteq T^2$, and therefore any two vertices $u$ and $v \in G$ have distance at most 2 in $T$. Hence, when the watchman is at $v_{j+1}$, all the vertices in $\bigcup_{l=0}^{j} N[v_l]$ are protected.

Additionally, the vertices in $N[v_{j+1}]$ are cleared. In other words, when the watchman is at $v_{j+1}$, all the vertices in $\bigcup_{l=0}^{j+1} N[v_l]$ are cleared. By induction it follows that when the watchman has reached $v_r$, all the vertices of $G$ are cleared. $\square$

# Chapter 5

# Other Graph Classes

In this chapter, we consider circular-arc graphs, another class of intersection graphs that are a natural generalization of interval graphs. We show that the watchman number of every connected circular-arc graph is at most 2. We also give an upper bound on the watchman number of a tree that is in the same order as its height.

## 5.1 Circular-arc Graphs

**Definition 13** *A circular-arc graph is an intersection graph obtained from a finite collection of arcs on a circle: Let $G = (V, E)$ be a graph with $V = \{v_1, v_2, \ldots, v_n\}$ and $C_i = (\theta_1(v_i), \theta_2(v_i))$ be the circular arc representing $v_i$ starting at the angle $\theta_1(v_i)$ and ending at $\theta_2(v_i)$, both angles measured counterclockwise from the positive x-axis. Then $\{v_i, v_j\} \in E$ if and only if $C_i \cap C_j \neq \emptyset$ for $i \neq j$. (See Figure 5.1 for an example.)*

**Theorem 24** *For every connected circular-arc graph $G$, $w(G) \leq 2$.*

**Proof** Let $G = (V, E)$ be a connected circular-arc graph with arcs $\{C_i\}$ representing the vertices $V = \{v_1, v_2, \ldots, v_n\}$.

We first arbitrarily choose one vertex $v_s \in V$ and place a *guard*, i.e. a watchman that does not move during the entire search process, on it. We can now restrict our search program to the vertices besides the closed neighborhood of $v_s$. Removing $N[v_s]$ from $V$ is like removing $C_s$ together with all other arcs $C_j$ that overlap with $C_s$, i.e. $C_s \cap C_j \neq \emptyset$. This results in a circular arc representation that has a hole in

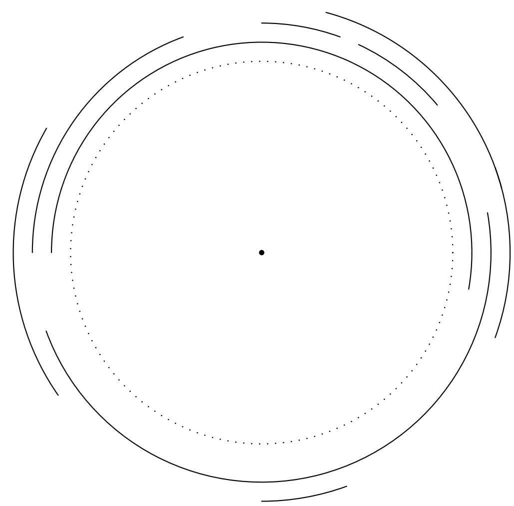

Figure 5.1: Circular-arc graph

it and can be flattened onto a line, making it an (not necessarily connected) interval graph. See Figure 5.2 for an illustration.

**Fact 1** *The subgraph of $G$ induced by $V - N[v_s]$ consists of connected components $G_1, G_2, \ldots, G_r$ that are interval graphs.*

Now we use another watchman clearing the individual interval graphs $G_1, G_2, \ldots, G_r$ subsequently using a walk constructed in Theorem 22. Note that all of those components are connected in $G$ through $N[v_s]$ as $G$ is connected. Also note that they are connected *only* through $N[v_s]$, i.e. all paths from a vertex $v_i \in G_i$ to another vertex $v_j \in G_j$ $(i \neq j)$ contain a vertex in $N[v_s]$. Therefore after clearing a whole component $G_i$, none of its vertices can get recontaminated from a contaminated vertex of another component $G_j$, $j > i$ because the one guard is stationed at $v_s$. Finally we have to make sure that (ii) of the definition of the watchman program is not violated and we indeed have a walk on $G$. This is the case as the second watchman can move from one component $G_i$ to another $G_j$ using a path of vertices of $N[v_s]$ as the subgraph induced by $N[v_s]$ is connected. □

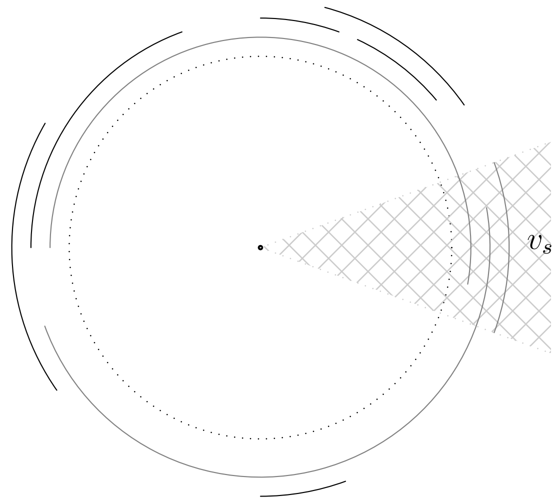


Figure 5.2: Cut circular-arc graph

## 5.2 Trees

In this section we analyze the watchman number for trees.

**Definition 14** *For a rooted tree* $T = (V, E)$ *with root* $r$ *we define the depth of a vertex* $v \in V$ *as the distance of* $v$ *from* $r$ *and write* $\mathrm{depth}(v) := d(v, r)$*. The set of all vertices of a given depth* $k$ *is called the* $k$*-th level of* $T$*. Then the height of the tree* $T$ *is given by the maximal depth of any vertex* $v \in T$ *and denoted by* $\mathrm{height}(T)$*. We define the tree-order of* $T$ *with respect to root* $r$ *by* $u < v$ *if* $u$ *lies below* $v$ *in* $T$*. We also define* $T_v$ *as the rooted subtree of* $T$ *at* $v$*. It follows that* $\mathrm{height}(T_v) \leq \mathrm{height}(T) - \mathrm{depth}(v)$.
*For a vertex* $v$ *we define the set of children of* $v$ *in* $T$ *by*

$$\mathrm{children}_T(v) := \{u \in V \mid \{u, v\} \in E \textit{ and } u < v\}$$

*and accordingly*

$$\mathrm{grandchildren}_T(v) := \{u \in V \mid u \in \mathrm{children}(v') \textit{ for some } v' \in \mathrm{children}_T(v)\}\,.$$

In [5] an upper bound on the watchman number on trees is given:

**Theorem 25 ([5])** *Let $T$ be a tree on $n$ vertices. Then $w(T) \leq \log_3(2n+7) - 3$.*

We deduce another upper bound that depends on the height of a tree and is better for dense trees, where every node tends to have many children. Note that for a perfect ternary tree $T$ on $n$ vertices, i.e. a tree where every internal node has three children, height$(T) = log_3(n)$.

**Theorem 26** *Let $T$ be a rooted tree with height $h$. Then*

$$w(T) \leq \left\lfloor \frac{h}{3} \right\rfloor + 1.$$

**Proof** We construct a quite watchman program that is very similar to some kind of 'multi-level' *depth first search* (DFS).

Let $T$ be a tree of height $h$ with root $r$. The cases $h = 0$ and $h = 1$ are trivial. Now first assume $h = 2$. Then $T$ can be cleared by 1 watchman as follows: Alternately visit the root $r$ and its children children$_T(r)$ in an arbitrary order. Note that once a subtree is cleared it cannot get recontaminated as it is only connected to other subtrees through the root $r$ that is always protected.

Now suppose the claim to be true for $h$, i.e. we can clear a tree of height $h$ using $k := \lfloor \frac{h}{3} \rfloor + 1$ watchmen. We now show that we can also clear a tree $T'$ of height $h+3$ (and thus with height $h+1$ and also $h+2$) and root $r'$ using $\lfloor \frac{h+3}{3} \rfloor + 1 = \lfloor \frac{h}{3} \rfloor + 2 = k+1$ watchmen:

Let the first watchman alternate through $r'$ and its children. During this process the additional $k$ watchmen can clear the lower part of the tree: While the first watchman resides at child $v \in$ children$_{T'}(r')$ all children of $v$ are cleared and we can clear the subtrees originating at their children, i.e. we clear $T'_{v''}$ with $v'' \in$ grandchildren$_{T'}(v)$ using $k$ watchmen as height$(T'_{v''})$ = height$(T') - 3 = h + 3 - 3 = h$. Note that no recontamination occurs among the subtrees $T'_{v''}$ as they are only connected through $v'$ that is protected by the first watchman, residing at its parent node $v$. Afterwards, the first watchman can proceed to visit the next child $v \in \text{children}_{T'}(r')$ after passing through $r'$ itself. Doing this, all previously cleared subtrees $T'_v$ with $v \in \text{children}_{T'}(r')$ cannot get recontaminated as the root $r'$ is always protected by the first watchman. $\square$

**Fact 2** *The inequality in Theorem 26 is only an upper bound: A degenerated tree of height $n$, i.e. a simple path of length $n+1$, has watchman number 1 for every integer $n \geq 0$.*

**Definition 15** *A vertex is central in a graph $G = (V, E)$ if its greatest distance from any other vertex is as small as possible. This distance is the radius of $G$, denoted by $\text{rad}(G)$.*

**Corollary 27** *Let $T$ be a tree. Then $w(T) \leq \left\lfloor \frac{\text{rad}(T)}{3} \right\rfloor + 1$.*

**Proof** Root $T$ at one of its central vertices. Then, by definition of $\text{rad}(T)$, $\text{rad}(T) = \text{height}(T)$. $\square$

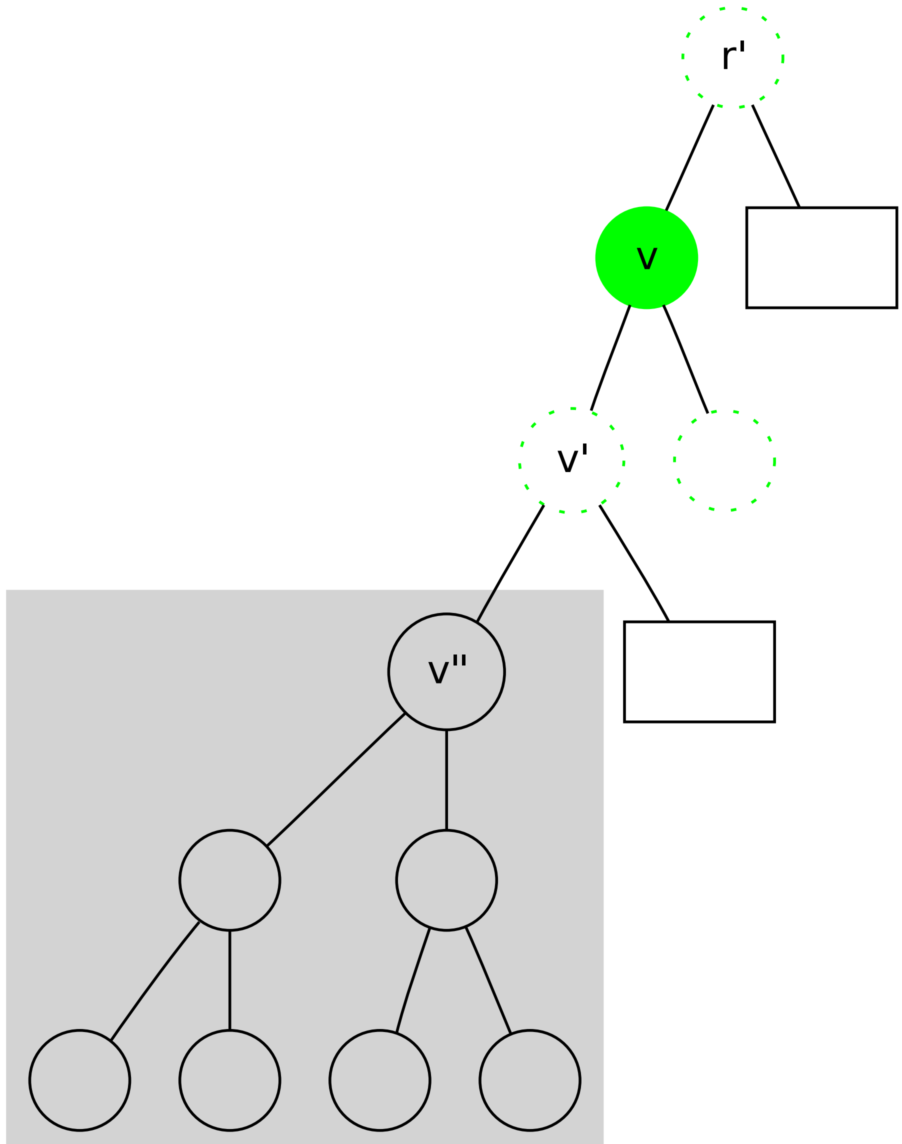


Figure 5.3: Tree $T$ of height 5 with $w(T) \leq 2$

# Chapter 6

# Conclusion and Open Problems

## 6.1 Our Results

In this paper we analyzed the watchman search problem, introduced by Dalzell et al. in [5] as a continuous model, and formulated it as a discrete model. This allowed us to compare the model with other, well-established discrete search models.

Our main results are:

- We showed the very close relationship of the watchman search program to domination search program. We proved that for any connected graph the watchman number and domination search number differs by at most one. This implies that determining the watchman number and also approximating it is **NP**-hard unless **P** = **NP**. Further, various lower and upper bounds on the watchman number can directly be inherited from the domination search number.

- In [5] Dalzell et al. characterized the 1-watchable graphs as precisely those whose dominant subgraphs are connected and have an interval representation. We showed that this is false. First, we constructed a family of graphs whose dominant subgraph contains an induced cycle of arbitrary length and has many asteroidal triples, yet 1-watchable. This disproves Lemma 2.2 and Lemma 2.3 from [5] that is the necessary condition of above statement.

  On the other hand, we also gave an example of a graph whose dominant subgraph is a connected interval graph but is 2-watchable and *not* 1-watchable, also disproving the sufficient condition of their characterization. Avoiding the notion of the dominant subgraph, we proved that the watchman number of every connected interval graph is one.

- In Chapter 5 we derived some results for the watchman number of specific graph classes. We proved that the watchman number of every connected circular-arc graph is at most 2. We gave an upper bound for the watchman number of a general tree that depends on its radius.

## 6.2 Outlook and Open Problems

The graph class of one-watchable graphs have not been fully characterized yet. It would be interesting to know whether this family of graphs matches an existing class and therefore implies the relationship of the watchman number to other graph parameters or properties, or if it defines a completely new superclass of interval graphs.

Further, we only gave an upper bound on the watchman number of trees. A better result would be to come up with an efficient algorithm that determines the actual watchman number of a tree. This seems to be possible following a similar approach to the recursive method of Megiddo et al. [12] to calculate the node search number of a tree efficiently.

Although we showed that not every $k$-watchable graph can be cleared using a *monotone* watchman program, it seems very likely to us that there is at least some kind of *loose* monotonicity that needs to be identified properly.

# Bibliography

[1] D. Aggarwal, S. K. Mehta, and J. S. Deogun. Domination search on graphs with low dominating-target-number. *Lecture Notes in Computer Science*, 3787:28–37, 2005.

[2] B. Alspach. Searching and sweeping graphs: a brief survey. *Matematiche (Catania)*, 59:5–37, 2006.

[3] D. Bienstock and P. Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991.

[4] R. Breisch. An intuitive approach to speleotopology. *A publication of the Southwestern Region of the National Speleological Society*, VI:72–78, 1967.

[5] J. Dalzell, D. Leach, and M. Walsh. The paranoid watchman: a search problem on graphs. In preparation.

[6] F. V. Fomin, D. Kratsch, and H. Müller. On the domination search number. *Discrete Applied Mathematics*, 2003.

[7] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theor. Comput. Sci.*, 399(3):236–245, 2008.

[8] P. C. Gilmore and A. J. Hoffman. A characterisation of comparability graphs and of interval graphs. *Canadian Journal of Mathematics*, 1964.

[9] P. A. Golovach. Equivalence of two formalizations of a search problem on a graph (russian). *Vestnik Leningrad. Univ. Mat. Mekh. Astronom.*, pages 10–14, 1989. translation in Vestnik Leningrad Univ. Math. 22 (1989), no. 1, 13-19.

[10] A. S. LaPaugh. Recontamination does not help to search a graph. *J. ACM*, 40(2):224–245, 1993.

[11] C. G. Lekkerkerker and J. C. Boland. Representation of a finite graph by a set of intervals on the real line. *Fund. Math.*, 51:45–64, 1962.

[12] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. ACM*, 35(1):18–44, 1988.

[13] T. Parsons. Pursuit evasion in a graph. *Theory and Applications of Graphs*, pages 426–411, 1976.