On tractability of Cops and Robbers game

Fedor V. Fomin[1], Petr A. Golovach[1], and Jan Kratochvíl[2]

[1] Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway,
fedor.fomin|petr.golovach@ii.uib.no*
[2] Department of Applied Mathematics, and Institute for Theoretical Computer Science,
Charles University, Malostranské nám. 25, 118 00 Praha 1, Czech Republic
honza@kam.mff.cuni.cz†

Abstract. The Cops and Robbers game is played on undirected graphs where a group of cops tries to catch a robber. The game was defined independently by Winkler-Nowakowski and Quilliot in the 1980s and since that time has been studied intensively. Despite of that, its computation complexity is still an open question. In this paper we prove that computing the minimum number of cops that can catch a robber on a given graph is NP-hard. Also we show that the parameterized version of the problem is W[2]-hard. Our proof can be extended to the variant of the game where the robber can move s times faster than cops. We also provide a number of algorithmic and complexity results on classes of chordal graphs and on graphs of bounded cliquewidth. For example, we show that when the velocity of the robber is twice cop's velocity, the problem is NP-hard on split graphs, while it is polynomial time solvable on split graphs when players posses the same speed. Finally, we establish that on graphs of bounded cliquewidth (this class of graphs contains, for example, graphs of bounded treewidth), the problem is solvable in polynomial time in the case the robber's speed is at most twice the speed of cops.

Key words: Pursuit-evasion games on graphs, complexity, parameterized complexity, algorithms, cliquewidth

1 Introduction

Cops and Robbers is a pursuit-evasion game with two players cop $\mathcal{C}$ and robber $\mathcal{R}$ which play alternately on a finite connected undirected graph G. Player $\mathcal{C}$ has a team of cops who attempt to capture the robber. At the beginning of the game $\mathcal{C}$ selects vertices and put cops on these vertices. Then $\mathcal{R}$ put the robber on a vertex. The players take turns starting with $\mathcal{C}$. At every move each of the cops can be either moved to an adjacent vertex or kept on the same vertex. (Several cops can occupy the same vertex at some move.) $\mathcal{R}$ responds by moving the robber to some vertex along some path of length at most s, which does not contain vertices occupied by cops. (In other words, cops are moving with a unit speed and the speed of robber is s, and robber cannot run through a vertex

* Supported by the Norwegian Research Council.

† Supported by Czech research grant 1M0545.

occupied by a cop.) We say that a cop *catches* the robber at some move if at that move they occupy the same vertex. Player $\mathcal{C}$ wins if in a finite number of moves one of his cops catches the robber. Player $\mathcal{R}$ wins if he can avoid such a situation. For an integer s and a graph G, we denote by $c_s(G)$ the minimum number of cops sufficient for $\mathcal{C}$ to win on graph G against the robber moving at the speed of s.

The variant of the game with $s = 1$, i.e. when cops and robber have the same speed, was studied intensively. The game was defined (for one cop) by Winkler and Nowakowski [25] and Quilliot [28] who also characterized graphs with the cop number one. Aigner and Fromme [2] initiated the combinatorial study of the problem with several cops and obtained a number of important results. In particular, they observed that if a girth of G (the minimum length of a cycle) is at least 5, then $c_1(G)$ is at least the minimum vertex degree of G. Another interesting result proved in [2] is that on planar graphs 3 cops can always catch the robber. This result can be generalized on graphs of bounded genus [27, 31]. Andreae [5] extended the result of Aigner and Fromme to graphs containing no fixed graph H as a minor. Different combinatorial (lower and upper) bounds on the cop number for different graph classes are discussed in [4, 13, 15, 16, 20, 22, 23] (see also the survey [3]).

There is a resemblance of Cops and Robbers game, at least for large values of $s \to \infty$, to the helicopter search game defined by Seymour and Thomas [32], which is the game-theoretic interpretation of the well known treewidth parameter. In Seymour-Thomas game the robber can move arbitrarily fast, but players make their moves simultaneously. See the survey of Bodlaender for an overview of pursuit-evasion games related to treewidth [7].

Despite of such an intensive study of the combinatorial properties of the game almost no algorithmic results on this game are known. Perhaps the only algorithmic result known about Cops and Robbers game (for $s = 1$) is the observation that determining whether the cop number of a graph on n vertices is at most k can be done by a backtracking algorithm which runs in time $O(n^{O(k)})$ (thus polynomial for fixed k) [6, 17, 19].

Similar result holds for every $s \geq 1$. Given an integer k and a graph G on n vertices, the question if $c_s(G) \leq k$ can be answered (and the corresponding winning strategy of k cops can be computed) by constructing the game graph on $2\binom{n+k-1}{k}n$ nodes (every node of the game graph corresponds to a possible position in G of k cops and one robber, taking into account two possibilities for the turn), and then by making use of backtracking find if some cop-winning position can be obtained from an initial position. While the proof of the following proposition is standard and easy (and we omit it here), it serves as the main tool for obtaining all polynomial time algorithms in this work.

Proposition 1. *For a given integer $k \geq 1$ and a graph G on n vertices, the question if $c_s(G) \leq k$ can be answered in time $\binom{n+k-1}{k}^2 \cdot n^{O(1)} = n^{O(k)}$.*

Thus for every fixed k, one can decide in polynomial time if k cops can catch the robber on a given graph G. There are several natural questions around

Proposition 1. The first is, what is the complexity of the problem when k is part of the input? Another question, is the problem fixed parameter tractable? There are many search and pursuit-evasion problems which are fixed parameter tractable, i.e. for which deciding if k searchers (cops) can catch evader (robber) on an n-vertex graph can be done in time $O(f(k) \cdot n^{O(1)})$ (we refer to Bodlaender's survey [7] for examples of such problems).

There are several variants of similar games like the k-pebbles game, or the cat and k-mouse game, which solutions require $n^{\Omega(k)}$ steps (see e.g. Adachi et al. [1]). However, all these games are played on directed graphs or the games should either start, or end in specified positions (holes or cheese for mouses), and the proofs are strongly based on these specific properties. Following this line of research, Goldstein and Reingold [17] proved that the version of the Cops and Robbers game on *directed* graphs is EXPTIME-complete. Also they have shown that the version of the game on undirected graphs when cops and robber are given their initial positions is also EXPTIME-complete. They also conjectured that the game on undirected graphs (for $s = 1$) is also EXPTIME-complete. Again, their proofs strongly relies on the specific settings (adding directions or fixing initial positions) and cannot be transferred to the standard Cops and Robbers game on undirected graphs, and their conjecture is still open.

Our results. We prove that for every $s \geq 1$, deciding if $c_s(G) \leq k$ is NP-hard. We also show that the parameterized version of the problem is $W[2]$-hard. Loosely speaking, this means that the existence of a $O(f(k) \cdot n^{O(1)})$-time algorithm deciding if $c_s(G) \leq k$, where f is a function only of the parameter k and G is a graph on n vertices, would imply that $FPT = W[2]$, which is considered to be very unlikely in parameterized complexity. (We refer to the books [12, 14, 24] for an information on parameterized complexity.) We also show that for $s \geq 2$, the problem remains NP-hard and $W[2]$-hard even when input is restricted to split graphs. We find it a bit surprising, especially for $s = \infty$, i.e. when the speed of robber is not bounded, because all known search and pursuit-evasion problems on undirected graphs which look quite similar to this case, are polynomially solvable or at least fixed parameter tractable for chordal graphs. For example, for helicopter search game [32] the minimum number of cops equals treewidth plus one and can be easily calculated for chordal graphs. For node searching (see [18]) the corresponding problem can be solved in polynomial time for split graphs but remains NP-complete on chordal graphs. See also [26] for related results. Note also that for $s = 1$ one cop always can capture robber on the chordal graph [28]. By continuing investigating the complexity of the problem on classes of chordal graphs, we show that for every fixed s, the computation of $c_s(G)$ on interval graphs can be done in polynomial time. Finally, we investigate the complexity of the problem on graphs of bounded cliquewidth. We prove that on graphs of bounded cliquewidth the computation of numbers $c_s(G)$ can be done in polynomial time for $s = 1, 2$. While most of polynomial time algorithms on graphs of bounded cliquewidth (and treewidth) are based on dynamic programming approach [11], this is not the case for the

Cops and Robbers problem. Our proof is based on combinatorial bounds and Proposition 1.

2 Cops and Robbers is NP hard

All this section is devoted to the proof of the following result

Theorem 1. *For every $s \geq 1$, the following problem is NP-hard*

INSTANCE: A graph G and a positive integer k.
QUESTION: Is $c_s(G) \leq k$?

Moreover, the parameterized version

INSTANCE: A graph G.
PARAMETER: A positive integer k.
QUESTION: Is $c_s(G) \leq k$?

of the Cops and Robbers problem is $W[2]$-hard for every $s \geq 1$.

2.1 Bipartite graphs with large girth and degrees of vertices

Let us start with auxiliary results. We want to construct a bipartite graph with girth at least six and large minimum vertex degree with some additional properties. (Let us remind that the girth of a graph G is the minimum cycle length in G.) The study of such graphs has a long history (see e.g. [8]). There are different approaches for obtaining such graphs. Most of them are geometrical or algebraic. For our reduction we use algorithmic construction which is based on the construction of Krishnan et al. [21].

For positive integers n, m and r we construct a bipartite graph $H(n, m, r)$ with rmn^2 edges and bipartition (X, Y), $|X| = |Y| = nm$. Set X is partitioned into sets $U_1, U_2, \dots, U_n$, and set Y is partitioned into sets $W_1, W_2, \dots, W_n$, $|U_i| = |W_i| = m$ for $i = 1, 2, \dots, n$. We denote by $H_{i,j}$ the subgraph of $H(n, m, r)$ induced by $U_i \cup W_j$, and by $\deg_{i,j}(z)$ the degree of vertex z in $H_{i,j}$. We also denote by E the set of edges in $H(n, m, r)$ and by $\text{dist}(x, y)$ the distance between vertices x and y in $H(n, m, r)$.

The graph $H(n, m, r)$ is constructed by the following procedure which starts from empty graph on vertices $X \cup Y$ and add edges according the following rules:

```
for $k := 1$ to $rm$ do
    let $t := \lceil \frac{k}{m} \rceil$;
    if $k$ is odd then
        for $i := 1$ to $n$ do
            for $j := 1$ to $n$ do
                choose a vertex $x \in U_i$ of minimum degree in $H_{i,j}$;
                let $S := \{z \in W_j \colon \mathrm{dist}(x,z) > 1$ and $\deg_{i,j}(z) < t+1\}$;
                select a vertex $y \in W_j$ such that
                $\mathrm{dist}(x,y) = \max_{z \in S} \mathrm{dist}(x,z)$; add $(x,y)$ to $E$;
    else
        for $j := 1$ to $n$ do
            for $i := 1$ to $n$ do
                choose a vertex $y \in W_j$ of minimum degree in $H_{i,j}$;
                let $S := \{z \in U_i \colon \mathrm{dist}(y,z) > 1$ and $\deg_{i,j}(z) < t+1\}$;
                select a vertex $x \in U_i$ such that
                $\mathrm{dist}(x,y) = \max_{z \in S} \mathrm{dist}(x,z)$; add $(x,y)$ to $E$;
```

Value of t is called the phase number of the algorithm. Clearly, the algorithm has to complete r phases. If k is odd then we say that n^2 edges, added by the algorithm for this value of k, are added during the odd phase t. Correspondingly, if k is even then we say that n^2 edges, added by the algorithm for this value of k, are added during the even phase t.

The following lemma, which is the direct analog of Lemma 1 from [21], establishes the key invariants maintained by the algorithm. We omit the proof of this lemma here.

Lemma 1. *For every $1 \leq t \leq r$ the following holds:*

1. *When the algorithm completes an odd phase t, the average degree of vertices of U_i in $H_{i,j}$ is r and $t-1 \leq \deg_{i,j}(x) \leq t+1$ for $x \in U_i$ and $i,j \in \{1,2,\ldots,n\}$;*
2. *When the algorithm completes an even phase t, the average degree of vertices of W_j in $H_{i,j}$ is r and $t-1 \leq \deg_{i,j}(y) \leq t+1$ for $y \in W_j$ and $i,j \in \{1,2,\ldots,n\}$.*

It can be easily seen that if set S is empty then the algorithm cannot add an edge. Next lemma gives sufficient condition, which makes such situation impossible.

Lemma 2. *If $r < \frac{m+3}{6}$ then the algorithm completes all r phases.*

This lemma is a simplified version of the lemma 2 of [21] and we omit its proof here.

Now we can summarize properties of the algorithm and of the graph $H(n,m,r)$ which will be used in our reduction.

Lemma 3. *Let* $m \geq 2n(r+1)\frac{(n(r+1)-1)^6-1}{(n(r+1)-1)^2-1}$. *Then*

1. *The algorithm constructs graph* $H(n,m,r)$ *in time* $O(r \cdot m \cdot n^2)$*;*
2. *For every vertex* $z \in V(H_{i,j})$ *and every* $i, j \in \{1, 2, \dots, n\}$*, we have* $r-1 \leq \deg_{i,j}(z) \leq r+1$*;*
3. *For every vertex* z*,* $\deg(z) \leq n(r+1)$*.*
4. *The girth of* $H(n,m,r)$ *is at least six.*

Proof. The first three items are immediate corollaries of Lemmata 1 and 2.

In order to prove 4, let us assume that a cycle of length $g = 2p$, $p \geq 1$, where g is the girth of $H(n,m,r)$, was created during the phase t of the algorithm. Without loss of generality, we can assume that the last edge (x,y) of this cycle was added during odd phase t, and $x \in U_i$, $y \in W_j$. Let $D = \{z \in W_j : \operatorname{dist}(x,z) \geq g\}$. Since vertex x had no neighbors in D, we have that for every $z \in D$ $\deg_{i,j}(z) = t+1$ during the even phase t. By Lemma 1, $|D| \leq \frac{m}{2}$. Thus $|W_j \setminus D| \geq \frac{m}{2}$. Clearly $\operatorname{dist}(x,z) \leq g-1 = 2p-1$ for every $z \in W_j \setminus D$. Let us estimate the number of vertices at distance at most $2p-1$ from x in $H(n,m,r)$. Since the maximum vertex degree in $H(n,m,r)$ is at most $n(r+1)$, we have that the number of vertices at distance at most $2p-1$ from x is at most

$$n(r+1) + n(r+1)(n(r+1)-1)^2 + \cdots + n(r+1)(n(r+1)-1)^{2(p-1)}$$
$$= n(r+1)\frac{(n(r+1)-1)^{2p}-1}{(n(r+1)-1)^2-1}.$$

Thus

$$n(r+1)\frac{(n(r+1)-1)^6-1}{(n(r+1)-1)^2-1} \leq \frac{m}{2} \leq n(r+1)\frac{(n(r+1)-1)^{2p}-1}{(n(r+1)-1)^2-1},$$

which yields $g = 2p \geq 6$. □

2.2 Proof of Theorem 1

Now we are ready to proceed with the proof of the main result of this section. We use reduction from the well known NP-complete Minimum Dominating set problem

INSTANCE: A graph G and a nonnegative integer k.
QUESTION: Does G contain a dominating set (i.e. a set of vertices D such that every vertex of G is either in D, or is adjacent to a vertex of D) of cardinality at most k?

Let G be a graph with the vertex set $V(G) = \{v_1, v_2, \dots, v_n\}$. Let $r = k+2$ and

$$m = \left\lceil 2n(r+1)\frac{(n(r+1)-1)^6-1}{(n(r+1)-1)^2-1}\right\rceil.$$

For every vertex $v_i \in V(G)$ we add $2m$ new vertices and make each new vertex adjacent to vertices from $N[v_i]$ (in G). We use m of the new vertices to compose the set U_i, and the other m vertices to compose the set W_i. Then we apply the algorithm from the previous section to construct the bipartite graph $H(n,m,r)$ on the vertex set

$$(U_1 \cup U_2 \cup \cdots \cup U_n) \cup (W_1 \cup W_2 \cup \cdots \cup W_n).$$

Denote the resulting graph by G'. By Lemma 3, G' is constructed in time polynomial in n and k.

Now we prove that graph G has a dominating set of size at most k if and only if $c_s(G') \leq k$.

We say that vertex is dominated by the cop if this vertex is occupied by the cop or some adjacent vertex is occupied by the cop.

Let $S \subseteq V(G)$ be a dominating set in G of size $\leq k$. Since cops placed in vertices of S dominate all vertices of G', for every vertex choice of robber he will be caught after the first move of cops.

In opposite direction, let us assume that G has no dominating set of size k and describe the strategy of the robber avoiding cops. Let S be the set of vertices chosen by cops for their initial position. Since this set is not a dominating set in G, we have that there is a vertex $v_i \in V(G)$ which is not dominated by cops. Degree of every vertex of $H(n,m,r)$ is at most $n(r+1)$ and thus k cops dominate at most $kn(r+1)$ vertices in U_i. The set U_i contains m vertices, therefore,

$$m = \left\lceil 2n(r+1)\frac{(n(r+1)-1)^6-1}{(n(r+1)-1)^2-1}\right\rceil > kn(r+1).$$

So there is a vertex $u \in U_i$ which is not dominated by cops. The robber chooses this vertex as his initial position. Suppose now that after some robber's move the robber occupies vertex $u \in U_i$ which is not dominated by cops. If after the next move of cops this vertex is still not dominated then the robber stays there. If it it becomes dominated, then the robber do the following. Let S be the set of vertices of G occupied by cops. Since this set is not a dominating set in G, there is vertex $v_j \in V(G)$ which is not dominated by cops standing at S. The vertex u has at least $r-1 = k+1$ neighbors in W_j. Since graph $H(n,m,r)$ has the girth at least six, we have that at least one of these neighbors is not dominated by cops. Then the robber moves into this vertex (note that he moves along the path of length 1). Clearly, this strategy of the robber gives him possibility to avoid cops. This completes the NP-hardness part of the proof.

To prove $W[2]$-hardness, it is sufficient to observe that our reduction from dominating set (which is $W[2]$-hard) is an FPT reduction.

3 Complexity on Split and Interval graphs

A graph G is a *split graph* if the vertex set of G can be partitioned into sets C and I, such that C is a clique, and I is an independent set. It is well known that the treewidth of a split graph can be computed in linear time (actually it is true for a larger class of chordal graphs). It is also well known that $c_1(G) = 1$ on a superclass of chordal graphs and can be computed in polynomial time [25]. Also the treewidth of a chordal graph can be computed in polynomial time, and thus the search game of Seymour-Thomas is tractable on chordal graphs. However, for $s \geq 2$ problem of computing of $c_s(G)$ becomes difficult even for split graphs.

Theorem 2. *For every $s \geq 2$ the following problem is NP-hard:*

INSTANCE: A split graph G, and a nonnegative integer k.
QUESTION: Is $c_s(G) \leq k$?

Moreover, for every $s \geq 2$ the parameterized version of the problem is $W[2]$-hard on split graphs.

Proof. The proof of this theorem uses the constructions from the proof of Theorem 1. It is known that the Minimum Dominating set problem is NP-complete (and its parameterized version is W[2]-hard) even when the input is restricted to split graphs [29].

Let G be a split graph with clique C and independent set $I = \{v_1, v_2, \dots, v_p\}$. Let also $r = k + 2$ and $m = \lceil 2(r+1)\frac{r^6-1}{r^2-1} \rceil$. Each vertex $v_i \in I$ is replaced by new m vertices, which form set V_i. Let $N(v_i)$ be the set of neighbors of v_i in the original graph G. We make every new vertex from V_i be adjacent to all vertices from $N(v_i)$. Then we add m vertices forming a set W to the clique (i.e. these vertices are joined by edges with each other and vertices of C). Now we construct p copies of the graph $H(1, m, r)$ with vertex sets $V_1 \cup W, V_2 \cup W, \dots, V_p \cup W$ ($V_i = X$ and $W = Y$ for each copy of $H(1, m, r)$). The resulting graph is denoted by G'. Clearly, this graph is a split graph, and can be constructed in polynomial time.

Now we prove that for any $s \geq 2$, graph G has a dominating set of size at most k if and only if $c_s(G') \leq k$.

Suppose that $S \subseteq V(G)$ is a dominating set in G and $|S| \leq k$. Clearly we can assume that $S \subseteq C$. It can be easily seen that S is a dominating set in G'. We place cops in vertices of S, and for every possible choice of an initial position, the robber would be captured after the first move of cops.

Assume now that for every $S \subseteq V(G)$, $|S| \leq k$, S is not a dominating set of G, and describe the strategy of the robber. Suppose that cops have chosen initial positions, and S is the set of vertices of G occupied by cops. Since this set is not a dominating set in G, there is $i \in \{1, 2, \dots, p\}$ such that vertices of V_i are not dominated by cops standing on vertices of S. Since each vertex $u \in W$ is adjacent to no more than $k + 3$ vertices of V_i and $k(k + 3) + 1 \leq m$, we have that there is vertex $x \in V_i$ which is not dominated by cops standing on vertices of W. The robber chooses this vertex as his initial position. Suppose now that after some moves the robber occupies vertex $x \in V_i$ which is not dominated by cops. If after next move of cops this vertex is still not dominated, then the robber stays there. Suppose that it became dominated. Let S be the set of vertices of G occupied by cops. Since this set is not a dominating set in G, there is $j \in \{1, 2, \ldots, p\}$ such that vertices of V_j are not dominated by cops standing on vertices of S. Vertex x has at least $k + 1$ adjacent vertices in W. So there is vertex $y \in W$ which is adjacent to x and is not occupied by cops. Now vertex y has at least $k+1$ neighbors in V_j. Since graph $H(1, m, r)$ has the girth at least six, at least one vertex $z \in V_j$ in the neighborhood of y is not dominated by cops. Then the robber can move from x to y and then to z. Such a strategy provides the robber an opportunity to avoid capture.

To establish the parameterized complexity on split graph we observe, that the parameterized version of the dominating set problem remains to be $W[2]$-hard on split graphs and that the described reduction from dominating set is an FPT reduction. □

Another well known class of chordal graphs are interval graphs. An *interval graph* is the intersection graph of a set of intervals on the real line, i.e. every vertex corresponds to an interval and two vertices are adjacent if and only if the corresponding intervals intersect. We show that for every interval graph G and integer s, $c_s(G)$ can be computed in polynomial time. Actually the only property of interval graphs we need is the existence in interval graphs dominating pairs. A *dominating pair* in a connected graph G is a pair of two (not necessary different) vertices u and v such that the vertex set of every u, v-path in G is a dominating set. A *caterpillar* is a tree which consists of a path, called *backbone*, and leaves adjacent to vertices of the backbone. For a graph G and integer p, the p-th power of G, G^p is the graph on vertex set $V(G)$, and vertices u, v are adjacent in G^p if and only if the distance between them is at most p in G.

Lemma 4. *Let T be a spanning caterpillar of a graph G, and let p be an integer such that G is a subgraph of T^p. Then $c_s(G) \leq \max\{1, ps - 1\}$.*

Proof. We describe a winning strategy for $k = \max\{1, ps - 1\}$ cops. Suppose that $P = (v_1, v_2, \ldots, v_r)$ is a backbone of T. Cops occupy first k vertices of the backbone. Then they move along P simultaneously. If after some robber's move he is standing on the vertex adjacent to the vertex occupied by a cop, then this cop makes capturing move.

For a vertex v we use $N[v]$ to denote the closed neighborhood of v, i.e. the set of all vertices adjacent or equal to v. We use induction to prove that if at some step cops occupy vertices $v_i, v_{i+1}, \ldots, v_{i+k-1}$ then the robber cannot move to any vertex of set $\bigcup_{j=1}^{i+k-1} N[v_j]$ without being captured after the next move of cops. Clearly, this holds after the first move of cops. Let us consider the i-th move. By the induction assumption, before this move of cops the robber is at some vertex $x \notin \bigcup_{j=1}^{i+k-2} N[v_j]$. If he is going to move to vertex $y \in \bigcup_{j=1}^{i+k-1} N[v_j]$ he has to go along some path of length at most s which does not contain cops. Since $G \subseteq T^p$, the distance between x and y in T is at most ps. Then $y \in \bigcup_{j=i}^{i+k-1} N[v_j]$, i.e that y is adjacent to a vertex occupied by some cop and thus the robber is caught at the next move of cops. □

Lemma 5. *Let G be a connected graph with dominating pair. Then $c_s(G) \leq 5s - 1$.*

Proof. Let u and v be a dominating pair, and P be a shortest u, v-path in G. Then P is the backbone of a spanning caterpillar T in G. Since P is a shortest path, $G \subseteq T^5$. Now we apply Lemma 4. □

Combining Proposition 1 with Lemma 5, we obtain the following result.

Corollary 1. *For every positive integer s, $c_s(G)$ can be computed in time $n^{O(s)}$ on graphs with a dominating pair.*

Corollary 1 yields polynomial time algorithms on many graph classes containing a dominating pair. This include not only interval graphs and cocomparability graphs, but more general class of AT-free graphs. (See [9, 10] for definition and properties of AT-free graphs.)

4 Graphs of bounded cliquewidth

Cliquewidth is a graph parameter that measures in a certain sense the complexity of a graph. This parameter was introduced by Courcelle, Engelfriet, and Rozenberg [11].

Let G be a graph, and k be a positive integer. A k-graph is a graph whose vertices are labeled by integers from $\{1, 2, \ldots, k\}$. We call the k-graph consisting of exactly one vertex labeled by some integer from $\{1, 2, \ldots, k\}$ an initial k-graph. The cliquewidth is the smallest integer k such that G can be constructed from initial k-graphs by means of repeated application of the following three operations:

- Disjoint union (denoted by $\oplus$).
- Relabeling: changing all labels i to j (denoted by $\rho_{i\to j}$).
- Join: connecting all vertices labeled by i with all vertices labeled by j (denoted by $\eta_{i,j}$).

If graph G has cliquewidth k it is possible to construct the expression tree for G. The expression tree is a rooted tree T of the following form:

- The nodes of T are of four types i, $\oplus$, η and ρ.
- Introduce nodes $i(v)$ are leaves of T, corresponding to initial k-graphs with vertices v, which are labeled i.
- A union node $\oplus$ stands for a disjoint union of graphs associated with children.
- A join node $\eta_{i,j}$ with one child is associated with the k-graph, which is the result of join operation for the graph corresponding to the child.
- A relabel node $\rho_{i\to j}$ also with one child is associated with the k-graph, which is the result of relabeling operation for the graph corresponding to the child.
- The graph G is isomorphic to the graph associated with the root of T (with all labels removed).

For node v of T we denote by T_v the subtree of T induced by v and it's descendants, and by G_v is denoted k-graph associated with this node. Clearly, T_v is the expression tree for G_v.

Theorem 3. *Let G be a connected graph with cliquewidth k. Then $c_1(G) \leq k$ and $c_2(G) \leq 2k$.*

Proof. If our graph has one vertex then the statement is trivial. So assume that G contains at least two vertices.

We start with the first bound. Let T be an expression tree for G. We describe a cops strategy, which is constructed by tracing of T starting from the root. The key idea of the cop's strategy is to force the robber stay in vertices of graph G_v, where v is a child of considered node of T.

It is assumed that at the beginning cops occupy some vertices of G. We say that a cop moves to vertex z if he is moved to this vertex by a sequence of moves. In the process of the pursuit cops are assigned to sets of vertices of the graph. Correspondingly, these cops (sets) are called *assigned*, and other cops are called *free*.

Let u be a vertex of T. It is assumed inductively that the robber occupies some vertex of G_u, and that all vertices of $V(G_u)$, which are adjacent to vertices of $V(G) \setminus V(G_u)$, are dominated by assigned cops. Suppose that $S_1, S_2, \ldots, S_r$ are disjoint sets of vertices of G_u, to which cops are assigned. The cop assigned to the set S_i occupies some vertex, which is adjacent to all vertices of this set, and every set has exactly one assigned cop. If u is the root, then $r = 0$. Now we consider different cases.

Case 1. *u is an introduce node.* Since this vertex is dominated by some cop, this case is trivial.

Case 2. *u is a union node.* Let $v_1, v_2, \ldots, v_t$ be the children of u. Since G_u is a disjoint union of $G_{v_1}, G_{v_2}, \ldots, G_{v_r}$, we have that the robber can stay only in vertices of the graph G_{v_i} for some $1 \leq i \leq r$. If for some $j \in \{1, 2, \ldots, r\}$ $S_j \cap V(G_{v_i}) = \emptyset$, then the cop assigned to this set is declared free. For other sets we put $S_j = S_j \cap V(G_{v_i})$. Finally, we put $u = v_i$ and cops proceed with the new list of assigned sets.

Case 3. *u is a join node $\eta_{i,j}$ with the child v.* Let $X \subseteq V(G_u)$ be the set of vertices labeled by i, and $Y \subset V(G_u)$ be the set of vertices labeled by j. If X is not included in the list of assigned sets, then vertex $z \in Y$ is chosen, some free cop is moved to this vertex, and this cop is assigned to X. Similarly, if Y is not included to the list of assigned sets then vertex $z \in X$ is chosen, some free cop is moved to this vertex and is assigned to Y. The game proceeds with the new list of assigned sets for $u = v$.

Case 4. *u is a relabel node $\rho_{i \to j}$ with the child v.* Let $X \subset V(G_u)$ be the set of relabeled vertices. If for some $t \in \{1, 2, \dots, r\}$, $X \subset S_t$, then set S_t is partitioned into X and $S_t \setminus X$, and one additional free cop is moved to a vertex dominating X. This cop is assigned to X and the one that was assigned to S_t is assigned to $S_t \setminus X$. Then cops proceed further with the new list of assigned sets for $u = v$.

By following this strategy, Cop player is guaranteed that at some moment he reaches a position in the game when it is his turn to make a move and that the robber occupies a vertex of some assigned set. Since each of the assigned vertices is dominated by a cop, it follows that at some moment Cop player can win the game by catching the robber.

Let us prove that k cops are sufficient to perform this strategy. We use here the following property: For every $u \in V(T)$ with assigned sets $S_1, S_2, \dots, S_r$, no label is used on vertices from two different sets. This property can be shown by inductive arguments. By definition, it holds when u is the root of T. Suppose that after some step of the pursuit two different sets S_i and S_j have vertices with same label. But it means that in the process of construction of G from G_u these sets have to be subjected to relabeling and join operations simultaneously. Then all vertices of these sets should be included into one assigned set after some join operation. Thus $r \le k$, which yields that $c_1(G) \le k$.

The second bound is proved similarly. Main difference is that we assign not one but two cops to a set. Let u be a vertex of T. For the case $s = 1$ cops were able to succeed by dominating all vertices of $V(G_u)$, which are adjacent to vertices of $V(G) \setminus V(G_u)$. In the case $s = 2$, this is not sufficient and cops also have to control all vertices of $V(G) \setminus V(G_u)$, which are adjacent to vertices of $V(G_u)$. Except this, the proof of this bound is almost identical to the case of $s = 1$ and we omit it here. □

In combination with Proposition 1, Theorem 3 implies that

Corollary 2. *For every graph G of bounded cliquewidth the numbers $c_1(G)$ and $c_2(G)$ can be computed in polynomial time.*

Let us remark that the results of this section cannot be extended for $s \ge 3$ because $c_s(G)$ is not bounded by the cliquewidth of a graph. Consider, for example, complete n-partite graph with partition sets $V_1, V_2, \dots, V_n$, $|V_i| = n$ for every $i \in \{1, 2, \dots, n\}$. Then we add n vertices $v_1, v_2, \dots, v_n$ and for every $i \in \{1, 2, \dots, n\}$ make v_i adjacent to all vertices from V_i. Let G_n be the resulting graph. It is easy to see that this graph has cliquewidth at most 3 and that $c_s(G_n) = n$ for $s \ge 3$.

5 Open problems

Many interesting algorithmic question around Cops and Robbers game remain open and we conclude with asking some of them.

- The most challenging question is due to Goldstein and Reingold in [17]: Is the testing of $c_1(G) \leq k$ EXPTIME-complete? If the answer is ”yes”, is the problem EXPTIME-complete for every fixed s? Can it be so that for large s, say for $s \geq \sqrt{n}$, the problem is in NP?
- We have shown that for every graph G of bounded cliquewidth and $s \leq 2$, the number $c_s(G)$ can be computed in polynomial time. What is the computational complexity of the problem on graphs of bounded cliquewidth for $s = 3$ or for $s = \infty$?
- For a graph G of treewidth k, for every $s \geq 1$, it is possible to prove that $c_s(G) \leq k + 1$, which implies that $c_s(G)$ can be computed in time $n^{O(k)}$. What is the parameterized complexity of computing c_s with the treewidth (or the cliquewidth) of a graph as a parameter?
- In the proof of Theorem 1, for a given graph G on n vertices, we construct a graph G' on $O(n^{10})$ vertices such that $\gamma(G) = c_s(G')$, where $\gamma(G)$ is the domination number of G. Combined with the non-approximability for dominating set problem [30], this implies the following

  Corollary 3. *There is a constant* $c > 0$ *such that there is no polynomial time algorithm to approximate* $c_s(G)$ *within a multiplicative factor* $c \log n$*, unless* $P = NP$.

  An interesting question here is if there is an $n^{1-\varepsilon}$-approximation algorithm for the Cops and Robbers game.
- We have shown that for every fixed s, the solution of the Cops and Robbers game can be solved in polynomial time on interval graphs. Can $c_\infty(G)$ be computed in polynomial time on interval graphs?

References

1. A. Adachi, S. Iwata, and T. Kasai, *Some combinatorial game problems require $\Omega(n^k)$ time*, J. ACM, 31 (1984), pp. 361–376.
2. M. Aigner and M. Fromme, *A game of cops and robbers*, Discrete Appl. Math., 8 (1984), pp. 1–11.
3. B. Alspach, *Searching and sweeping graphs: a brief survey*, Matematiche (Catania), 59 (2006), pp. 5–37.
4. T. Andreae, *Note on a pursuit game played on graphs*, Discrete Appl. Math., 9 (1984), pp. 111–115.
5. T. Andreae, *On a pursuit game played on graphs for which a minor is excluded*, J. Combin. Theory Ser. B, 41 (1986), pp. 37–47.
6. A. Berarducci and B. Intrigila, *On the cop number of a graph*, Adv. in Appl. Math., 14 (1993), pp. 389–403.

7. H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoretical Computer Science, 209 (1998), pp. 1–45.
8. B. Bollobás, *Extremal graph theory*, vol. 11 of London Mathematical Society Monographs, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1978.
9. A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph classes: a survey*, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.
10. D. G. Corneil, S. Olariu, and L. Stewart, *Asteroidal triple-free graphs*, SIAM J. Discrete Math., 10 (1997), pp. 399–430.
11. B. Courcelle, J. Engelfriet, and G. Rozenberg, *Context-free handle-rewriting hypergraph grammars.*, in Graph-Grammars and Their Application to Computer Science, H. Ehrig, H.-J. Kreowski, and G. Rozenberg, eds., vol. 532 of Lecture Notes in Computer Science, Springer, 1990, pp. 253–268.
12. R. G. Downey and M. R. Fellows, *Parameterized complexity*, Springer-Verlag, New York, 1999.
13. S. L. Fitzpatrick and R. J. Nowakowski, *Copnumber of graphs with strong isometric dimension two*, Ars Combin., 59 (2001), pp. 65–73.
14. J. Flum and M. Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
15. P. Frankl, *Cops and robbers in graphs with large girth and Cayley graphs*, Discrete Appl. Math., 17 (1987), pp. 301–305.
16. P. Frankl, *On a pursuit game on Cayley graphs*, Combinatorica, 7 (1987), pp. 67–70.
17. A. S. Goldstein and E. M. Reingold, *The complexity of pursuit on a graph*, Theoret. Comput. Sci., 143 (1995), pp. 93–112.
18. J. Gustedt, *On the pathwidth of chordal graphs*, Discrete Appl. Math., 45 (1993), pp. 233–248.
19. G. Hahn and G. MacGillivray, *A note on k-cop, l-robber games on graphs*, Discrete Math., 306 (2006), pp. 2492–2497.
20. Y. O. Hamidoune, *On a pursuit game on Cayley digraphs*, European J. Combin., 8 (1987), pp. 289–295.
21. K. M. Krishnan, R. Singh, L. S. Chandran, and P. Shankar, *A combinatorial family of near regular LDPC codes*, ArXiv Computer Science e-prints, cs/0609146, (2006).
22. M. Maamoun and H. Meyniel, *On a game of policemen and robber*, Discrete Appl. Math., 17 (1987), pp. 307–309.
23. S. Neufeld and R. J. Nowakowski, *A vertex-to-vertex pursuit game played with disjoint sets of edges*, in Finite and infinite combinatorics in sets and logic (Banff, AB, 1991), vol. 411 of NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., Kluwer Acad. Publ., Dordrecht, 1993, pp. 299–312.
24. R. Niedermeier, *Invitation to fixed-parameter algorithms*, vol. 31 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2006.
25. R. Nowakowski and P. Winkler, *Vertex-to-vertex pursuit in a graph*, Discrete Math., 43 (1983), pp. 235–239.
26. S.-L. Peng, M.-T. Ko, C.-W. Ho, T.-s. Hsu, and C. Y. Tang, *Graph searching on some subclasses of chordal graphs*, Algorithmica, 27 (2000), pp. 395–426.
27. A. Quilliot, *A short note about pursuit games played on a graph with a given genus*, J. Combin. Theory Ser. B, 38 (1985), pp. 89–92.
28. A. Quilliot, *Some results about pursuit games on metric spaces obtained through graph theory techniques*, European J. Combin., 7 (1986), pp. 55–66.
29. V. Raman and S. Saurabh, *Short cycles make w-hard problems hard: Fpt algorithms for w-hard problems in graphs with no short cycles*, Accepted for publication in Algorithmica.
30. R. Raz and S. Safra, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, in STOC, 1997, pp. 475–484.
31. B. S. W. Schroeder, *The copnumber of a graph is bounded by* $\lfloor \frac{3}{2}$ *genus* $(G)\rfloor + 3$, in Categorical perspectives (Kent, OH, 1998), Trends Math., Birkhäuser Boston, Boston, MA, 2001, pp. 243–263.

32. P. D. Seymour and R. Thomas, *Graph searching and a min-max theorem for tree-width*, J. Combin. Theory Ser. B, 58 (1993), pp. 22–33.