



Pursuing a fast robber on a graph[☆]

Fedor V. Fomin^a, Petr A. Golovach^{a,*}, Jan Kratochvíl^b, Nicolas Nisse^c, Karol Suchan^{d,e}

^a Department of Informatics, University of Bergen, N-5020 Bergen, Norway

^b Institute for Theoretical Computer Science, Department of Applied Mathematics, Charles University, Praha, Czech Republic

^c MASCOITE, INRIA, I3S, CNRS, UNS, Sophia Antipolis, France

^d Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Santiago, Chile

^e Faculty of Applied Mathematics, AGH - University of Science and Technology, Cracow, Poland

ARTICLE INFO

Article history:

Received 17 October 2008

Received in revised form 28 September 2009

Accepted 13 December 2009

Communicated by G. Italiano

Keywords:

Pursuit-evasion game on graphs

Cops and Robbers

Complexity

Parameterized complexity

Cliquewidth

Planar graph

ABSTRACT

The Cops and Robbers game as originally defined independently by Quilliot and by Nowakowski and Winkler in the 1980s has been much studied, but very few results pertain to the algorithmic and complexity aspects of it. In this paper we prove that computing the minimum number of cops that are guaranteed to catch a robber on a given graph is NP-hard and that the parameterized version of the problem is W[2]-hard; the proof extends to the case where the robber moves s time faster than the cops. We show that on split graphs, the problem is polynomially solvable if $s = 1$ but is NP-hard if $s = 2$. We further prove that on graphs of bounded cliquewidth the problem is polynomially solvable for $s \leq 2$. Finally, we show that for planar graphs the minimum number of cops is unbounded if the robber is faster than the cops.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Cops and Robbers is a pursuit-evasion game with two players: \mathcal{C} (Cops) and \mathcal{R} (Robber) who 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 puts cops on these vertices. Then \mathcal{R} puts the robber on a vertex. The players then take turns starting with \mathcal{C} . At every move each of the cops can either move to an adjacent vertex or not move at all (several cops can occupy the same vertex at some move); \mathcal{R} responds by moving the robber to some vertex along a path of length at most s not containing vertices occupied by cops. In other words, the cops are moving with a unit speed and the speed of the robber is s , and the robber cannot run through a vertex occupied by a cop. When s is unbounded ($s = \infty$), it means that at his turn the robber can move anywhere in the connected component of $G \setminus X$ containing his current position, where X is the subset of vertices occupied by the cops. We say that a cop catches the robber at some move if after 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 the graph G against the robber moving at the speed of s .

The variant of the game with $s = 1$, i.e. when the cops and the robber have the same speed, has been studied intensively. The game was defined (for one cop) by Winkler and Nowakowski [33] and Quilliot [36] who also characterized graphs with 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 the 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

[☆] Preliminary extended abstracts of this paper appeared in the proceedings of TCS'08 [18] and WG'08 [32].

* Corresponding author. Tel.: +44 (0) 1913342510.

E-mail address: petr.golovach@durham.ac.uk (P.A. Golovach).

¹ Corresponding address: School of Engineering and Computing Sciences, Durham University, South Road, DH1 3LE Durham, UK.

cops can always catch the robber. This result can be generalized to graphs of bounded genus [35,39]. 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 [4,9,11,16,20,21,26,29,30] (see also the surveys [3,19,24]).

There is a resemblance of the Cops and Robbers game, at least for large values of $s \rightarrow \infty$, to the helicopter search game defined by Seymour and Thomas [40], which is the game-theoretic interpretation of the well-known treewidth parameter. In the Seymour–Thomas game the robber can move arbitrarily fast and players make their moves simultaneously. See the survey of Bodlaender for an overview of pursuit–evasion games related to treewidth [7].

Despite 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 (implicit in [33]) 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 $n^{O(k)}$ (thus polynomial for fixed k) [6,25] (see also [22]).

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 to 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 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 the evader (robber) on an n -vertex graph can be done in time $f(k) \cdot n^{O(1)}$ (we refer to Bodlaender’s survey [7] for examples of such problems).

There is a feature of the problem we find interesting and which distinguishes the problem from many known “hard” problems. Meyniel conjectured that $c_1(G) \in O(\sqrt{n})$ for graphs on n vertices. It was proved by Bollobás et al. [9] that this conjecture essentially holds for sparse random graphs. Currently best upper bound for general graphs was given by Chiniforooshan [11], who has shown that there is a constant $c > 0$ such that for any graph G on n vertices, $c_1(G) \leq c \cdot \frac{n}{\log n}$. By trying all possible values of k , $1 \leq k \leq c \cdot \frac{n}{\log n}$, and using Proposition 1, we can compute the number $c_1(G)$ in time

$$\sum_{1 \leq k \leq c \cdot \frac{n}{\log n}} \binom{n+k-1}{k}^2 \cdot n^{O(1)}. \quad (1)$$

Assuming for simplicity that $c \cdot \frac{n}{\log n}$ is an integer, it can be easily seen that the sum above is bounded by $c \cdot \frac{n}{\log n} \cdot \binom{2n}{c \cdot n/\log n}^2 \cdot n^{O(1)}$. By making use of Stirling formula and trivial calculations, it is possible to prove that

$$\binom{2n}{c \cdot n/\log n} \leq \frac{e^{\frac{1}{24n}} \cdot c \log n}{\sqrt{\pi \cdot n(2c \log n - 1)}} \left[\frac{4}{(c/\log n)^{c/\log n} \cdot (2 - c/\log n)^{2-c/\log n}} \right]^n = 2^{o(n)}.$$

Hence the sum (1) is bounded by $2^{o(n)} \cdot n^{O(1)}$ and thus there is a *subexponential* time algorithm computing $c_1(G)$.

While the hardness result proved in this paper does not collapse any of the widely believed complexity hierarchies, we still find it quite interesting, because most of the natural NP-hard problems are believed not to be solvable in subexponential time [27].

There are several variants of similar games, for example the k -pebbles game, or the cat and k -mouse game, whose 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 mice), and the proofs are strongly based on these specific properties. Following this line of research, Goldstein and Reingold [22] 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 the cops and the robber are given their initial positions is also EXPTIME-complete. They conjectured that the game on undirected graphs (for $s = 1$) is also EXPTIME-complete. Again, their proofs strongly rely 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 and organization of the paper

In Section 2 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, i.e. a Fixed Parameter Tractable (FPT), 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 [15,17,31] for information on parameterized complexity.

For $s \geq 2$, the hardness results can be refined for very restricted classes of graphs. In Section 3, we show that 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 the 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 [40], the minimum number of cops equals treewidth plus one and can be easily calculated for chordal graphs. For node searching (see [23]), the corresponding problem can be solved in polynomial time for split graphs, but remains NP-complete on chordal graphs. See also [34] for related results. Note also that, for $s = 1$, one cop can always capture the robber on a chordal graph [36]. By continuing to investigate 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. In Section 4, 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 [13], this is not the case for the Cops and Robbers problem. Our proof is based on combinatorial bounds and Proposition 1. Finally, in Sections 5 and 6, we consider the game on planar graphs. The behavior of the game for $s \geq 2$ is very different from the game for $s = 1$. In particular, while $c_1(G) \leq 3$ for planar graphs [2] (and the cop number of the $n \times n$ -grid is 2 for $n \geq 2$), the value of $c_2(G)$ can be arbitrarily large. We prove that for $s = 2$, $c_2(G) \in \Omega(\sqrt{\log n})$ cops are necessary to capture a robber in the $n \times n$ square-grid. In Section 6, we extend this result by showing that for $s = 2$, every planar graph H that contains the $n \times n$ -grid with $n \in \Omega(2^{k^2})$ as an induced subgraph has cop number at least k . We conclude with open problems in Section 7. Preliminary version of these result appeared in [18,32].

2. Cops and robbers is NP-hard

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 (recall that the girth of a graph is the length of its shortest cycle) at least six and large minimum vertex degree with some additional properties. 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 geometric or algebraic. For our reduction we use an algorithmic construction based on the construction of Krishnan et al. [28].

For positive integers n, m , and r , we construct a bipartite graph $H(n, m, r)$ with rmn^2 edges and a bipartition (X, Y) , $|X| = |Y| = nm$. The 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 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 an empty graph on vertices $X \cup Y$ and adds 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  $S := \{z \in W_j : \text{dist}(x, z) > 1 \text{ and } \deg_{i,j}(z) < t + 1\}$ ;
        select a vertex  $y \in S$  such that  $\text{dist}(x, y) = \max_{z \in S} \text{dist}(x, z)$ ; 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  $S := \{z \in U_i : \text{dist}(y, z) > 1 \text{ and } \deg_{i,j}(z) < t + 1\}$ ;
        select a vertex  $x \in S$  such that  $\text{dist}(x, y) = \max_{z \in S} \text{dist}(x, z)$ ; 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 the 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 the 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 [28], 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 $i, j \in \{1, 2, \dots, n\}$;
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 $i, j \in \{1, 2, \dots, n\}$.

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

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

Lemma 2 is a simplified version of the lemma 2 of [28] 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 the 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 Lemmas 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 an odd phase t , and $x \in U_i, y \in W_j$. Let $D = \{z \in W_j : \text{dist}(x, z) \geq g\}$. Since vertex x has 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 $\text{dist}(x, z) \leq g - 1 = 2p - 1$, 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

$$\begin{aligned} & n(r+1) + n(r+1)(n(r+1)-1)^2 + \dots + n(r+1)(n(r+1)-1)^{2(p-1)} \\ &= n(r+1) \frac{(n(r+1)-1)^{2p}-1}{(n(r+1)-1)^2-1}. \end{aligned}$$

Thus

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

which yields $g = 2p \geq 6$. \square

2.2. Proof of Theorem 1

Now we are ready to proceed with the proof of the main result of this section. Recall that a *dominating set* for a graph G is a subset $D \subseteq V(G)$ such that every vertex of G is either in D , or is adjacent to a vertex of D . We use a 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 of cardinality at most k ?

Let G be a graph with 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

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

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 a vertex is dominated by a 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 on the vertices of S dominate all vertices of G' , for every vertex choice of the robber, he will be caught after the first move of the 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. The 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 a 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 becomes dominated, then the robber does 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 a vertex $v_j \in V(G)$ which is not dominated by the 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 a 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 $c_1(G) = 1$ on a superclass of chordal graphs [33]. Also, the treewidth of a chordal graph can be computed in polynomial time, and thus the search game of Seymour–Thomas [40] is tractable on chordal graphs. However, for $s \geq 2$, the problem of computing $c_s(G)$ 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 [37].

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 m new 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 adjacent to all vertices in $N(v_i)$. Then we add m vertices forming a set W to the clique (i.e. these vertices are joined by edges with each other and with all 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_j 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 $j \in \{1, 2, \dots, p\}$ such that vertices of V_j are not dominated by cops standing on vertices of S . The vertex x has at least $k + 1$ adjacent vertices in W . So there is a 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 the graph $H(1, m, r)$ has 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 with an opportunity to avoid capture.

To establish the parameterized complexity on split graphs, we observe that the parameterized version of the dominating set problem remains $W[2]$ -hard on split graphs and that the reduction from DOMINATING SET described above is an FPT reduction. \square

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 an integer s , $c_s(G)$ can be computed in polynomial time. Actually, the only property of interval graphs we need is the existence of dominating pairs. A *dominating pair* in a connected graph G is a pair of (not necessarily 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 an integer p , the p th power of G , G^p is the graph on vertex set $V(G)$; 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 G be a graph with a spanning caterpillar T , 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 $P = (v_1, v_2, \dots, v_r)$ 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 the 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 $v_i, v_{i+1}, \dots, v_{i+k-1}$ then the robber cannot move to any vertex of set $\bigcup_{j=1}^{i+k-1} N[v_j]$ without being captured 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 a 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. y is adjacent to a vertex occupied by some cop and thus the robber is caught at the next move of cops. \square

Lemma 5. *Let G be a connected graph with a 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. \square

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 for many graph classes containing a dominating pair. These include not only interval graphs and comparability graphs, but also the more general class of AT-free graphs (Asteroidal Triple free graphs). See [10,12] 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 [13].

Let G be a graph, and k be a positive integer. A k -graph is a graph whose vertices are labeled by integers from $\{1, 2, \dots, k\}$. We call the k -graph consisting of exactly one vertex labeled by some integer from $\{1, 2, \dots, k\}$ an initial k -graph. The cliquewidth of a graph 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 $\rho_{i \rightarrow j}$).
- Join: connecting all vertices labeled by i with all vertices labeled by j (denoted by $\eta_{i,j}$).

If a graph G has cliquewidth k , then there is an *expression tree* for G which corresponds to the procedure of constructing of k -graph isomorphic to G by the described operations. 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 $\rho_{i \rightarrow j}$ 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).

Notice that a *node* refers to the vertices of T , by opposition to the *vertices* of G . For a node v of T , we denote by T_v the subtree of T rooted at v , and by G_v the k -graph associated with this node. Clearly, T_v is an expression tree for the subgraph G_v of G .

Theorem 3. *Let G be a connected graph of 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 strategy is to force the robber to stay in the vertices of the graph G_v , where v is a child of the node of T under consideration.

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

Let u be a node 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, \dots, 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 of T , then $r = 0$. Now we consider different cases.

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

Case 2. u is a *union node*. Let v_1, v_2, \dots, v_t be the children of u in T . Since G_u is a disjoint union of $G_{v_1}, G_{v_2}, \dots, G_{v_t}$, we have that the robber can stay only in vertices of the graph G_{v_i} for some $1 \leq i \leq t$. If for some $j \in \{1, 2, \dots, t\}$ $S_j \cap V(G_{v_i}) = \emptyset$, 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 the cops proceed with the new list of assigned sets.

Case 3. u is a *join node* $\eta_{i,j}$ with child v . Let $X \subseteq V(G_u)$ be the set of vertices labeled by i , and $Y \subseteq V(G_u)$ be the set of vertices labeled by j . If X is not included in the list of assigned sets, then a 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 in 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* $\rho_{i \rightarrow j}$ with child v . Let $X \subseteq V(G_u)$ 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 node $u \in V(T)$ 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 this 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 $r \leq k$, which yields that $c_1(G) \leq k$.

The second bound is proved similarly. The main difference is that we assign not one but two cops to a set. Let u be a node 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 for this, the proof of this bound is almost identical to the case of $s = 1$ and we omit it here. \square

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 $s \geq 3$ because $c_s(G)$ is not bounded by the cliquewidth of a graph. Consider, for example, a 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 $s \geq 3$.

5. Fast robber in grids

This section is devoted to the proof of the following theorem. This result must be put in contrast with the fact that $c_1(G) \leq 2$ for any square-grid G .

Theorem 4. For any $n \times n$ square-grid G , $c_2(G) \in \Omega(\sqrt{\log(n)})$.

To prove Theorem 4, we propose an evasion strategy for a robber with speed 2 against $k \geq 1$ cops in any $n \times n$ square-grid, where $n \in \Omega(2^{k^2})$. In the following, $k \geq 1$ is fixed and we fix an ordering of the k cops: cop_1, \dots, cop_k . Intuitively, the strategy we design for the robber is defined recursively. There are $k + 1$ levels in our strategy. For any $0 < i \leq k$, the level- i strategy uses the level- $(i - 1)$ strategy as a subroutine. The key point is that the level- i strategy only deals with i cops: cop_1, \dots, cop_i , and maintains as an invariant the fact that cop_i remains "far enough" from the robber.

Let us start with some definitions.

5.1. Recursive partition of a grid

Let $a > 0$ and $b > 2$ be two constants whose values will be specified later. We define two sequences of integers $(zoom_1, \dots, zoom_{k+1})$ and $(size_0, \dots, size_k)$ and $size_0 = 2$. For every $1 \leq i \leq k$, we set

$$zoom_i = ab^i,$$

and

$$size_i = size_0 \cdot \prod_{j=1}^i zoom_j.$$

Finally, let $zoom_{k+1} = 2$ and $n = size_k \cdot zoom_{k+1} = 4 \cdot a^k \cdot b^{k(k+1)/2} \in \Omega(2^{k^2})$. In the following, G denotes the $n \times n$ square-grid, and n is the size of the grid.

The main idea behind our analysis is to fix a recursive partition of G into gradually smaller subgrids of levels k down to 0 . That is, we give a fractal-like structure to G which consists of $4 = (zoom_{k+1})^2$ vertex-disjoint square-grids of size $size_k$. These four $size_k \times size_k$ square-grids are called the k -subgrids of G . Then, recursively, for $i = k$ down to 1 , a i -subgrid consists of $(zoom_i)^2$ vertex-disjoint square-grids of size $size_{i-1}$. These subgrids are the $(i - 1)$ -subgrids of G . Therefore, for any $i \leq k$, G consists of $(\prod_{j=i+1}^{k+1} zoom_j)^2$ vertex-disjoint i -subgrids.

Let us introduce a coordinate system for subgrids at each level i , $0 \leq i \leq k$. The coordinates of an i -subgrid H are $(abs(H), ord(H))$, which correspond to the row (bottom-up) and column (left-right) occupied by H in the partition of G into i -subgrids. In other words, a vertex v is in H if and only if the abscissa (column) of v is between $(abs(H) - 1) \cdot size_i + 1$ and $abs(H) \cdot size_i$, and the ordinate (row) of v is between $(ord(H) - 1) \cdot size_i + 1$ and $ord(H) \cdot size_i$. Let $\ell \geq 1$ and let H^i be an i -subgrid of G . We note by $around_\ell(H^i)$ the subgrid induced by the i -subgrids H , such that $|ord(H^i) - ord(H)| \leq \ell$ and $|abs(H^i) - abs(H)| \leq \ell$.

Notation. In the following, $around_1(H^i)$ will be denoted by $around(H^i)$. Moreover, we define a new sequence of integers $(margin_1, \dots, margin_k)$, which is a sequence of safety distances. And for any i , $0 \leq i < k$, and any i -subgrid H^i , $margin(H^i)$ will denote $around_{margin_i}(H^i)$.

Definition 1. For any $i \leq k$, cop_i is far enough from a $(i - 1)$ -subgrid R if this cop does not occupy a vertex in $around(R)$.

Definition 2. For any $i \leq k$, cop_i is not too close to a $(i - 1)$ -subgrid R if this cop does not occupy a vertex in $margin(R)$.

For any $i, j \leq k$, an i -subgrid of G is adjacent to a j -subgrid if they are vertex-disjoint and there is an edge of G that is incident to a vertex of each of them. A path of i -subgrids is a sequence (G_1, \dots, G_r) of i -subgrids of G such that G_j is adjacent to G_{j+1} , $1 \leq j < r$. The length of such a path is simply the number of i -subgrids. In the following, we will recursively design a strategy of the robber in terms of paths of i -subgrids. More precisely, a level- i strategy for the robber will be a path of $(i - 1)$ -subgrids ($i \geq 1$). That is, when the robber is occupying a vertex of some i -subgrid R and need to reach a vertex of an i -subgrid R' adjacent to R , the robber will follow a path P of $(i - 1)$ -subgrids. Recursively, to go from one $(i - 1)$ -subgrid of P to an adjacent one, the robber will follow the level- $(i - 1)$ strategy, i.e., a path of $(i - 2)$ -subgrids, and so on.

Notation. At any step t of the game, the i -subgrid occupied by the robber at this step, i.e., the i -subgrid that contains the vertex occupied by the robber, is denoted by R_t^i (or R^i if no confusion can occur).

The next definition is crucial in our proof. Somehow, it describes a position of the robber that is safe with respect to a subset of the cops.

Definition 3. Let $i \leq k$. The robber occupies an i -nice position if, for all $1 \leq j \leq i$, cop_j is not too close to R^{j-1} and $margin(R^{j-1})$ is contained in R^j . Any position is a 0-nice position.

As we said, a level- i strategy for the robber is a path of $(i - 1)$ -subgrids that the robber must follow to go from an i -subgrid to an adjacent one. More precisely, the goal of a strategy of level- i consists of the following. Given that the robber is occupying an i -nice position in some i -subgrid H of G , and given any i -subgrid D adjacent to H , a level- i strategy will be defined as a path of $(i - 1)$ -subgrids that allows the robber to reach an i -nice position in D , in such a way that cop_i always remains far enough from the robber, i.e., cop_i never enters $around(R_t^{i-1})$. The key (and maybe counterintuitive) point when defining a level- i strategy is that it is defined considering only the position and the moves of cop_i . Actually, for any $j < i$, cop_j is taken into account by the level- j strategy used inductively in the definition of the level- i strategy. However, for any $j \leq k$, a level- j strategy does not care about cop_{j+1}, \dots, cop_k , and it is defined as if they did not exist.

The last constraint we need to impose on a level- i strategy concerns its duration. For our purpose, we need the robber to go from a i -nice position in some i -subgrid into an i -nice position in any adjacent i -subgrid “quickly”. Hence, we define a new sequence of integers $(time_0, \dots, time_k)$ which is a sequence of numbers of rounds. For any $i \leq k$, $time_i$ is an upper bound on the numbers of rounds required by the robber following a level- i strategy, in order to go from an i -nice position on an i -subgrid to an i -nice position on a neighboring i -subgrid. Because the robber has speed 2 and $size_0 = 2$, $time_0 = 1$. Note that $time_i \geq zoom_i \cdot time_{i-1}$, indeed, if the robber goes straight ahead, it may cross at least $zoom_i (i - 1)$ -subgrids to go from an i -subgrid to an adjacent one. Actually, following our level- i strategy, the robber may cross more $(i - 1)$ -subgrids in order to evade cop_i . We define a new sequence of integers $(detour_1, \dots, detour_k)$ which is a sequence of extra distances. For any $i \leq k$, $detour_i$ is an upper bound on the number of additional $(i - 1)$ -subgrids that the robber (following the level- i

strategy) needs to travel in order to cross a i -subgrid. More precisely, starting from an i -nice position, the length of the path of $(i - 1)$ -subgrids (i.e., the number of $(i - 1)$ -subgrids of such a path) that the robber will follow to go into an i -nice position in a neighboring i -subgrid is upper bounded by $zoom_i + detour_i$. In other words, $time_i = (zoom_i + detour) \cdot time_{i-1}$. The robber is fast if he goes from an i -subgrid to an adjacent one in less rounds than a cop, i.e., if $time_i < size_i$.

5.2. Some equalities and technical lemmas

In this section, we specify the relationships between the sequences of integers we introduced in the previous section. We also prove two technical lemmas that will be useful in the proof of the correctness of the strategy we design for the robber.

We first set the relationships between $(size_i)_{i \leq k}$, $(margin_i)_{i \leq k}$, $(time_i)_{i \leq k}$ and $(detour_i)_{i \leq k}$. Recall that $size_0 = 2$ and $time_0 = 1$. For any i , $1 \leq i \leq k$:

$$margin_i = \left\lceil \frac{4 + size_{i-1}/time_{i-1}}{size_{i-1}/time_{i-1} - 1} \right\rceil, \quad detour_i = 2 \cdot \left\lceil \frac{(2 \cdot margin_i + 2)size_{i-1}/time_{i-1}}{size_{i-1}/time_{i-1} - 1} \right\rceil. \tag{2}$$

$$time_i = (zoom_i + detour_i)time_{i-1} \tag{3}$$

From Eqs. (2) and (3), and the fact that $size_i = zoom_i \cdot size_{i-1}$, we get that $size_i/time_i = \frac{zoom_i}{zoom_i + detour_i} \cdot size_{i-1}/time_{i-1} \geq \beta_i \cdot size_{i-1}/time_{i-1}$, where β_i is defined by:

$$\beta_i = \frac{zoom_i}{zoom_i + \frac{2+4 \cdot size_{i-1}/time_{i-1} + 14 \cdot (size_{i-1}/time_{i-1})^2}{(size_{i-1}/time_{i-1} - 1)^2}}$$

Now, we will specify the values of the constants a and b used to define the sequence $(zoom_i)_{i \leq k}$. Let us set $2 > \alpha > 1$, and let $a = \lceil \frac{20}{(\alpha-1)^2} \rceil \cdot \lceil \frac{2}{\ln(2/\alpha)} \rceil$ and let b be an integer such that $b > \max\{2, \frac{\ln(2/\alpha)}{2}\}$. From now on, we assume that a and b satisfy these (in)equalities. For these values of a and b , we can prove the following lemmas.

Lemma 6. For any $0 \leq i \leq k$, $time_i < size_i$.

Proof. We prove by induction on i , $0 \leq i \leq k$, that $2 \geq size_i/time_i > \alpha > 1$.

It is straightforward for $i = 0$. Let $i > 0$, and let us assume that the result is valid for any $j < i$. $size_i/time_i \geq 2 \cdot \prod_{j=1}^i \beta_j$, thus we need to prove that $\prod_{j=1}^i \beta_j > \alpha/2$. Actually, we prove that $1/\prod_{j=1}^i \beta_j < 2/\alpha$. By the induction hypothesis, we get that, for any $0 \leq j \leq i - 1$, $2 \geq size_j/time_j > \alpha$. Thus, $\frac{2+4 \cdot size_{j-1}/time_{j-1} + 14 \cdot (size_{j-1}/time_{j-1})^2}{(size_{j-1}/time_{j-1} - 1)^2} < \lceil \frac{20}{(\alpha-1)^2} \rceil$. Hence, we obtain that $1/\beta_j < (zoom_j + \lceil \frac{20}{(\alpha-1)^2} \rceil)/zoom_j < 1 + 1/(\frac{2}{\ln(2/\alpha)} \cdot b^j)$.

For any i , $0 \leq i \leq k$:

$$\begin{aligned} 2 &> 1 - (1/b)^k \\ \ln(2/\alpha) &> \frac{\ln(2/\alpha)}{2} \cdot \frac{1/b - (1/b)^{k+1}}{1 - 1/b} \quad (\text{because } b > 2) \\ &> \sum_{1 \leq j \leq i} 1/\left(\frac{2}{\ln(2/\alpha)} \cdot b^j\right) \\ &\geq \sum_{1 \leq j \leq i} \ln\left(1 + 1/\left(\frac{2}{\ln(2/\alpha)} \cdot b^j\right)\right) \quad (\text{because } x \geq \ln(1 + x)) \\ &= \ln\left(\prod_{1 \leq j \leq i} \left(1 + 1/\left(\frac{2}{\ln(2/\alpha)} \cdot b^j\right)\right)\right) \\ 2/\alpha &> \prod_{1 \leq j \leq i} \left(1 + 1/\left(\frac{2}{\ln(2/\alpha)} \cdot b^j\right)\right) \\ 2/\alpha &> 1/\prod_{1 \leq j \leq i} \beta_j. \quad \square \end{aligned}$$

Lemma 7. For any $1 \leq i \leq k$, $detour_i + 4 \cdot margin_i + 2 < zoom_i$

Proof. For any $1 \leq i \leq k$, $detour_i + 4 \cdot margin_i + 2$

$$\begin{aligned} &< \frac{24 \cdot (size_{i-1}/time_{i-1})^2 + 4 \cdot size_{i-1}/time_{i-1} - 8}{(size_{i-1}/time_{i-1} - 1)^2} \\ &< \frac{20}{(\alpha-1)^2} \quad (\text{because of Lemma 6}) \\ &< zoom_i = \frac{20}{(\alpha-1)^2} \cdot \frac{2}{\ln(2/\alpha)} \cdot b^i \quad \left(\text{because } b > \max\left\{2, \frac{\ln(2/\alpha)}{2}\right\}\right). \quad \square \end{aligned}$$

5.3. Evasion strategy for the robber

In this section, we describe an evasion strategy for the robber against k cops in the $n \times n$ square-grid $G(n = 4 \cdot a^k \cdot b^{k(k+1)/2})$. More precisely, for any i , $1 \leq i \leq k$, we describe a level- i strategy \mathcal{S}_i , using \mathcal{S}_{i-1} as a subroutine, allowing the robber to evade i cops in some i -subgrid of G . For any i , $0 \leq i \leq k$, Strategy \mathcal{S}_i consists of a path of $(i-1)$ -subgrids that allows the robber to go from one i -subgrid to an adjacent one. The key point is that Strategy \mathcal{S}_i only deals with cop_i . Let $0 \leq i \leq k$. In the following, $d_i = \text{detour}_i/2$, for any $i \leq k$. We now describe Strategy \mathcal{S}_i .

Let R^i be the i -subgrid that is initially occupied by the robber. Let D^i be any i -subgrid adjacent to R^i . For ease of description, we assume that D^i is below R^i , i.e., D^i has smaller ordinate (row) than R^i . At each step $t \geq 0$ of the game, let R_t^{i-1} and C_t^{i-1} be the $(i-1)$ -subgrids occupied by the robber and cop_i respectively.

Roughly speaking, the robber keeps going down toward the next $(i-1)$ -subgrid below it, i.e., in the direction of D^i . Let f be the first step, when C_f^{i-1} becomes adjacent to $\text{margin}(R_f^{i-1})$, if it happens. In this situation, there are two cases. Either cop_i is not below the robber, in which case the robber keeps on going down but it goes a bit further in the same direction. Otherwise, the robber chooses to avoid cop_i by following a path of d_i $(i-1)$ -subgrids to the right or to the left before continuing to go down.

For $i = 0$, \mathcal{S}_0 is any shortest strategy that allows the robber to go from a 0-subgrid to an adjacent one. Note that, because $\text{size}_0 = 2$ and the robber has speed 2, \mathcal{S}_0 takes at most 1 step. Let $0 < i \leq k$. Let O be the greatest ordinate (row) of a $(i-1)$ -subgrid H such that $\text{margin}(H)$ is contained in D^i , and let A be the greatest abscissa (column) of a $(i-1)$ -subgrid H in D^i . For our purpose, we only consider the case when the initial position of the robber is a i -nice position. In particular, $\text{margin}(R_{\text{initial}}^{i-1})$ is contained in R^i .

The strategy proceeds as follows and it is depicted in Fig. 1. In Fig. 1(a)–(c), the hatched zone corresponds to the path of $(i-1)$ -subgrids covered by the robber during the game. There are three cases to be considered.

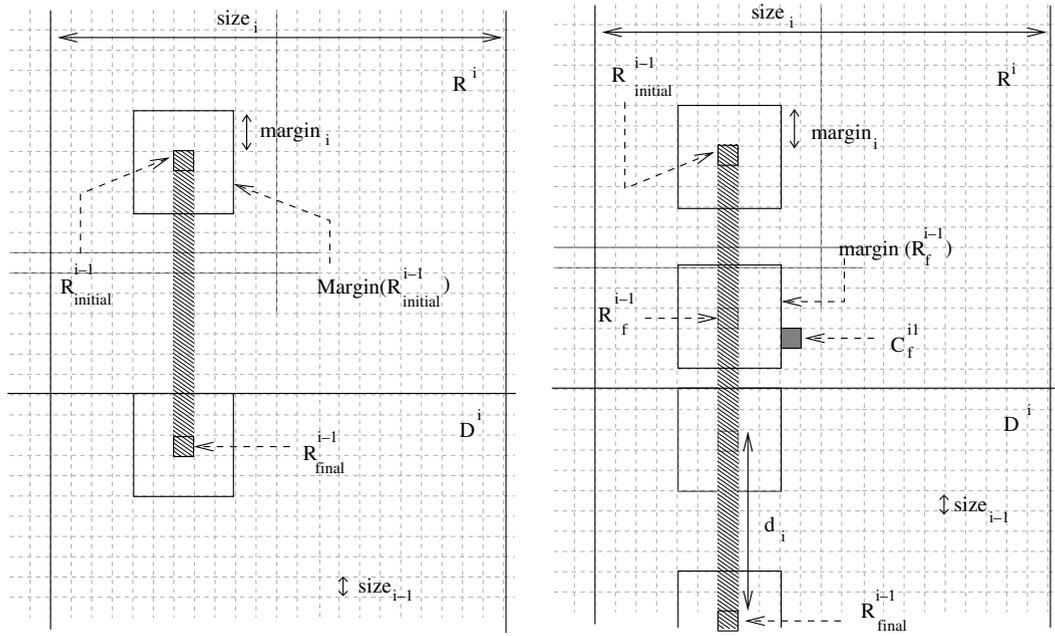
- First, we consider the case when f is not defined or when C_f^{i-1} is above R_f^i , i.e., $\text{ord}_{C_f^{i-1}} = \text{ord}_{R_f^{i-1}} + \text{margin}_i + 1$. In this case, let R_{final}^{i-1} be the $(i-1)$ -subgrid with $\text{abs}(R_{\text{final}}^{i-1}) = \text{abs}(R_{\text{initial}}^{i-1})$ and $\text{ord}(R_{\text{final}}^{i-1}) = O$. Let P be the shortest path of $(i-1)$ -subgrids between R_{initial}^{i-1} and R_{final}^{i-1} . The robber follows P using the level- $(i-1)$ strategy \mathcal{S}_{i-1} to go from one $(i-1)$ -subgrid of P to the next one. This case is depicted in Fig. 1(a). Note that, the length of the path of $(i-1)$ -subgrids followed by such a strategy is at most zoom_i .
- Second, we consider the case when C_f^{i-1} is to the left or to the right of $\text{margin}(R_f^{i-1})$, i.e., $\text{ord}(R_f^{i-1}) + \text{margin}_i \geq \text{ord}(C_f^{i-1}) \geq \text{ord}(R_f^{i-1}) - \text{margin}_i - 1$ and $\text{abs}(C_f^{i-1}) = \text{abs}(R_f^{i-1}) + \text{margin}_i + 1$ or $\text{abs}(C_f^{i-1}) = \text{abs}(R_f^{i-1}) - \text{margin}_i - 1$. In this case, let R_{final}^{i-1} be the $(i-1)$ -subgrid with $\text{abs}(R_{\text{final}}^{i-1}) = \text{abs}(R_{\text{initial}}^{i-1})$ and $\text{ord}(R_{\text{final}}^{i-1}) = O - d_i$. Let P be the shortest path of $(i-1)$ -subgrids between R_{initial}^{i-1} and R_{final}^{i-1} . The robber follows P using the level- $(i-1)$ strategy \mathcal{S}_{i-1} to go from one $(i-1)$ -subgrid of P to the next one. This case is depicted in Fig. 1(b). Note that, the length of the path of $(i-1)$ -subgrids followed by such a strategy is at most $\text{zoom}_i + d_i$.
- Finally, let us consider the case when C_f^{i-1} is below $\text{margin}(R_f^{i-1})$, i.e., $\text{ord}(C_f^{i-1}) = \text{ord}(R_f^{i-1}) - \text{margin}_i - 1$ and $\text{abs}(R_f^{i-1}) - \text{margin}_i \leq \text{abs}(C_f^{i-1}) \leq \text{abs}(R_f^{i-1}) + \text{margin}_i$. In this case, let us first assume that R_f^{i-1} is closer to the left side of R^i (and D^i), i.e., $A - \text{abs}(R_f^{i-1}) \geq \text{zoom}_i/2$. Let R_r^{i-1} be the $(i-1)$ -subgrid with $\text{abs}(R_r^{i-1}) = \text{abs}(R_f^{i-1}) + d_i$ and $\text{ord}(R_r^{i-1}) = \text{ord}(R_f^{i-1})$, and let R_{final}^{i-1} be the $(i-1)$ -subgrid with $\text{abs}(R_{\text{final}}^{i-1}) = \text{abs}(R_r^{i-1})$ and $\text{ord}(R_{\text{final}}^{i-1}) = O - d_i$. Let P_1 be the shortest path of $(i-1)$ -subgrids between R_{initial}^{i-1} and R_r^{i-1} , let P_2 be the shortest path of $(i-1)$ -subgrids between R_r^{i-1} and R_f^{i-1} , and let P_3 be the shortest path of $(i-1)$ -subgrids between R_f^{i-1} and R_{final}^{i-1} . The robber follows $P = P_1 \cup P_2 \cup P_3$ using the level- $(i-1)$ strategy \mathcal{S}_{i-1} to go from one $(i-1)$ -subgrid of P to the next one. This case is depicted in Fig. 1(c). Note that, the length of the path of $(i-1)$ -subgrids followed by such a strategy is at most $\text{zoom}_i + 2 \cdot d_i = \text{zoom}_i + \text{detour}_i$. The case when R_f^{i-1} is closer to the right side of R^i is symmetric, i.e., the single difference comes from $\text{abs}(R_r^{i-1}) = \text{abs}(R_f^{i-1}) - d_i$.

Lemma 8. Let $0 \leq i \leq k$. Let us assume that at some step the robber occupies an i -nice position in some i -subgrid R^i . Assuming that, at any step t of \mathcal{S}_i , the cops cop_j , $j > i$, remain outside around(R_t^i), the strategy \mathcal{S}_i allows the robber to reach an i -nice position in any i -subgrid adjacent to R^i , in at most time_i steps, and such that, at any step t of \mathcal{S}_i , cop_i remains outside around(R_t^{i-1}).

Proof. This lemma gives the main characteristics of \mathcal{S}_i that allow us to prove Theorem 4. The proof is by induction on $i \leq k$.

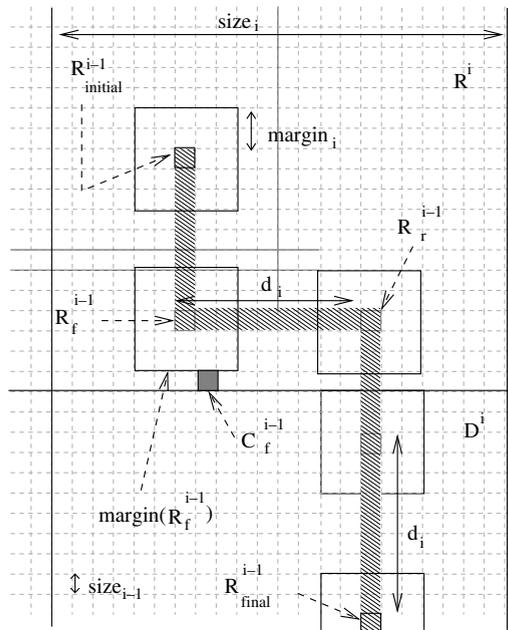
First, let us prove the lemma for $i = 0$. Since any 0-subgrid has size $\text{size}_0 = 2$ and the robber has speed 2, at most $\text{time}_0 = 1 < \text{size}_0$ step is sufficient for the robber to go from any position in a 0-subgrid to any position in an adjacent 0-subgrid.

Let $i > 0$ and assume that the lemma is valid for $i-1$. Note that, while cop_i does not occupy a vertex in around(R_t^{i-1}), the induction hypothesis may be applied. First, the path of $(i-1)$ -subgrids followed by \mathcal{S}_i has length at most $\text{zoom}_i + \text{detour}_i$, therefore, if cop_i always remains outside around(R_t^i), by the induction hypothesis, Strategy \mathcal{S}_i takes at most $\text{time}_i = (\text{zoom}_i + \text{detour}_i)\text{time}_{i-1}$ steps. It remains to prove that cop_i never enters around(R_t^i) and at the last step of \mathcal{S}_i , cop_i is outside of margin(R_t^i).



(a) Illustration of Strategy 1.

(b) Illustration of Strategy 2.



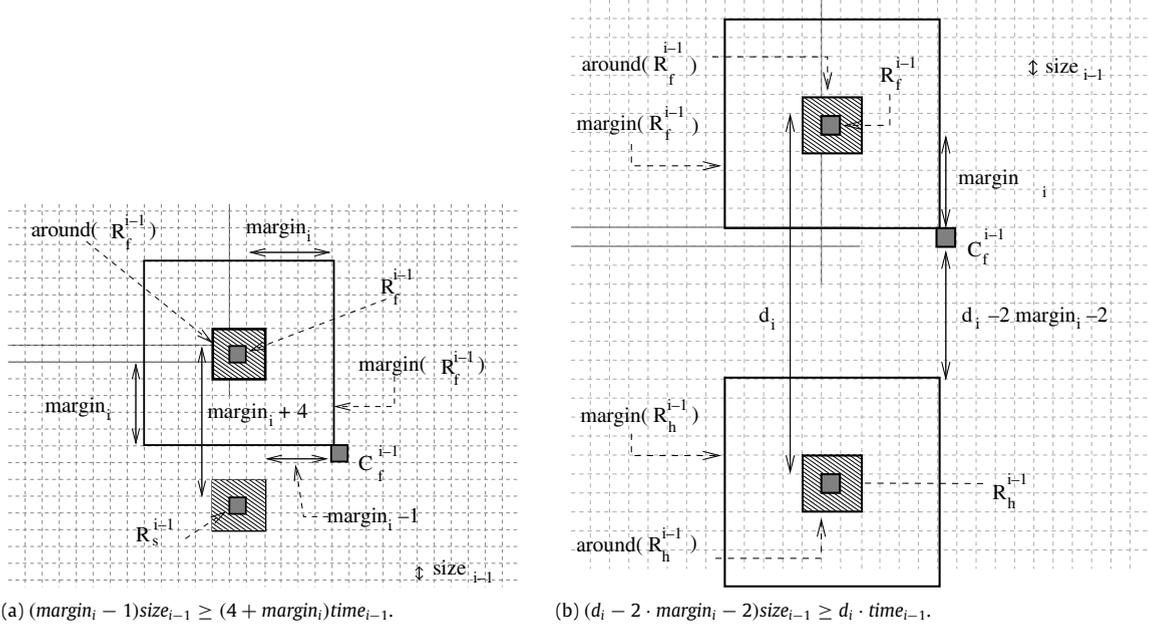
(c) Illustration of Strategy 3.

Fig. 1. Three possible paths of $(i - 1)$ -subgrids followed by δ_i to go from R^i to D^i .

If f is not defined, that is, if cop_i never approaches the robber to occupy a subgrid C_f^{i-1} adjacent to $margin(R_f^{i-1})$, then always remains outside of $margin(R_t^i)$.

Therefore, let us consider the case when f is defined. There are three cases to be considered.

- First, we consider the case when C_f^{i-1} is above R_f^i , i.e., $ord(C_f^{i-1}) = ord(R_f^{i-1}) + margin_i + 1$. To prove that cop_i never enters $around(R_t^{i-1})$, let us apply the induction hypothesis. cop_i needs at least $size_{i-1}$ steps to go from one $(i - 1)$ -subgrid to the one below. By the induction hypothesis, while cop_i does not enter $around(R_t^{i-1})$, at most $time_{i-1}$ steps are sufficient for the robber to go from a $(i - 1)$ -nice position in a $(i - 1)$ -subgrid to a $(i - 1)$ -nice position in the next $(i - 1)$ -subgrid in P . By Lemma 6, $size_{i-1} > time_{i-1}$. Therefore, cop_i never enters $around(R_t^{i-1})$. Moreover, each time the robber reaches a $(i - 1)$ -nice position in a new $(i - 1)$ -subgrid R_t^{i-1} , cop_i is outside of $margin(R_t^{i-1})$. In particular, this is the case at the step



(a) $(margin_i - 1) \cdot size_{i-1} \geq (4 + margin_i) \cdot time_{i-1}$. (b) $(d_i - 2 \cdot margin_i - 2) \cdot size_{i-1} \geq d_i \cdot time_{i-1}$.
Fig. 2. Illustration of Eq. (2): (a) cop_i must never enter $around(R_f^{i-1})$, and (b) the robber must reach a nice position, i.e., cop_i must not enter $margin(R_f^{i-1})$.

when $margin(R_f^{i-1})$ is contained in D^i for the first time. Hence, when the robber reaches a $(i - 1)$ -nice position in R_{final}^{i-1} , he actually occupies an i -nice position.

- Second, we consider the case when C_f^{i-1} is to the right of $margin(R_f^{i-1})$, i.e., $ord(R_f^{i-1}) + margin_i \geq ord(C_f^{i-1}) \geq ord(R_f^{i-1}) - margin_i - 1$ and $abs(C_f^{i-1}) = abs(R_f^{i-1}) + margin_i + 1$ or $abs(C_f^{i-1}) = abs(R_f^{i-1}) - margin_i - 1$.

Let s be some step of the game, and let us prove that cop_i cannot enter $around(R_s^{i-1})$ while the robber is occupying R_s^{i-1} . Indeed, by applying the induction hypothesis, the robber will leave R_s^{i-1} in at most $(ord_{R_f^{i-1}} - ord_{R_s^{i-1}} + 1) \cdot time_{i-1}$ steps after f . Two cases must be considered.

- If $ord(R_s^{i-1}) \geq ord(R_f^{i-1}) - margin_i - 3$, the length of the path of $(i - 1)$ -subgrids between C_f^{i-1} and R_s^{i-1} , is at least $(|abs(R_s^{i-1}) - abs(C_f^{i-1})| - 2)$ (this distance is minimum for $ord(C_f^{i-1}) = ord(R_s^{i-1})$). Therefore, cop_i requires at least $(|abs(R_s^{i-1}) - abs(C_f^{i-1})| - 2) \cdot size_{i-1} = (margin_i - 1) \cdot size_{i-1}$ steps to enter $around(R_s^{i-1})$. In this case, the robber leaves R_s^{i-1} in strictly less than $(margin_i + 4) \cdot time_{i-1}$ steps. By Eq. (2), cop_i cannot enter $around(R_s^{i-1})$ while the robber is occupying it (cf. Fig. 2(a)).
- Otherwise, the distance that cop_i must cover in order to enter $around(R_s^{i-1})$ is minimum when $ord(C_f^{i-1})$ is minimum, that is, $ord(C_f^{i-1}) = ord(R_f^{i-1}) - margin_i - 1$. In this case, cop_i requires at least $(|abs(C_f^{i-1}) - abs(R_s^{i-1})| - 2 + ord(C_f^{i-1}) - ord(R_s^{i-1}) - 2) \cdot size_{i-1} = (margin_i - 1 + ord(R_f^{i-1}) - margin_i - 1 - ord(R_s^{i-1}) - 2) \cdot size_{i-1} = (ord(R_f^{i-1}) - ord(R_s^{i-1}) - 4) \cdot size_{i-1}$ steps to enter $around(R_s^{i-1})$. By Eq. (2) and Lemma 6, we get $(ord(R_f^{i-1}) - ord(R_s^{i-1}) - 4) \cdot size_{i-1} = (ord(R_f^{i-1}) - ord(R_s^{i-1}) - margin - 3) \cdot size_{i-1} + (margin - 1) \cdot size_{i-1} > (ord(R_f^{i-1}) - ord(R_s^{i-1}) - margin - 3) \cdot time_{i-1} + (margin_i + 4) \cdot time_{i-1} = (ord(R_f^{i-1}) - ord(R_s^{i-1}) + 1) \cdot time_{i-1}$. Again, cop_i cannot enter $around(R_s^{i-1})$ while the robber is occupying it.

It remains to prove that the robber reaches an i -nice position in D^i . That is, we prove that cop_i is above $margin(R_{final}^{i-1})$ at the last step $final$ of the game, and $margin(R_{final}^{i-1})$ is a subgraph of D^i . Let top (resp., $right$) be the greatest ordinate (row), respectively abscissa (column), of a $(i - 1)$ -subgrid in D^i . Note that, $ord(R_{final}^{i-1}) = top - margin_i - d_i$ and $abs(R_{final}^{i-1}) = abs(R_f^{i-1})$.

By Lemma 7, $d_i + 2 \cdot margin_i < zoom_i$. Thus, $top - margin_i > ord(R_{final}^{i-1}) > top - zoom_i + margin_i$. Moreover, $margin(R_f^{i-1})$ is a subgraph of R^i , thus $right - zoom_i + margin_i < abs(R_{final}^{i-1}) < right - margin_i$. Therefore, $margin(R_{final}^{i-1})$ is a subgraph of D^i .

The distance that cop_i must cover in order to enter $margin(R_{final}^{i-1})$ is minimum when $ord(C_f^{i-1})$ is minimum, that is, $ord(C_f^{i-1}) = ord(R_f^{i-1}) - margin_i - 1$. In this case, cop_i requires at least $(|abs(C_f^{i-1}) - abs(R_{final}^{i-1})| - 2 + ord(C_f^{i-1}) - ord(R_{final}^{i-1}) - 2) \cdot size_{i-1} = (margin_i - 1 + ord(R_f^{i-1}) - margin_i - 1 - ord(R_{final}^{i-1}) - 2) \cdot size_{i-1} = (ord(R_f^{i-1}) - ord(R_{final}^{i-1}) - 4) \cdot size_{i-1}$ steps to enter $margin(R_{final}^{i-1})$. By Eq. (2) and Lemma 6, we get that $(ord(R_f^{i-1}) - ord(R_{final}^{i-1}) - 4) \cdot size_{i-1} = (ord(R_f^{i-1}) - ord(R_{final}^{i-1}) - d_i + 2 \cdot margin_i + 2) \cdot size_{i-1} + (d_i - 2 \cdot margin_i - 2) \cdot size_{i-1} > (ord(R_f^{i-1}) - ord(R_{final}^{i-1}) - d_i + 2 \cdot margin_i +$

$2) \cdot \text{time}_{i-1} + d_i \cdot \text{time}_{i-1} = (\text{ord}(R_f^{i-1}) - \text{ord}(R_{\text{final}}^{i-1}) + 2 \cdot \text{margin}_i + 2) \cdot \text{time}_{i-1}$. Moreover, by applying the induction hypothesis, at most $(\text{ord}(R_f^{i-1}) - \text{ord}(R_{\text{final}}^{i-1}))\text{time}_{i-1}$ steps are sufficient for the robber to reach an $(i - 1)$ -nice position in R_{final}^{i-1} . Therefore, cop_i does not occupy $\text{margin}(R_{\text{final}}^{i-1})$ at the end of the strategy. Therefore, the robber reaches an i -nice position in D^i .

- Finally, let us consider the case when C_f^{i-1} is below $\text{margin}(R_f^{i-1})$, i.e., $\text{ord}(C_f^{i-1}) = \text{ord}(R_f^{i-1}) - \text{margin}_i - 1$ and $\text{abs}(R_f^{i-1}) - \text{margin}_i \leq \text{abs}(C_f^{i-1}) \leq \text{abs}(R_f^{i-1}) + \text{margin}_i$.

Let *right* be the greatest abscissa (column) of a $(i - 1)$ -subgrid in D^i , and let *final* be the last step of the game. Note that, by Lemma 7, $\text{abs}(R_f^{i-1}) < \text{right} - \text{margin}_i$. Therefore, by the same analysis as in the last item of the previous case, $\text{margin}(R_{\text{final}}^{i-1})$ is a subgrid of D^i .

Then, we prove that, at any step s , cop_i remains outside *around* (R_s^{i-1}) for any $(i - 1)$ -subgrid R_s^{i-1} on the path between R_f^{i-1} and R_r^{i-1} . For this purpose, it is sufficient to observe that the configuration is similar to the previous strategy, by rotating the grid. Moreover, this observation proves that when the robber arrives in R_r^{i-1} , cop_i neither can stand inside $\text{margin}(R_r^{i-1})$, nor block the bottom side of $\text{margin}(R_r^{i-1})$.

Let us show that cop_i cannot block the bottom side of $\text{margin}(R_s^{i-1})$ for any remaining step s of the game. Indeed, the distance that cop_i must cover in order to block the bottom side of $\text{margin}(R_s^{i-1})$ is minimum when $\text{ord}(C_f^{i-1}) = \text{ord}(R_f^{i-1}) - \text{margin}_i - 1$ and $\text{abs}(C_f^{i-1}) = \text{abs}(R_f^{i-1}) + \text{margin}_i$. In this case, cop_i requires at least $S = (\text{abs}(R_s^{i-1}) - \text{abs}(C_f^{i-1}) - \text{margin}_i - 1 + \text{ord}(C_f^{i-1}) - \text{ord}(R_s^{i-1}) + \text{margin}_i) \cdot \text{size}_{i-1}$ steps to block the bottom side of $\text{margin}(R_s^{i-1})$. By Inequality (2) and Lemma 6, $S = (\text{abs}(R_s^{i-1}) - \text{abs}(R_f^{i-1}) - 2 \cdot \text{margin}_i - 2 + \text{ord}(R_f^{i-1}) - \text{ord}(R_s^{i-1})) \cdot \text{size}_{i-1} = (d_i - 2 \cdot \text{margin}_i - 2) \cdot \text{size}_{i-1} + (\text{ord}(R_f^{i-1}) - \text{ord}(R_s^{i-1})) \cdot \text{size}_{i-1} > d_i \cdot \text{time}_{i-1} + (\text{ord}(R_f^{i-1}) - \text{ord}(R_s^{i-1})) \cdot \text{time}_{i-1} > (d_i + \text{ord}(R_f^{i-1}) - \text{ord}(R_s^{i-1})) \cdot \text{time}_{i-1}$ which is the upper bound on the number of steps after f required to leave R_s^{i-1} (by applying the induction hypothesis). Hence, cop_i cannot block the bottom side of $\text{margin}(R_s^{i-1})$ for any $(i - 1)$ -subgrid R_s^{i-1} on the path between R_r^{i-1} and R_{final}^{i-1} . Therefore, the configuration is the same as in the previous cases and the lemma holds. \square

5.4. Proof of Theorem 4

In this section, we prove that, for any $k \geq 1$, one robber with speed 2 can infinitely evade k cops with speed one in any grid of size more than $4a^k b^{k(k+1)/2}$, where a and b are defined as previously. Recall that G is the grid of size $2 \cdot \text{size}_k = 2 \cdot \text{size}_0 \cdot \prod_{1 \leq i \leq k} \text{zoom}_i = 4 \cdot a^k \cdot b^{k(k+1)/2}$. Note that, if one robber can infinitely evade k cops in G , it can perform the same strategy and evade k cops as well in any bigger grid. It remains to prove that the strategy described in the previous section enables the robber to infinitely evade k cops in G .

Now, let us assume that k cops are placed on vertices of G . G is divided into 4 vertex-disjoint subgrids of size size_k (i.e., k -subgrids). Let us fix an ordering of the cops ($\text{cop}_1, \dots, \text{cop}_k$). Choose one of the k -subgrids not occupied by cop_k , and denote it by R^k . Notice that, R^k contains at least four $(k - 1)$ -subgrids $R_1^{k-1}, \dots, R_4^{k-1}$ such that $\text{margin}(R_i^{k-1})$, $1 \leq i \leq 4$, are disjoint and entirely contained in R^k . Any position inside these subgrids is nice at level k . Recursively, choose one not occupied by cop_{k-1} to be R^{k-1} , and proceed until finding R^0 . Any position inside R^0 is k -nice and we may pick it as the initial position for the robber. The top level strategy consists in traversing the four k -subgrids of G along the cycle given by their adjacencies. Lemma 8 (by taking $i = k$) proves that, starting from a k -nice position in some k -subgrid R^k , the robber can reach a k -nice position in any k -subgrid adjacent to R^k , without being caught by the cops. By repeating this process, the robber can infinitely evade k cops in G , which proves Theorem 4.

6. Fast robber in planar graphs

We have proved that the number of cops needed to capture a fast robber in a grid G may be arbitrarily large. Note that the classical transformations of edge removal, vertex removal, and edge contraction do not preserve bounded cop number. Moreover, there are graphs of arbitrarily large treewidth [7] (that is, somehow containing a large grid) and cop number two.

Proposition 2. *For any $k \geq 1$, there is a planar graph H with $c_2(H) \leq 2$, such that a graph G with $c_2(G) \geq k$ can be obtained from H by contracting edges (resp., by removing edges, resp., by removing vertices).*

Sketch of the Proof. We sketch the proof for G obtained from H by contracting edges. Let G be a $n \times n$ square-grid. Let P be the central column (defined in the intuitive way) of G , and let H be the graph obtained by replacing each vertical edge but those of P by a path of length $3n + 7$. The strategy for two cops consists in moving along P from one row to another, until they occupy two consecutive rows L_i and L_{i+1} while the robber is occupying a vertex in a path P' of length $3n + 7$ between those two rows. Then, both cops go toward P' following L_i and L_{i+1} respectively. In at most $\lceil n/2 \rceil$ steps, both cops eventually occupy the same column P'' as the robber. If the robber occupies the path between the cops, he will eventually be caught. Otherwise, the robber is occupying a vertex v of P'' between L_i and L_{i-1} (or, symmetrically, between L_{i+1} and L_{i+2}). Note that, the distance between v and L_i is at most $n + 1$. Finally, the cop that is occupying $P'' \cap L_{i+1}$ goes to $P'' \cap L_{i-1}$. By following L_{i+1}

then P and then L_{i-1} , this cop reaches $P'' \cap L_{i-1}$ in at most $n + 3$ steps. Because the subpath of P'' between L_i and L_{i-1} has length $3n + 7$, the robber still occupies a vertex of this path when one cop is occupying $P'' \cap L_i$ and the other cop is occupying $P'' \cap L_{i-1}$. Therefore, he will eventually be caught. \square

Nevertheless, we can define a larger family of planar graphs of unbounded cop number than the grids themselves. Let size_k be defined as in the previous section.

Theorem 5. *Let H be a planar graph containing an $n \times n$ square-grid G with $n = 4 \cdot \text{size}_k$ as an induced subgraph, then $c_2(H) \geq k$.*

Proof. Notice that if for each of the vertices of degree 2 in G (the “corners”) we contract one incident edge, then we obtain a 3-connected planar graph. By a theorem of Whitney (see Theorem 4.3.2 of [14]), a 3-connected planar graph has only one embedding into the sphere, modulo topological equivalence. So the embedding of G is also unique, and any embedding of the whole H has to respect it. That means that for any two vertices u, v of G , and for any path P between u and v such that all internal vertices of P belong to $H \setminus G$, whatever be the embedding of H into the sphere, only two cases may occur, otherwise there would be a crossing between an edge of P and an edge of G . Either P belongs to the external face of the embedding of G , or P belongs to the face limited by an induced 4-vertices cycle C of G , such that $u, v \in V(C)$. In the latter case, since G is an induced subgraph, P contains at least one vertex in $H \setminus G$.

Notice that G can be partitioned into sixteen subgrids of size size_k . Consider the four of them that are in the center of this partition, and together form a subgrid of size $2 \cdot \text{size}_k$. Denote it by G' . Because H has to respect the embedding of G , there cannot be a path P in $H \setminus G'$ between two vertices of G' strictly shorter than the paths in G' , i.e., G' is an isometric induced subgraph of H .

The escape strategy used in the proof of Theorem 4 can be easily adopted to H , with the robber restricted to stay in G' . The reasoning used in the proof can be easily extended to see that the robber can find a k -nice initial position, and keep moving in order to keep his position k -nice forever. The arguments used to show that the cops cannot get too close to the robber applying our strategy remain valid, since G' preserves the distances. In other words, there are no “short-cuts” available in H . Notice that it is a particular property of our escape strategy, that the absence of short-cuts in $H \setminus G'$ ensures that the robber can still escape in H . Indeed, the strategy of the robber is mainly based on the distance between the robber and the cops. \square

7. Open problems

Many interesting algorithmic questions about the Cops and Robbers game remain open and we conclude with asking some of them.

- We do not know if the decision version $c_s(G) \leq k$ of the problem is in NP and we put it as the first open question. We guess that for small values of s the problem is not in NP—we would find existence of a verifier for the problem that executes in polynomial time quite surprising. The related challenging question is due to Goldstein and Reingold in [22]: Is deciding $c_1(G) \leq k$ EXPTIME-complete? If the answer is “yes”, is the problem EXPTIME-complete for every fixed s ? However, for large s , say for $s \geq \sqrt{n}$, we do not exclude a possibility that there can be a characterization of $c_s(G)$ in terms of some combinatorial graph parameters which can bring the problem in NP. This type of result would be very interesting.
- 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 the 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 [38], 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 $n^{1-\epsilon}$ -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 found in polynomial time on interval graphs. Can $c_\infty(G)$ be computed in polynomial time on interval graphs?
- We proved that if $s \geq 2$ then c_s is unbounded for planar graphs. Can c_s be computed in polynomial time for planar graphs? This question is open even for grids.

Acknowledgement

The first author was supported by the Norwegian Research Council. The third author was supported by Czech research grant 1M0545. The fourth author was supported by CONICYT via Anillo en Redes ACT08 and by the European project 1ST FET AEOLUS. The last author was partially supported by Basal project CMM, Universidad de Chile, Santiago, Chile.

References

- [1] A. Adachi, S. Iwata, T. Kasai, Some combinatorial game problems require $\Omega(n^k)$ time, *J. ACM* 31 (1984) 361–376.
- [2] M. Aigner, M. Fromme, A game of cops and robbers, *Discrete Appl. Math.* 8 (1984) 1–11.
- [3] B. Alspach, Searching and sweeping graphs: A brief survey, *Matematiche (Catania)* 59 (2006) 5–37.
- [4] T. Andreae, Note on a pursuit game played on graphs, *Discrete Appl. Math.* 9 (1984) 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) 37–47.
- [6] A. Berarducci, B. Intrigila, On the cop number of a graph, *Adv. in Appl. Math.* 14 (1993) 389–403.
- [7] H.L. Bodlaender, A partial k -arborescence of graphs with bounded treewidth, *Theoret. Comput. Sci.* 209 (1998) 1–45.
- [8] B. Bollobás, *Extremal Graph Theory*, in: London Mathematical Society Monographs, vol. 11, Academic Press Inc., Harcourt Brace Jovanovich Publishers, London, 1978.
- [9] B. Bollobás, G. Kun, I. Leader, Cops and robbers in a random graph. ArXiv e-prints, <http://arxiv.org/abs/0805.2709v1>, (2008).
- [10] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, in: SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.
- [11] E. Chiniforooshan, A better bound for the cop number of general graphs, *J. Graph Theory* 58 (1) (2008) 45–48.
- [12] D.G. Corneil, S. Olariu, L. Stewart, Asteroidal triple-free graphs, *SIAM J. Discrete Math.* 10 (1997) 399–430.
- [13] B. Courcelle, J. Engelfriet, G. Rozenberg, Context-free handle-rewriting hypergraph grammars, in: H. Ehrig, H.-J. Kreowski, G. Rozenberg (Eds.), *Graph-Grammars and Their Application to Computer Science*, in: Lecture Notes in Computer Science, vol. 532, Springer, 1990, pp. 253–268.
- [14] R. Diestel, *Graph Theory*, third edition, Springer-Verlag, 2005.
- [15] R.G. Downey, M.R. Fellows, *Parameterized Complexity*, Springer-Verlag, New York, 1999.
- [16] S.L. Fitzpatrick, R.J. Nowakowski, Copnumber of graphs with strong isometric dimension two, *Ars Combin.* 59 (2001) 65–73.
- [17] J. Flum, M. Grohe, *Parameterized Complexity Theory*, in: Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.
- [18] F.V. Fomin, P.A. Golovach, J. Kratochvíl, On tractability of Cops and Robbers game, in: Proceedings of the 5th IFIP International Conference on Theoretical Computer Science, TCS 2008, in: *Computer Science*, vol. 237, Springer, Boston, 2008, pp. 171–185.
- [19] F.V. Fomin, D.M. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science* 399 (2008) 236–245.
- [20] P. Frankl, Cops and robbers in graphs with large girth and Cayley graphs, *Discrete Appl. Math.* 17 (1987) 301–305.
- [21] P. Frankl, On a pursuit game on Cayley graphs, *Combinatorica* 7 (1987) 67–70.
- [22] A.S. Goldstein, E.M. Reingold, The complexity of pursuit on a graph, *Theoret. Comput. Sci.* 143 (1995) 93–112.
- [23] J. Gustedt, On the pathwidth of chordal graphs, *Discrete Appl. Math.* 45 (1993) 233–248.
- [24] G. Hahn, Cops, robbers and graphs, *Tatra Mt. Math. Publ.* 36 (2007) 163–176.
- [25] G. Hahn, G. MacGillivray, A note on k -cop, l -robber games on graphs, *Discrete Math.* 306 (2006) 2492–2497.
- [26] Y.O. Hamidoune, On a pursuit game on Cayley digraphs, *European J. Combin.* 8 (1987) 289–295.
- [27] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity, *J. Comput. System Sci.* 63 (2001) 512–530.
- [28] K.M. Krishnan, R. Singh, L.S. Chandran, P. Shankar, A combinatorial family of near regular LDPC codes. ArXiv Computer Science e-prints, cs/0609146, (2006).
- [29] M. Maamoun, H. Meyniel, On a game of policemen and robber, *Discrete Appl. Math.* 17 (1987) 307–309.
- [30] S. Neufeld, 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, in: NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci., vol. 411, Kluwer Acad. Publ., Dordrecht, 1993, pp. 299–312.
- [31] R. Niedermeier, *Invitation to Fixed-parameter Algorithms*, in: Oxford Lecture Series in Mathematics and its Applications, vol. 31, Oxford University Press, Oxford, 2006.
- [32] N. Nisse, K. Suchan, Fast Robber in Planar Graphs, Proceedings of the 34th Workshop on Graph Theoretic Concepts in Computer Science (WG 2008), in: Springer-Verlag Lecture Notes in Computer Science, vol. 5344, 2008, pp. 312–323.
- [33] R. Nowakowski, P. Winkler, Vertex-to-vertex pursuit in a graph, *Discrete Math.* 43 (1983) 235–239.
- [34] S.-L. Peng, M.-T. Ko, C.-W. Ho, T.-s. Hsu, C.Y. Tang, Graph searching on some subclasses of chordal graphs, *Algorithmica* 27 (2000) 395–426.
- [35] A. Quilliot, A short note about pursuit games played on a graph with a given genus, *J. Combin. Theory Ser. B* 38 (1985) 89–92.
- [36] A. Quilliot, Some results about pursuit games on metric spaces obtained through graph theory techniques, *European J. Combin.* 7 (1986) 55–66.
- [37] V. Raman, S. Saurabh, Short cycles make w -hard problems hard: Fpt algorithms for w -hard problems in graphs with no short cycles, *Algorithmica* 52 (2008) 203–225.
- [38] R. Raz, 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.
- [39] B.S.W. Schroeder, The copnumber of a graph is bounded by $\lfloor \frac{3}{2} \text{genus}(G) \rfloor + 3$, in: *Categorical perspectives*, Kent, OH, 1998, in: Trends Math., Birkhäuser Boston, Boston, MA, 2001, pp. 243–263.
- [40] P.D. Seymour, R. Thomas, Graph searching and a min–max theorem for tree-width, *J. Combin. Theory Ser. B* 58 (1993) 22–33.