



**HAL**  
open science

## A cops and robber game and the meeting time of synchronous directed walks

Walid Ben-Ameur, Walid Ben-Ameur, Alessandro Maddaloni

► **To cite this version:**

Walid Ben-Ameur, Walid Ben-Ameur, Alessandro Maddaloni. A cops and robber game and the meeting time of synchronous directed walks. *Networks*, 2024, 10.1002/net.22234 . hal-04607550

**HAL Id: hal-04607550**

**<https://hal.science/hal-04607550>**

Submitted on 10 Jun 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A cops and robber game and the meeting time of synchronous directed walks

Walid Ben-Ameur<sup>1</sup> and Alessandro Maddaloni<sup>\*1</sup>

<sup>1</sup>SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris,  
Palaiseau, France

June 10, 2024

## Abstract

In a previous work, the authors showed that the maximum number of infinitely long synchronous directed walks that never meet is equal to the dimension  $d$  of the no-meet matroid, namely the largest order of a collection of vertex-disjoint cycles. Given  $w > d$ , we want to compute the meeting time of  $w$  walks: the first time step  $t_w$  such that, given any set of  $w$  walks, at least two of them must meet no later than  $t_w$ . We precisely prove that the meeting time is at most  $n - w + 2$ , where  $n$  is the number of vertices. A connection is established with a cops and robber game on directed graphs with helicopter cops and an invisible slow robber. The meeting time of  $w$  walks equals the capture time in this game, when at most  $w - 1$  capture attempts are allowed. While this capture time can be computed in polynomial time, we show that it is NP-hard to compute the minimum number of cops  $h$  needed to catch the robber. More insights are also given on the number  $h$  and its relation to pathwidth and other graph parameters. Finally we analyze these game measures on digraph tensor products.

**Keywords:** Capture time, Complexity, Cop number, Cops and robber games, Digraphs, Hunters and rabbit, No-meet matroid, Walks.

## 1 Introduction

Consider a set of infinitely long walks through a digraph. They have the no-meet property if the  $i$ -th vertices appearing in the walks are disjoint for every  $i \geq 1$ . Consider, for example, a set of agents or robots simultaneously walking through a network. Since the walks are simultaneous and one time step is needed to traverse each edge, the walks are called synchronous walks. Assuming that two robots should never be on the same vertex at the same time leads to the no-meet property. It is proved in [4] that the largest possible cardinality of a set of walks with the no-meet

---

\*corresponding author: [alessandro.maddaloni@telecom-sudparis.eu](mailto:alessandro.maddaloni@telecom-sudparis.eu)

property is equal to the dimension  $d$  of the so-called no-meet matroid. It is also shown there that  $d$  equals the maximum number of vertices belonging to a collection of vertex disjoint cycles. We are going to prove that the number  $d$  is nothing but the minimum number of capture attempts in a cops and robber game. Each time a cop patrols a vertex for a time step costs one token. The number of capture attempts can then be seen as the number of tokens spent to capture the robber.

In this paper, we are interested in the first time walks meet. More precisely, given a number  $w > d$ , we aim to study the first time step  $t_w$  such that, given any set of  $w$  walks, at least two of them must meet no later than  $t_w$ . We show that this is exactly the capture time of the game when at most  $c = w - 1$  capture attempts are allowed. The main result of the paper is that the capture time with  $c \geq d$  attempts is at most  $n - c + 1$ , where  $n$  is the number of vertices. In other words, given any set of  $w > d$  directed walks, at least two of them meet at time at most  $n - w + 2$ . Notice the difference with classical work (see, e.g., [12, 25]) related to the meeting time of random walks modeled as the hitting time of some Markov chain. In our paper, we focus on the worst case in terms of meeting time, and the moves are not random.

We might also see a link with multi-agent path finding (MAPF) problems reviewed in [29]. Each agent has a source location and a target location that he wants to reach. At each time step, an agent can either move to a neighbor (in an undirected graph) or stay in his position (which is equivalent to assume that there is a loop for each vertex). The most basic version assumes that two agents cannot be simultaneously on the same vertex, and the goal is to minimize the time needed to reach all targets. This problem and some of its variants are known to be NP-hard (see, e.g., [30, 32]). Observe that our result about meeting time has some implications for MAPF problems. Assume that the agent's moves should follow the edges of a directed graph, and that an agent can wait in a vertex only if there is a loop there. Assume the number of agents  $w$  is strictly larger than the dimension  $d$  of the associated no-meet matroid. We can certify that if the MAPF problem has a solution, then the number of time steps to simultaneously reach all targets (or the makespan) is at most  $n - w$ . This observation is useful only if the dimension  $d$  is strictly less than  $n$  which implies that there are some vertices without loops.

Other applications related to finite state machines are mentioned in [4] where the concept of  $h$ -robust distinguishing sequences is introduced, and the meeting time  $t_w$  is shown to be an upper bound of the robustness threshold  $h$ .

Cops and robber games have been introduced by Quillot in [26] and intensively studied over the last four decades in their many variants. For a nice introduction, mainly on undirected graphs, see, e.g., [8].

Seymour and Thomas [27] introduced a variant with helicopter cops, namely the cops need not follow the graph edges. In their version the robber must follow the graph edges, but has unlimited speed so in one time step he (or she) can visit any vertex on a cop free path starting from his position. When the robber is visible to the cops, the cop number of an undirected graph, being the minimum number of cops needed to guarantee the capture, equals the treewidth of the graph plus one [27]. If the robber is invisible to the cops, the cop number equals the pathwidth of the graph (or vertex separation number) plus one [22, 20]. In both versions the winning strategy for the cops is monotone, namely the robber territory never increases. These

kind of games are also formulated as graph searching problems, for a bibliography see [15].

Much work has been done mostly on undirected graphs, but different game versions (and their relation to graph parameters) are studied on directed graphs too, see for example [3, 5, 19, 24, 31]. Barát [3] studied the invisible version of the game on directed graphs, and he showed that the cop number is either the directed pathwidth  $pw$  of the digraph or  $pw + 1$ . The author also conjectured that the game is monotone.

An analogue of the game we analyze has been studied on undirected graphs under the name hunters and rabbit (or prince and princess) [1, 11], where helicopter cops play against an invisible robber moving at speed one. In fact, our game is a directed version of hunters and rabbit where loops are admitted.

In hunters and rabbit, the cop (or hunter) number can be determined for special graphs such as paths, cycles, grids or hypercubes [1, 7]. An upper bound on the cop number for trees is given in [18]. In [13] the authors show that the cop number is upper bounded by  $pw + 1$ , but can be arbitrarily smaller. However, when a monotone capture is required,  $pw$  cops are needed. The complexity of computing the cop number is not known, however the problem is FPT when parametrized by the size of a minimum vertex cover [?].

Capture time of a strategy using a fixed number of cops has been introduced by Bonato et al. [9]. In most of the literature, the concept is related to the (original) variant of the game where cops have no helicopter, so they both follow the edges with speed one and the robber is visible. In [9] they showed that for graphs with cop number one, the capture time is at most  $n - 4$ , where  $n$  is the number of vertices of the graph. Later, Brandt et al. [10] proved that the bound  $O(n^{k+1})$  is tight for graphs whose cop number is  $k \geq 2$ . The latter result extends to directed graphs, while Kinnersley showed [21] that there are directed graphs with cop number one whose capture time is  $\Omega(n^2)$ .

The concept of capture attempts has been studied for node searching, namely the version with helicopter cops and invisible robber with unlimited speed: in [14] Fomin and Golovach showed that, for this variant, the number of capture attempts equals the smallest length of an interval graph containing the original graph, they also proved that this is equivalent to the vertex separation sum and that calculating the minimum number of capture attempts in node searching is NP-hard.

The game we study is played on a directed graph  $D = (V, A)$  where loops are allowed and each vertex has at least one outgoing arc, we assume that the cops need not follow the digraph arcs, while the robber must move to a vertex adjacent from his current position (differently from vertex search games). We also assume that the robber is invisible to cops, while he can see the cops' positions. Note that in most cops and robber games, the robber can also stay on his current vertex (this can be obtained by adding loops on all the vertices in our version).

We show that the cop number in our version of the game is upper bounded by the directed pathwidth  $pw$  of the digraph plus one, but it can be arbitrarily smaller. Moreover we show that the game is not monotone, but when a monotone capture is required, the cop number is  $pw$  or  $pw + 1$ . These same results have been proved for the hunters and rabbit game [13]. We also prove that the cop number is at least the

minimum outdegree of any induced subgraph of  $D$ . Finding an induced subgraph whose minimum outdegree is maximized can be done in polynomial time. The cop number is also upper bounded by the size of a minimum feedback vertex set.

In many variants of cops and robber games (see e.g. [16, 23]) computing the cop number is NP-hard. We will show that computing the cop number for our game is NP-hard too. It remains NP-hard even if one requires that the capture occurs before some time step. On the other hand, our results imply a polynomial algorithm to compute the minimum number of capture attempts.

The paper is organized as follows. In Section 2 we give the main definitions related to the game. Some connections between the proposed game and no-meet matroids are presented in Section 3. The main result related to capture time versus capture attempts is shown in Section 4. Section 5 is dedicated to the cop number. Finally, digraph tensor products are considered in Section 6.

## 2 Game definitions

Consider a directed graph  $D = (V, A)$  where loops are allowed and  $\delta^+(D)$ , the minimum outdegree of  $D$ , is at least one. We define a game on  $D$  which we will refer to as HCISR (helicopter cops invisible slow robber). A set of cops wants to capture a robber moving on  $D$  with the following rules:

1. At step 1 the cops pick  $W_1 \subseteq V$ , then the robber picks a vertex  $r_1 \in V$ .
2. At step  $i + 1$  (for any  $i \geq 1$ ) the cops pick  $W_{i+1} \subseteq V$  and the robber picks  $r_{i+1} \in N^+(r_i)$ .

In other words, at each step the cops can pick any vertex, while the robber must pick a vertex adjacent from his current position (we use  $N^+(v)$  to denote the set of vertices  $y$  such that  $(v, y)$  is an arc of  $D$ . If  $S$  is a set,  $N^+(S) := \{y \in V(D) \mid \exists x \in S, \text{ with } (x, y) \in A(D)\}$ ).

We say that the cops capture the robber if  $r_i \in W_i$  for some  $i \geq 1$ . The capture is at time  $t$ , if  $t$  is the minimum index such that  $r_t \in W_t$ . The cops do not know the vertex picked by the robber; their strategy is defined by the sequence  $(W_i)_{i \geq 1}$ , while the robber strategy is defined by  $(r_i)_{i \geq 1}$ . A cop strategy is winning if, by playing that strategy, they can capture the robber regardless of his strategy. We say that a cop strategy uses  $h$  cops if  $h$  is the maximum of  $|W_i|$  over all the indices. The capture time of a cop winning strategy is the maximum time step  $T$ , over all possible robber's strategies, such that the cops capture the robber at time  $T$ . A cop winning strategy uses  $c$  attempts if  $c = \sum_{i=1}^T |W_i|$ , where  $T$  is the capture time of the cops' strategy.

Let  $R_i$  be the set of vertices where the robber can be at step  $i$ . Observe that if the robber was not yet captured at time step  $i - 1$ , then we have  $R_i = N^+(R_{i-1}) \setminus W_i$ . We say that a capture is monotone if  $R_j \subseteq R_i$  for every  $j \geq i$ .

The cop number of  $D$ , denoted by  $cn(D)$ , is the minimum  $h$  such that there exists a winning strategy using  $h$  cops. The capture attempts number of  $D$ , denoted by  $ca(D)$  is the minimum  $c$  such that there exists a cop winning strategy using  $c$  attempts. The capture time using  $c \geq ca(D)$  attempts, denoted by  $ct(D, c)$  is the minimum capture time of a cop winning strategy using  $c$  attempts. When fixing a time

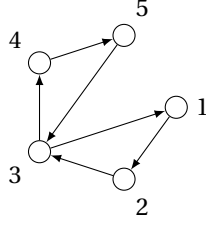


Figure 1: Example of a digraph  $D$  with two intersecting 3-cycles.

limit  $l$ , we can define  $cn(D, l)$  (resp.  $ca(D, l)$ ) as the minimum number of cops (resp. capture attempts) needed for the cops to capture no later than time step  $l$ . Observe that  $ca(D, l) \leq c$  is equivalent to  $ct(D, c) \leq l$ . As an example consider a directed cycle  $Y$  on  $n$  vertices: we have  $cn(Y) = 1$ ,  $ca(Y) = n$ ,  $ct(Y, n) = 1$ ,  $cn(Y, l) = \lceil \frac{n}{l} \rceil$ ,  $ca(Y, l) = n$ . Notice that if the cycle contains a loop around each vertex, the cop number raises to 2 while it equals 1 even if there is at least one vertex without loop. For illustration, consider a digraph  $D$  containing a directed cycle on 4 vertices numbered from 1 to 4 and 3 loops (no loop around 4). Then  $cn(D, 1) = 4$ , while  $cn(D, l) = 2$  for  $2 \leq l \leq 7$ , and  $cn(D, 8) = 1$  implying that  $cn(D) = 1$ . A possible winning strategy for  $l = 2$  is to put cops in vertices 2 and 3 at time 1 (so  $W_1 = \{2, 3\}$  and  $R_1 = \{1, 4\}$ ) and then move the cops to vertices 1 and 2 at time 2 to catch the robber. A winning strategy in the last case (i.e., when  $l = 8$ ) is given in the following table (we show the cop position  $W_t$  and the robber territory  $R_t$  for  $1 \leq t \leq 8$ ). The cop is trying to reduce the territory of the robber using the fact that there is no loop around vertex 4. At time 2, the size of the territory decreased from 3 to 2. Then the cop is “pushing” the robber until the territory contains again vertex 4. Using one more time the lack of possibilities at vertex 4, the cop is able to reduce the size of the territory from 2 to 1. He continues pushing the robber to reach again vertex 4, and capture him at vertex 1. While the strategy seems intuitive, we will show later that computing an optimal winning strategy is an NP-hard problem, so one should not always expect a simple winning strategy. Observe that the capture in this case is not monotone. If we impose mono-

$t$	1	2	3	4	5	6	7	8
$W_t$	{3}	{1}	{2}	{3}	{1}	{2}	{3}	{1}
$R_t$	{1, 2, 4}	{2, 3}	{3, 4}	{1, 4}	{2}	{3}	{4}	$\emptyset$

Table 1: Winning strategy for the cops on a cycle with four vertices and loops on all vertices except vertex 4.

tonicity, it is not possible to capture the robber using only one cop. Using 2 cops is however sufficient to monotonously capture the robber no later than time step 2 (the strategy is the one described above where  $W_1 = \{2, 3\}$  and  $W_2 = \{1, 2\}$ ). We also notice that  $ca(D) = 4 = ca(D, l)$  for any  $l \geq 1$  and  $ct(D, c) = 1$  for any  $c \geq 4$ .

To better illustrate the definitions, let us consider another example where  $D$  is a 5-vertex digraph containing two intersecting directed cycles: one going through

vertices 1, 2 and 3 and another containing vertices 3, 4 and 5 (Fig 1). The values of

$l$	1	2	3	4	5
$cn(D, l)$	5	2	1	1	1
$ca(D, l)$	5	4	3	3	3

Table 2: The values of  $cn(D, l)$  and  $ca(D, l)$  for the 5-vertex digraph with two intersecting 3-cycles.

$cn(D, l)$  and  $ca(D, l)$  for different values of  $l$  are given in Table 2. For  $l = 1$ , the number of cops and the number of capture attempts needed to capture the robber are obviously equal to the number of vertices. If the cops are more patient and accept to capture the robber only on time step 2, then only 2 cops are needed. A possible strategy is  $W_1 = \{3, 4\}$  and  $W_2 = \{2, 3\}$ . This leads to 4 capture attempts which is exactly the minimum required number of capture attempts to guarantee the end of the game no later than time step 2. If we can wait until time step 3, then one cop is sufficient to do the job. A possible winning strategy is to keep the cop at vertex 3 for all 3 time steps. This leads to 3 capture attempts which is again equal to  $ca(D, 3)$ . If  $l$  increases further, both  $cn(D, l)$  and  $ca(D, l)$  stay constant. Observe also that we obviously have  $ca(D, l) \leq l \cdot cn(D, l)$  since each cop can make at most one capture attempt at each time step. We have seen in Table 2 that  $ca(D, l)$  and  $l \cdot cn(D, l)$  can be equal for some values of  $l$  and different for others. In terms of capture time for the considered graph, we have  $ct(D, 3) = 3$ ,  $ct(D, 4) = 2$ , and  $ct(D, 5) = 1$  while capture is not guaranteed with less than 3 attempts.

$D = (V, A)$	Directed graph (assuming the minimum outdegree $\delta^+(D) \geq 1$ )
$N(D)$	The no-meet matroid related to $D$
$d(N(D))$	Dimension of $N(D)$ ; equal to the size of the largest cycle subdigraph
$\mathcal{L}(D, l)$	A layered digraph related to $D$ containing exactly $l$ layers
$cn(D, l)$	Minimum number of cops to capture the robber no later than time $l$
$cn(D)$	The cop number: minimum number of cops to capture the robber
$ca(D, l)$	Minimum number of capture attempts to catch the robber no later than time $l$ ; equal to the size of a minimum vertex cut in $\mathcal{L}(D, l)$
$ca(D)$	Minimum number of attempts to catch the robber; equal to $d(N(D))$
$ct(D, c)$	Minimum capture time if $c$ capture attempts are allowed
$W_t$	The set of vertices patrolled by cops at time $t$
$R_t$	Robber territory: possible locations assuming he was not yet caught
$t_w$	Meeting time of $w$ walks ( $w > d(N(D))$ ): first time step such that, given any set of $w$ walks, at least two of them meet no later than $t_w$

Table 3: Notation summary

For the sake of clarity, most of notation is summarized in Table 3.

### 3 From no-meet matroid to capture attempts

Consider a digraph  $D = (V, A)$  where the minimum outdegree  $\delta^+(D)$  is at least 1. Given a positive integer  $l$ , we define the  $l$  layers expansion of  $D$ ,  $\mathcal{L}(D, l)$  as follows. The vertex set of  $\mathcal{L}(D, l)$  includes a source  $s$ , a sink  $t$  and  $l$  layers so that each layer is a copy of  $V$ . We indicate the copy of  $u \in V$  on layer  $j$  as  $u^j$ . We call  $V^j$  the vertices of the layer  $j$  and we have  $V(\mathcal{L}(D, l)) = \{s, t\} \cup \bigcup_{j=1}^l V^j$ . We connect the source  $s$  to all the vertices of  $V^1$  and the sink  $t$  from all the vertices of the last layer  $V^l$ . A copy of each arc of  $A$  is made between two consecutive layers so that

$$A(\mathcal{L}(D, l)) = \left( \bigcup_{v \in V} (s, v^1) \right) \cup \left( \bigcup_{\substack{(u,v) \in A \\ j \in \{1, \dots, l-1\}}} (u^j, v^{j+1}) \right) \cup \left( \bigcup_{v \in V} (v^l, t) \right).$$

By construction, there is a one-to-one mapping between  $(s, t)$ -paths in  $\mathcal{L}(D, l)$  and walks of length  $l-1$  in  $D$ . Moreover,  $w$  internally vertex-disjoint  $(s, t)$ -paths in  $\mathcal{L}(D, l)$  correspond to  $w$  walks of length  $l-1$  in  $D$  so that for  $j = 1, \dots, l$  the  $j^{\text{th}}$  vertices of the walks are all different (no-meet condition). A subset of vertices  $S \subseteq V$  is said to be independent if there exist arbitrarily long walks starting at  $S$  (one walk starting from each vertex) that never meet. It is shown in [4] that  $V$  and the set of independent sets form a matroid  $N(D)$  entitled no-meet matroid. The class of no-meet matroids strictly contains transversal matroids and is strictly contained inside the class of gammoids [4]. The dimension of the no-meet matroid denoted by  $d(N(D))$  is the size of a maximum-size independent set. A maximum-size independent set is called a basis. All bases of a matroid have the same size and every independent set is a subset of at least one basis. Consider, for example, the digraph related to Figure 1. The set  $\{1, 2, 3\}$  is a basis while the set  $\{2, 5\}$  is a dependent set.

Observe that a vertex  $(s, t)$ -cut in  $\mathcal{L}(D, l)$  corresponds to a choice of capture attempts that allows to catch the robber no later than time  $l$  in the game HCSIR played on  $D$ .  $ca(D, l)$  can then be found by computing a minimum-size vertex  $(s, t)$ -cut in  $\mathcal{L}(D, l)$ . This is summarized in the next observation.

**Observation 3.1.**  *$ca(D, l)$  is the minimum size of a vertex  $(s, t)$ -cut in  $\mathcal{L}(D, l)$  and is equal to the maximum number of vertex-disjoint  $(s, t)$ -paths in  $\mathcal{L}(D, l)$ .*

*Proof.* The minimum size of a vertex  $(s, t)$ -cut in  $\mathcal{L}(D, l)$  is equal to the maximum number of vertex-disjoint  $(s, t)$ -paths in  $\mathcal{L}(D, l)$  by direct application of Menger's theorem. Consider now capture attempts corresponding to a minimum-size  $(s, t)$ -cut: if the  $j^{\text{th}}$  copy  $u^j$  of a vertex  $u$  belongs to the vertex-cut, a capture attempt is performed on vertex  $u$  at time  $j$ . Any path in  $\mathcal{L}(D, l)$  should necessarily contain one of the vertices of the  $(s, t)$ -cut. In other words, any walk in  $D$  will be intersected no later than  $l$  implying that the robber is captured. If the number of capture attempts is strictly less than the size of an optimal vertex  $(s, t)$ -cut, then the subset  $S$  of vertices of  $\mathcal{L}(D, l)$  corresponding to capture attempts (if a capture attempt occurs on vertex  $u$  at time  $j$ , then  $u^j$  belongs to  $S$ ) is not a vertex  $(s, t)$ -cut. This implies that  $\mathcal{L}(D, l)$  contains a path not intersected by  $S$ . As a consequence,  $D$  contains a walk that can be followed by the robber to escape (at least until time  $l$ ).  $\square$



As already mentioned, the dimension of the no-meet matroid  $d(N(D))$  is the size of a largest set  $S$  such that there are  $|S|$  infinitely-long vertex-disjoint synchronous walks starting at  $S$ . Therefore, the combination with the previous observation leads to our second observation.

**Observation 3.2.** *Let  $D$  be a digraph with  $\delta^+(D) \geq 1$  and let  $N(D)$  be its related no-meet matroid. Then,  $d(N(D)) = \min_l ca(D, l) = ca(D)$ .*

It is proved in [4] that the dimension of the matroid equals the size of the largest cycle subdigraph (i.e., collection of vertex-disjoint cycles). Notice that this does not mean that each basis is the vertex-set of a largest cycle subdigraph. It only implies that there is at least a maximum cycle subdigraph whose vertex-set is a basis. It can be computed in  $O(n^3)$  using matching in bipartite graphs (see, e.g., Theorem 13.8.1 in [2]). Therefore there exists an  $O(n^3)$  algorithm to compute  $ca(D)$ .

We also know from [4] that the minimum in Observation 3.2 is achieved for some  $l \in O(n^5)$ . In other words, there exists a constant  $C$  such that for any  $D$  and  $l \geq Cn^5$ ,  $ca(D, l) = ca(D) = d(N(D))$ . It is then enough to consider a number of layers  $l \geq Cn^5$  to check the independence of a set. The observation about the number of layers implies that the capture time is at most  $Cn^5$  when  $c \geq d(N(D))$  attempts are allowed. We will prove in next section that the capture time is in fact bounded by  $n + 1 - c$ .

## 4 Capture time vs capture attempts

Given a digraph  $D$ , the function  $ca(D, l)$  is obviously non-increasing in  $l$ . As already mentioned, we know that after some  $\bar{l} \in O(n^5)$  the function  $ca(D, l)$  is constant. Here we show that the above estimate can be lowered and we get a sharp result related to the capture time of the game HCSIR.

Note that for  $c \geq ca(D)$ ,  $ct(D, c) = l$  where  $l$  is the minimum integer such that  $ca(D, l) = c$ . We will show that the capture time  $ct(D, c)$  of the game using  $c \geq ca(D)$  attempts is at most  $n - c + 1$ . As a consequence, we will deduce that given any set of  $w > d(N(D))$  walks, at least two of them meet no later than  $n - w + 2$ .

Consider a maximal collection of vertex-disjoint  $(s, t)$ -paths  $\mathcal{F}$  on  $\mathcal{L}(D, l)$ : for  $j = 1, \dots, l - 1$  it defines a matching between  $V^j$  and  $V^{j+1}$ . Let  $M_{\mathcal{F}}^j \subseteq V^j$  be the matched vertices and  $U_{\mathcal{F}}^j := V^j \setminus M_{\mathcal{F}}^j$  the unmatched vertices. For a collection  $\mathcal{F}$  and  $j = 1, \dots, l - 2$ , let  $I_{\mathcal{F}}^j$  be the set of vertices  $u^j \in U_{\mathcal{F}}^j$  that either have a copy  $u^{j+1} \in U_{\mathcal{F}}^{j+1}$  or such that the vertex  $z^{j+2} \in M_{\mathcal{F}}^{j+2}$  matched with  $u^{j+1}$  has a copy  $z^{j+1} \in U_{\mathcal{F}}^{j+1}$ .

Let us start with the following lemma proved through augmenting-path techniques.

**Lemma 4.1.** *Let  $l \geq 2$ , if  $\mathcal{F}$  is a collection of  $ca(D, l) > d(N(D))$  disjoint  $(s, t)$ -paths on  $\mathcal{L}(D, l)$ , then for every  $1 \leq k \leq l - 1$ , there exists a collection  $\mathcal{G}$  of  $ca(D, l)$  disjoint  $(s, t)$ -paths such that  $U_{\mathcal{F}}^k = U_{\mathcal{G}}^k$  and a path  $P = sv_1^1 \dots v_k^k$ , with  $v_j^j \in U_{\mathcal{G}}^j$  for  $j = 1, \dots, k$  and  $v_k^{k+1} \in M_{\mathcal{F}}^{k+1}$ .*

*Proof.* First of all note that  $ca(D, l) < |V(D)|$ , otherwise  $ca(D, l) = |V(D)|$  and we would have  $d(N(D)) = |V(D)|$ , which is incompatible with  $ca(D, l) > d(N(D))$ . Therefore  $U_{\mathcal{F}}^j \neq \emptyset$  for  $j = 1, \dots, l$ . Note that  $U_{\mathcal{F}}^j$  and  $U_{\mathcal{F}}^{j+1}$  cannot contain the copies of the same vertices (i.e., there exists at least one vertex  $v \in V$  such that  $v^j \in U_{\mathcal{F}}^j$  while  $v^{j+1} \notin U_{\mathcal{F}}^{j+1}$ ), since otherwise  $\mathcal{F}$  would contain a matching between the copies of the same set of vertices, implying that  $D$  contains a cycle subdigraph of size  $ca(D, l)$  and thus  $ca(D, l) \leq d(N(D))$ .

We use an inductive argument on  $k$  to show the property of the Lemma: when  $k = 1$ , there must exist  $v_1^1 \in U_{\mathcal{F}}^1$  such that  $v_1^2 \in M_{\mathcal{F}}^2$ , otherwise  $U_{\mathcal{F}}^1$  and  $U_{\mathcal{F}}^2$  would contain the copies of the same vertices. We can just take  $\mathcal{G} = \mathcal{F}$  and the path  $P = sv_1^1$  is as required.

Now assume the lemma is true for a given  $k \leq l - 2$  and let's show it holds for  $k + 1$ . Among all collections  $\mathcal{X}$  of  $ca(D, l)$  disjoint  $(s, t)$ -paths on  $\mathcal{L}(D, l)$ , such that  $U_{\mathcal{X}}^{k+1} = U_{\mathcal{F}}^{k+1}$  and  $U_{\mathcal{X}}^{k+2} = U_{\mathcal{F}}^{k+2}$  let  $\mathcal{F}'$  be the one that maximizes  $|I_{\mathcal{X}}^k|$ . By induction we have the existence of a collection  $\mathcal{G}'$  and a path  $P$  as in the hypothesis. In particular  $U_{\mathcal{F}'}^k = U_{\mathcal{G}'}^k$ , thus if we join the arcs of  $\mathcal{G}'$  until layer  $k$  with the arcs of  $\mathcal{F}'$  from layer  $k$  onward, we still have a collection  $\mathcal{G}$  of  $ca(D, l)$  disjoint  $(s, t)$ -paths that maximizes  $|I_{\mathcal{X}}^k|$ . Moreover  $U_{\mathcal{G}}^{k+1} = U_{\mathcal{F}'}^{k+1} = U_{\mathcal{F}}^{k+1}$ . Let  $v_k^k$  be the last vertex of the path  $P$ , we have that  $v_k^{k+1} \in M_{\mathcal{G}}^{k+1}$ . This means that  $\mathcal{G}$  contains an arc from  $v_k^{k+1}$  to some  $y_1^{k+2} \in M_{\mathcal{G}}^{k+2}$  and  $\mathcal{L}(D, l) \setminus \mathcal{G}$  has the copy of this arc between  $v_k^k$  and a  $y_1^{k+1}$ : call  $a_1$  this arc. We will show that  $y_1^{k+1} \in U_{\mathcal{G}}^{k+1}$  and thus  $P \cup a_1$  is a path as required since  $y_1^{k+2} \in M_{\mathcal{G}}^{k+2}$  (see Figure 2 for illustration).

Assume by contradiction that  $y_1^{k+1} \in M_{\mathcal{G}}^{k+1}$ , then  $\mathcal{G}$  contains an arc from a vertex  $x_1^k$  to  $y_1^{k+1}$ : call  $b_1$  this arc. If  $x_1^{k+1} \in M_{\mathcal{G}}^{k+1}$ , then  $\mathcal{G}$  contains an arc from  $x_1^{k+1}$  to some  $y_2^{k+2}$  and  $\mathcal{L}(D, l) \setminus \mathcal{G}$  contains an arc from  $x_1^k$  to  $y_2^{k+1}$ : call  $a_2$  this arc. Note that, given that  $y_2^{k+2}$  is matched from  $x_1^{k+1} \neq v_k^{k+1}$  (since  $x_1^k \neq v_k^k$ ), then  $y_2^{k+1} \neq y_1^{k+1}$ . Similarly, if  $y_2^{k+1} \in M_{\mathcal{G}}^{k+1}$ , then  $\mathcal{G}$  contains an arc from a vertex  $x_2^k$  to  $y_2^{k+1}$ : call  $b_2$  this arc and note that  $x_2^k \neq x_1^k$ . As long as  $y_j^{k+1}, x_j^{k+1} \in M_{\mathcal{G}}^{k+1}$ , we can define  $a_j, b_j, x_j^k, y_j^{k+1}$  as above, with the  $x_j^k$  being all distinct and the  $y_j^{k+1}$  being all distinct. Now, since  $M_{\mathcal{G}}^k$  does not contain the same copy of vertices as  $M_{\mathcal{G}}^{k+1}$ , there must exist  $h \geq 1$  such that either  $x_h^{k+1} \in U_{\mathcal{G}}^{k+1}$  or  $y_{h+1}^{k+1} \in U_{\mathcal{G}}^{k+1}$ . Figure 2 illustrates the situation where  $h = 2$  and  $x_h^k \in U_{\mathcal{G}}^{k+1}$ .

Let  $Q$  be the path of  $\mathcal{G}$  from  $s$  to  $x_h^k$ . Define  $\hat{\mathcal{F}}$  as the collection obtained from  $\mathcal{G}$  by removing the arcs of  $Q$  and the arcs  $\{b_1, \dots, b_h\}$  and adding the arcs of the path  $P$  and the arcs  $\{a_1, \dots, a_h\}$ . It is easy to see that  $\hat{\mathcal{F}}$  is still a collection of  $ca(D, l)$  disjoint  $(s, t)$ -paths with  $U_{\hat{\mathcal{F}}}^{k+1} = U_{\mathcal{G}}^{k+1} = U_{\mathcal{F}}^{k+1}$  and  $U_{\hat{\mathcal{F}}}^{k+2} = U_{\mathcal{G}}^{k+2} = U_{\mathcal{F}}^{k+2}$ . Note that  $U_{\hat{\mathcal{F}}}^k = (U_{\mathcal{G}}^k \setminus \{v_k^k\}) \cup \{x_h^k\}$  and we have  $v_k^{k+1} \notin U_{\mathcal{G}}^{k+1}$ ,  $y_1^{k+1} \notin U_{\mathcal{G}}^{k+1}$ , so  $v_k^k \notin I_{\mathcal{G}}^k$ . On the other hand either  $x_h^{k+1} \in U_{\hat{\mathcal{F}}}^{k+1}$  or  $y_{h+1}^{k+1} \in U_{\hat{\mathcal{F}}}^{k+1}$ , namely  $x_h^k \in I_{\hat{\mathcal{F}}}^k$ . It means that  $|I_{\hat{\mathcal{F}}}^k| = |I_{\mathcal{G}}^k| + 1 = |I_{\mathcal{F}'}^k| + 1$ , which contradicts the maximality of  $\mathcal{F}'$ .  $\square$

**Corollary 4.2.** *Let  $l \geq 2$ , if  $ca(D, l) > d(N(D))$ , then  $ca(D, l - 1) \geq ca(D, l) + 1$ .*

*Proof.* By Lemma 4.1 applied to  $k = l - 1$  we have a collection  $\mathcal{G}$  and a path  $P =$

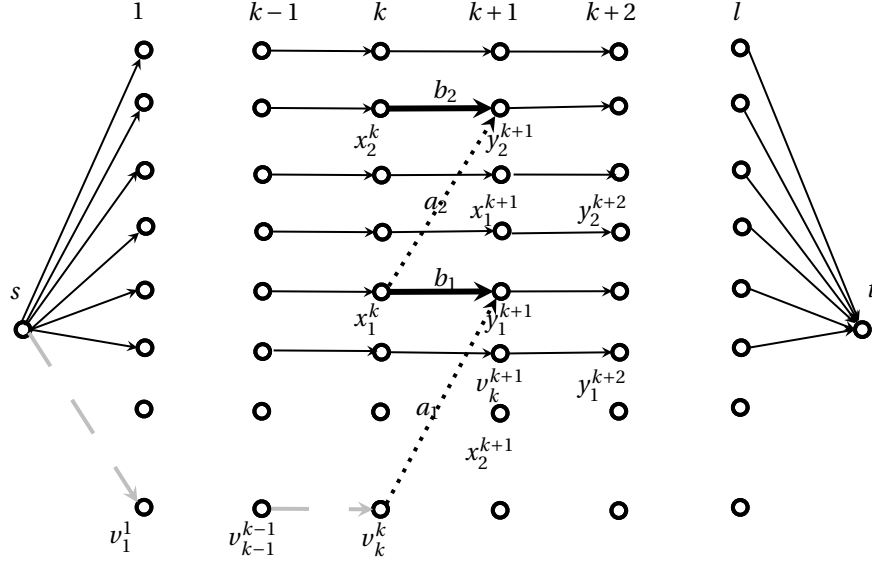


Figure 2: Illustration of the proof of Lemma 4.1.

$sv_1^1 \dots v_{l-1}^{l-1}$  on  $\mathcal{L}(D, l)$  such that  $v_i^i \in U_{\mathcal{G}}^i$  for  $i = 1, \dots, l-1$ . Consider the collection  $\mathcal{G}'$  on  $\mathcal{L}(D, l-1)$  that contains the same arcs as  $\mathcal{G}$  on the first  $l-1$  layers, plus the arcs of the path  $P$  and all arcs from  $M_{\mathcal{G}}^{l-1} \cup \{v_{l-1}^{l-1}\}$  to  $t$ . This is a collection of  $ca(D, l) + 1$  disjoint  $(s, t)$ -paths on  $\mathcal{L}(D, l-1)$ .  $\square$

Corollary 4.2 implies that  $ca(D, l)$  starts with  $ca(D, 1) = n$  and is strictly decreasing with increasing  $l$ , until its value reaches  $d(N(D))$  and remains constant. This implies the following upper bound for the number of capture attempts.

**Theorem 4.3.** *Let  $D$  be a digraph on  $n$  vertices with  $\delta^+(D) \geq 1$ . Let  $c$  be such that  $d(N(D)) \leq c \leq n$ . The number of attempts needed to capture the robber no later than time  $n - c + 1$ , satisfies  $ca(D, n - c + 1) \leq c$ .*

*Proof.* Suppose by contradiction  $ca(D, n - c + 1) > c$ , then by applying Corollary 4.2  $n - c$  times we have

$$ca(D, 1) \geq ca(D, 2) + 1 \geq ca(D, 3) + 2 \geq \dots \geq ca(D, n - c + 1) + n - c > n.$$

This is impossible since  $ca(D, 1) = n$ .  $\square$

Observe that Corollary 4.2 implies that  $ct(D, c)$  defined for  $d(N(D)) \leq c \leq n$  is a non-increasing piecewise constant function reaching all integer values between

$ct(D, d(N(D)))$  and  $1 = ct(D, n)$ . In other words, by allowing one more attempt, we cannot reduce the capture time by more than one time step.

**Theorem 4.4.** *The capture time using  $c$  attempts satisfies  $ct(D, c) \leq n - c + 1$ , and  $ct(D, c) - 1 \leq ct(D, c + 1) \leq ct(D, c)$ .*

*Proof.* The inequality  $ct(D, c) \leq n - c + 1$  is a direct consequence of  $ca(D, n - c + 1) \leq c$ . Moreover,  $ct(D, c + 1)$  is obviously less than or equal to  $ct(D, c)$ . Assume that  $ct(D, c + 1) = l < ct(D, c)$ . Then,  $ca(D, l) \leq c + 1$ . Combined with  $ct(D, c) > l$ , we get that  $ca(D, l) = c + 1 > d(N(D))$ . From Corollary 4.2, we can infer that  $ca(D, l + 1) \leq ca(D, l) - 1 = c$ . Therefore,  $ct(D, c) \leq l + 1 = ct(D, c + 1) + 1$  ending the proof.  $\square$

The above bounds are sharp as we can see by considering the family of digraphs obtained by taking a cycle on  $r$  vertices, and a path oriented towards the cycle and containing  $n - r + 1$  vertices disjoint from the cycle except for the terminal vertex of the path. Let  $v_1, v_2, \dots, v_{n-r+1}$  be the vertices of the path while  $v_{n-r+1}, \dots, v_n$  belong to the cycle. The digraph  $D$  contains an edge  $v_i v_{i+1}$  for  $i < n$  in addition to the edge  $v_n v_{n-r+1}$ . The dimension is  $r$ , while  $ca(D, l) = n - l + 1$  for  $1 \leq l \leq n - r + 1$ . This can be checked as follows. Using Theorem 4.3 we deduce that  $ca(D, l) \leq n - l + 1$  for  $1 \leq l \leq n - r + 1$ . To prove equality, we only have to exhibit  $n - l + 1$  synchronous disjoint walks of length  $l - 1$ . They are provided below where each row shows the  $l$  vertices belonging to each walk.

$v_1$	$v_2$	$\cdots$	$v_{l-1}$	$v_l$
$v_2$	$v_3$	$\cdots$	$v_l$	$v_{l+1}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
$v_{n-l+1}$	$v_{n-l+2}$	$\cdots$	$v_{n-1}$	$v_n$

Since these walks correspond to vertex-disjoint paths in  $\mathcal{L}(D, l)$ , we need at least  $n - l + 1$  capture attempts to intersect each one of these walks implying that  $ca(D, l) \geq n - l + 1$ . If  $l \geq n - r + 1$ , then  $ca(D, l) = r$  since  $ca(D, l)$  is non-increasing with increasing  $l$  and we just proved that  $ca(D, l = n - r + 1) = r$ . The capture time values immediately follow from what we know about  $ca(D, l)$ :  $ct(D, c) = n - c + 1$  for  $r \leq c \leq n$ .

As a consequence of the above theorem, we get the main result of the paper, namely an upper bound for the meeting time. Remember that  $t_w$  is the first time step such that, given any set of  $w$  walks, at least two of them must meet no later than  $t_w$ .

**Theorem 4.5.** *Let  $d(N(D)) < w \leq n$  and  $c = w - 1$ , then  $t_w = ct(D, c)$  and  $t_w \leq n - w + 2$ .*

*Proof.* From the definition of  $ct(D, c)$ , a robber can escape until time  $t = ct(D, c) - 1$  when using  $c \geq d(N(D))$  capture attempts. This means the layered graph  $\mathcal{L}(D, t)$  contains  $w = c + 1$  disjoint paths, hence there exist  $w = c + 1$  walks in  $D$  that do not meet until time  $t$ , implying  $t_w \geq ct(D, c)$ . On the other hand, suppose that  $t_w > ct(D, c)$ , then we can find  $w$  paths that are disjoint in  $\mathcal{L}(D, l = ct(D, c))$  making it impossible to capture the robber with  $c$  attempts in time at most  $ct(D, c)$ . In other words, we have  $t_w = ct(D, c)$ . Using Theorem 4.4, we deduce that  $t_w \leq n - w + 2$ .  $\square$

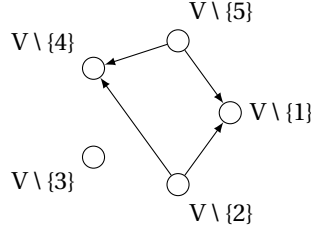


Figure 3: The digraph  $D_4$  obtained from the digraph  $D$  of Fig 1.

Let us state Theorem 4.5 in a different way. Consider a number  $d(N(D)) < w \leq n$  and build the digraph  $D_w$  whose vertex set corresponds to subsets of  $V(D)$  of size exactly  $w$ . An arc from  $\{v_{i_1}, \dots, v_{i_w}\}$  to  $\{v_{j_1}, \dots, v_{j_w}\}$  exists in  $D_w$  only if  $\{v_{j_1}, \dots, v_{j_w}\}$  can be obtained from  $\{v_{i_1}, \dots, v_{i_w}\}$  by replacing each vertex by exactly one of its successors in  $D$  (in other words,  $w$  walkers located at  $\{v_{i_1}, \dots, v_{i_w}\}$  can move to  $\{v_{j_1}, \dots, v_{j_w}\}$  in exactly one step without meeting). Theorem 4.5 implies that each path of  $D_w$  contains at most  $n - w$  arcs (and is then cycle-free). As an example, consider again the digraph  $D$  of Figure 1 where  $d(N(D)) = 3$  and take  $w = 4$ . The digraph  $D_4$  is shown in Figure 3. We can see that  $D_4$  does not contain paths of length 2.

Another consequence of Corollary 4.2 is that  $ca(D, n) = d(N(D))$ , therefore to compute  $d(N(D))$  and  $ca(D)$  it is enough to compute the maximum number of disjoint  $(s, t)$ -paths on  $\mathcal{L}(D, n)$ . However, we already have an algorithm to compute the two measures based on matching in bipartite graphs [4].

## 5 Cop number

In this section we give some considerations on the cop number. Let us start by recalling the definition of directed path decomposition and pathwidth.

A directed path decomposition of  $D$  is a collection of subsets (called bags)  $B_1, \dots, B_k$  such that

1.  $\bigcup_{i=1}^k B_i = V(D)$
2. If  $i < j < k$ , then  $B_i \cap B_k \subseteq B_j$
3. For every arc  $uv$  there exist  $i \leq j$  such that  $u \in B_i$  and  $v \in B_j$ .

The width of a path decomposition is  $\max_i |B_i| - 1$ . The directed pathwidth of  $D$ , denoted by  $pw(D)$ , is the minimum possible width of a directed path decomposition of  $D$ .

An equivalent measure is the directed vertex separation. Let  $\sigma = (v_1, \dots, v_n)$  be an ordering of  $V(D)$ . The separation of  $\sigma$  is  $\max_i |N^+(\{v_{i+1}, \dots, v_n\}) \cap \{v_1, \dots, v_i\}|$ . The directed vertex separation of  $D$  is the minimum separation among all possible orderings. This number equals the directed pathwidth of  $D$  [31].

Similarly to what is shown in [3] for their version of the game, the number of cops

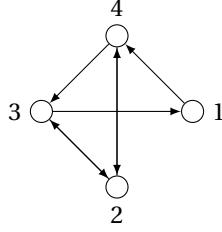


Figure 4: Example of a digraph  $D$  with  $cn(D) < pw(D)$ .

for our game is upper bounded by the pathwidth plus one. This follows immediately from the fact that our game is not more difficult for the cops than the same game with an arbitrarily fast robber, but we give a direct proof to better illustrate the relation with the pathwidth.

**Theorem 5.1.** *The cop number  $cn(D)$  of the game HCISR is at most  $1 + pw(D)$ .*

*Proof.* Let  $B_1, \dots, B_k$  be a directed path decomposition of  $D$ . We prove that the sequence  $W_1 = B_1, \dots, W_k = B_k$  is a winning strategy for the cops.

To prove this, we will inductively show that for  $t \leq k$ , if  $r_t \in W_a$ , with  $a \leq t$ , then the robber is captured.

The statement is clearly true at the initial time step (the cops are in  $W_1$ ). For the inductive step assume  $r_{t+1} \in W_b$  with  $b \leq t+1 \leq k$ . If  $b = t+1$  the robber is captured, so assume  $b \leq t$ . At step  $t$ , the robber is at  $r_t \in W_a$ , for some  $a$ : if  $a \leq t$  the robber is captured by induction, otherwise assume  $a \geq t+1$ . By property 3 of a directed path decomposition, since  $r_t r_{t+1}$  is an arc of  $D$ , there exist  $i \leq j$  such that  $r_t \in W_i$  and  $r_{t+1} \in W_j$ . Note that  $r_t \in W_a \cap W_i$ , so if  $i \leq t$ , then by property 2  $r_t \in W_t$ , so the robber is captured. If  $i \geq t+1$ , since  $r_{t+1} \in W_b \cap W_j$ , with  $b \leq t+1 \leq j$ , we have that  $r_{t+1} \in W_{t+1}$  and again the robber is captured.  $\square$

Note that this bound is tight: it is easy to see that directed cycles with loops on all vertices have directed pathwidth 1 and two cops are needed to capture the robber. On the other hand, for a loopless directed cycle  $C$ ,  $pw(C) = cn(C) = 1$ . Furthermore the digraph in Figure 4 has cop number 1 (strategy 3,2,3,4,2,4) and it is not hard to see (by checking all possible vertex orderings) that the directed vertex separation and hence the directed pathwidth is 2. We will construct in Section 6 digraphs with cop number arbitrarily less than the pathwidth.

Observe that HCISR is not monotone as there is no monotone capture with one cop from the digraph of Figure 4. We can actually prove the following

**Theorem 5.2.** *A monotone capture for HCISR on a digraph  $D$  requires at least  $pw(D)$  cops, and at most  $1 + pw(D)$  cops.*

*Proof.* Let us start with the upper bound. We are going to show the monotonicity of the winning strategy introduced in the proof of Theorem 5.1. We already proved that if the robber was not captured before time  $t$ , then the robber territory satisfies

$R_t \subseteq V(D) \setminus \cup_{i=1}^{i=t} W_i$ . Let us prove by induction that  $R_{t+1} \subseteq R_t$ . We obviously have  $R_1 = V(D) \setminus W_1$  and  $R_2 \subseteq V(D) \setminus (W_1 \cup W_2)$  leading to  $R_2 \subseteq R_1$ . Let us assume that  $R_t \subseteq R_{t-1}$ . Since both sets  $R_t$  and  $R_{t+1}$  are subsets of  $V(D) \setminus \cup_{i=1}^{i=t} W_i$ , we only have to show that  $(V(D) \setminus \cup_{i=1}^{i=t} W_i) \setminus R_t$  does not intersect  $R_{t+1}$ . Let  $v \in (V(D) \setminus \cup_{i=1}^{i=t} W_i) \setminus R_t$ . Since  $v \notin R_t \cup W_t$ ,  $v \notin N^+(R_{t-1})$ . This implies (by induction hypothesis) that  $v \notin N^+(R_t)$ . Hence  $v \notin R_{t+1}$ . As a consequence,  $R_{t+1} \subseteq R_t$ , proving the induction. The strategy is hence monotone and the upper bound is valid.

To prove the lower bound, let  $W_1, \dots, W_k$  be the monotone winning strategy for the cops and let  $V(D) \setminus W_1 = R_1 \supseteq \dots \supseteq R_k = \emptyset$  be the corresponding sets of possible robber positions. Let  $\bar{R}_i = V(D) \setminus R_i$  be their complements.

Suppose, by contradiction, that the number of cops  $h$  is strictly less than  $pw(D)$  and observe that  $k > 1$ . We can order the vertices of  $D$  so that the vertices of  $\bar{R}_i$  appear before those of  $\bar{R}_{i+1} \setminus \bar{R}_i$  for every  $i = 1, \dots, k-1$ . Let  $\sigma = (v_1, \dots, v_n)$  be such an ordering (where  $n := |V(D)|$ ). By equality of pathwidth and vertex separation, there exists  $l$ , with  $h < l < n$ , such that  $|N^+(\{v_{l+1}, \dots, v_n\}) \cap \{v_1, \dots, v_l\}| \geq pw(D)$ . Observe that  $\{v_1, \dots, v_l\} \subset \bar{R}_1 = W_1$  would imply that  $pw(D) \leq |N^+(\{v_{l+1}, \dots, v_n\}) \cap \{v_1, \dots, v_l\}| \leq l \leq |W_1| \leq h$ , contradicting the assumption  $h < pw(D)$ . By construction of  $\sigma$ , we can then assume that there exists a time step  $t < k$  such that  $\bar{R}_t \subseteq \{v_1, \dots, v_l\}$  and  $\bar{R}_{t+1} \supseteq \{v_1, \dots, v_l\}$ . This means that  $R_t \supseteq \{v_{l+1}, \dots, v_n\}$ , therefore  $|N^+(R_t) \cap \{v_1, \dots, v_l\}| \geq pw(D)$ . But there are less than  $pw(D)$  cops, thus  $R_{t+1} \cap \{v_1, \dots, v_l\} \neq \emptyset$ , which is a contradiction.  $\square$

It is a known result, see e.g. [6], that it is NP-hard to approximate the pathwidth of a graph up to a constant, this holds also for directed pathwidth, given that the pathwidth of a graph  $G$  equals the directed pathwidth of the bioorientation of  $G$ . Therefore we have the following result.

**Corollary 5.3.** *It is NP-hard to compute the minimum number of cops required for a monotone capture in the game HCISR on a given digraph  $D$ .*

Next theorem introduces further bounds for the cop number. We use  $D[S]$  to denote the subdigraph induced by a set of vertices  $S \subset V$ . Let  $FV(D)$  denote the size of a minimum feedback vertex set (a vertex set intersecting all directed cycles).

**Theorem 5.4.**  $\max_{S \subset V} \delta^+(D[S]) \leq cn(D) \leq FV(D)$ .

*Proof.* For the cops to win, there must exist a time step  $t$  such that  $R_t = \emptyset$ . Now if a strategy uses  $h < \delta^+(D)$  cops, we have  $R_1 \neq \emptyset$  and for every  $i$ ,  $|N^+(R_i)| > h$ , implying  $R_{i+1} \neq \emptyset$ . This clearly implies that the cop number is at least the minimum outdegree of the graph. This also holds for any induced subgraph  $D[S]$  leading to the lower bound (since the cop number of  $D$  cannot be smaller than the cop number of any induced subgraph). The validity of the upper bound is also obvious: by putting a cop on each vertex of an optimal feedback vertex set, the robber is necessarily captured since all directed cycles are intersected.  $\square$

While computing the size of an optimal feedback vertex set is an NP-hard problem, the lower bound of Theorem 5.4 is in fact easy to compute. Once we have a procedure to check whether  $D$  contains an induced digraph  $D[S]$  such that  $\delta^+(D[S]) \geq k$

for some  $k$ , one can use it to compute the lower bound through binary search. Such a procedure is straightforward: continuously update the digraph by deleting each vertex whose outdegree is  $< k$  until there are no possible updates. If the remaining digraph is not empty, we have a subdigraph whose minimum outdegree is  $\geq k$ , otherwise no such subdigraph exists. This approach to maximize the minimum outdegree was already proposed in literature, for example, in [28] for community detection.

Simple bounds for  $cn(D, l)$  are recalled in the following Proposition.

**Proposition 5.5.**  $\lceil \frac{ca(D, l)}{l} \rceil \leq cn(D, l) \leq ca(D, l)$ .

*Proof.* Each cop among the  $cn(D, l)$  cops can perform at most one capture attempt at each time step leading to  $ca(D, l) \leq l \cdot cn(D, l)$ . Moreover, we need at most one cop to carry out each capture attempt, implying that  $cn(D, l) \leq ca(D, l)$ .  $\square$

We know from the previous section that  $ca(D, l)$  can be computed in polynomial time, implying that we have an  $l$ -approximation of  $cn(D, l)$ . However, computing  $cn(D, l)$  is NP-hard for every fixed  $l$ . We will show this using reductions from the following partition problem: given positive integers  $(a_1, \dots, a_n)$  decide whether there exists  $X \subset \{1, \dots, n\}$  such that  $\sum_{i \in X} a_i = \sum_{i \notin X} a_i$ . In other words, one wants to partition the given integers into two subsets having the same sum. This is a well-known NP-complete problem [17].

**Theorem 5.6.** *For every  $l \geq 2$  it is NP-hard to compute the number  $cn(D, l)$  of a given digraph  $D$ .*

*Proof.* Let  $(a_1, \dots, a_n)$  be an instance of the partition problem and let  $m := \frac{\sum_i a_i}{2}$ . We can clearly assume  $m \in \mathbb{N}$  (otherwise we have a NO instance). We construct a digraph  $D$  such that  $cn(D, l) = m$  if and only if  $(a_1, \dots, a_n)$  is a yes instance of partition. The digraph  $D$  contains, for each  $i = 1, \dots, n$ , the biorientation of the complete graph on  $a_i$  vertices. Let  $V_1, \dots, V_n$  be their vertex sets. The digraph  $D$  also contains  $l - 2$  disjoint copies of the biorientation of the complete graph on  $m$  vertices. Let  $Z_1, \dots, Z_{l-2}$  be their vertex sets. All vertices of  $D$  have also a loop. An example of this construction is shown in Figure 5.

Suppose there exists a solution  $X$  to the partition problem: a winning strategy using  $m$  cops is  $Z_1, \dots, Z_{l-2}, \cup_{i \in X} V_i, \cup_{i \notin X} V_i$ . Note that this strategy uses exactly  $l$  time steps and  $m$  cops. Now, using Theorem 5.4, we get  $cn(D, l) \geq cn(D) \geq \delta^+(D) = m$ , therefore  $cn(D, l) = m$ .

Vice versa, suppose  $cn(D, l) = m$ . Let  $t(i)$  be the last time step  $t$  such that  $R_t \cap Z_i \neq \emptyset$ . Then we necessarily have  $W_{1+t(i)} = Z_i$  since  $N^+(Z_i) = Z_i$  and  $|Z_i| = m$ . So the cops need  $l - 2$  time steps to eliminate the  $Z_i$ 's from the robber's territory and can only use 2 time steps to eliminate  $V_1, \dots, V_n$ . Therefore in one of the two time steps they must cover  $\cup_{i \in X} V_i$ , for some  $X \subset \{1, \dots, n\}$  and since they only use  $m$  cops it means that  $\sum_{i \in X} a_i = \sum_{i \notin X} a_i = m$  and thus  $X$  is a solution of partition.  $\square$

With a similar construction we can reduce partition to the problem of determining whether  $cn(D) = m$  on a particular digraph  $D$ .



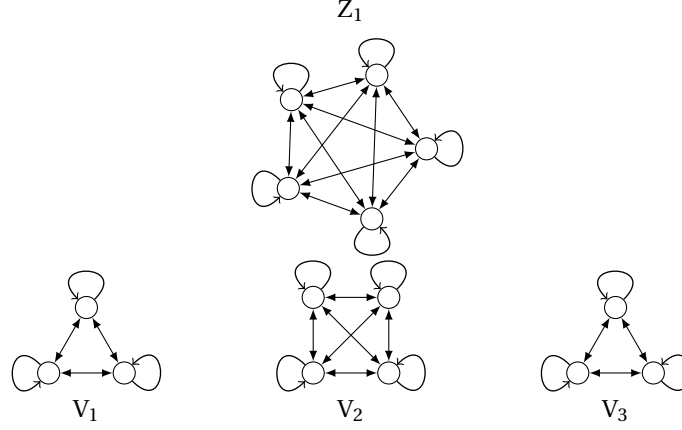


Figure 5: Construction from Theorem 5.6 with partition instance  $(3, 4, 3)$  and  $l = 3$ .

**Theorem 5.7.** *It is NP-hard to compute the number  $cn(D)$  of a given digraph  $D$  even if it does not contain loops.*

*Proof.* Let  $(a_1, \dots, a_n)$  be an instance of the partition problem and let  $m := \frac{\sum_i a_i}{2}$ . We can clearly assume  $m \in \mathbb{N}$  (otherwise we have a NO instance). We can also assume that  $a_i > 2$  for  $i = 1, \dots, n$  (if this is not the case, consider the equivalent instance  $(3a_1, \dots, 3a_n)$ ). We construct a loop-free digraph  $D$  such that  $cn(D) = m$  if and only if  $(a_1, \dots, a_n)$  is a yes instance of partition.

The digraph  $D$  contains, for each  $i = 1, \dots, n$ , the biorientation of the complete graph on  $a_i$  vertices. Let  $V_1, \dots, V_n$  be their vertex sets and let  $V := \cup_i V_i$ . The digraph  $D$  also contains 3 disjoint copies of an independent set on  $m$  vertices. Let  $I_1, I_2, I_3$  be their vertex sets. The digraph  $D$  contains arcs from every vertex in  $I_j$  to every vertex in  $I_{j+1}$  for  $j = 1, 2$ ; arcs from every vertex in  $V$  to every vertex in  $I_1$  and arcs from every vertex in  $I_3$  to every vertex in  $V$ . We show that  $cn(D) = m$  if and only if the partition instance is true. Suppose there exists a solution  $X$  to the partition problem: a winning strategy using  $m$  cops is  $W_1 = I_1, W_2 = I_1, W_3 = \cup_{i \in X} V_i, W_4 = \cup_{i \notin X} V_i, W_5 = I_3, W_6 = I_3$ . To see this, observe that  $R_1 = V \cup I_2 \cup I_3; R_2 = V \cup I_3; R_3 = I_1 \cup (\cup_{i \notin X} V_i); R_4 = I_1 \cup I_2; R_5 = I_2, R_6 = \emptyset$ . Therefore this is a winning strategy on  $D$  using  $m$  cops, which are necessary by Theorem 5.4.

Assume partition is false and let  $(W_t)_{t \geq 1}$  be a cop strategy using  $m$  cops. For a subset of vertices  $S$ , we say that  $S(t)$  is available (to the robber) if  $R_t \cap S \neq \emptyset$ . We show that the strategy cannot be winning by inductively showing that for  $t \geq 1$ , if  $V(t)$  is not available, then  $I_1(t)$  and  $I_3(t)$  are available, therefore  $R_t \neq \emptyset$ .

For  $t = 1$ ,  $V(t)$  is available since  $|V| > m$ .

Assume  $V(t+1)$  is not available. Since partition is false, there are at least three different time steps  $s \leq t+1$  in which  $W_s \cap V \neq \emptyset$ .

To see this, assume, by contradiction, that there are at most two time steps  $s', s''$  when some cop is in  $V$ . Let  $W', W''$  be the set of cops in  $V$  at time step  $s', s''$  respectively. Let  $\alpha_i := |W' \cap V_i|$ . Note that, since the  $V_i$  are vertex sets of complete digraphs,

if  $\alpha_i \leq a_i - 2$ , then  $W'' \supseteq V_i$ ; moreover if  $\alpha_i = a_i - 1$ , then  $|W'' \cap V_i| \geq a_i - 1$ . Since  $|W'|, |W''| \leq m$ ,

$$\begin{aligned} 2m \geq \sum_{i=1}^n (|W' \cap V_i| + |W'' \cap V_i|) &\geq \sum_{\alpha_i = a_i} (a_i + 0) + \sum_{\alpha_i = 0} (0 + a_i) + \\ &+ \sum_{1 \leq \alpha_i \leq a_i - 2} (\alpha_i + a_i) + \sum_{\alpha_i = a_i - 1} 2(a_i - 1) > \sum_{i=1}^n a_i = 2m \end{aligned}$$

which is contradictory. The last inequality is strict because  $a_i > 2$  and there must be at least one index for which  $1 \leq \alpha_i \leq a_i - 1$ , otherwise the set  $X := \{i \mid \alpha_i = a_i\}$  would be a solution of the partition instance.

Let  $t_1 < t_2 < t_3 \leq t + 1$  be the last three time steps before  $t + 1$  such that  $W_s \cap V \neq \emptyset$ . We have that

$$\forall \tau \text{ s.t. } t_1 \leq \tau < t + 1, \quad I_3(\tau) \text{ is not available} \quad (1)$$

otherwise, since  $N^+(I_3) = V$ ,  $t_1, t_2, t_3$  are not maximal. Note that, using the induction hypothesis, (1) means that  $V(\tau)$  is available for  $t_1 \leq \tau < t + 1$ .

Now  $t_3 = t + 1$ , because if  $t_3 < t + 1$ , then, given the maximality of  $t_3$  and the fact that  $V(t + 1)$  is not available, we would have  $V(t_3)$  is not available, contradicting (1). Note that  $V(t_1), I_1(t_1), I_2(t_1)$  are available if  $t_1 = 1$ , because  $W_{t_1} \cap V \neq \emptyset$ . Assume  $t_1 > 1$ : if  $V(t_1 - 1)$  is available, then  $I_1(t_1)$  is available ( $W_{t_1} \cap V \neq \emptyset$ ); if  $V(t_1 - 1)$  is not available, then by induction hypothesis,  $I_1(t_1 - 1)$  is available and so  $I_2(t_1)$  is available. It can be proved using another induction argument that for every  $\tau$  with  $t_1 \leq \tau < t_2$ , either  $I_1(\tau)$  or  $I_2(\tau)$  is available: it is true for  $\tau = t_1$ , moreover if  $I_1(\tau)$  is available, then one among  $I_1(\tau + 1), I_2(\tau + 1)$  must be available (recall that  $V(\tau)$  is available); if  $I_2(\tau)$  is available, then, by (1),  $W_{\tau+1} = I_3(\tau + 1)$ , so  $I_1(\tau + 1)$  is available. Note that, since  $W_{t_2} \cap V \neq \emptyset$ ,  $I_2(t_2 - 1)$  cannot be available, otherwise  $I_3(t_2)$  would be available, contradicting (1). Therefore  $I_1(t_2 - 1)$  (and  $V(t_2 - 1)$ ) are available. It follows that  $I_2(t_2), I_1(t_2), V(t_2)$  are available. By inductive reasoning, using (1), we have that  $W_\tau = I_3$  and  $I_2(\tau), I_1(\tau), V(\tau)$  available for every  $t_2 < \tau < t_3$ . Thus  $I_2(t_3 - 1), I_1(t_3 - 1), V(t_3 - 1)$  are available, so  $I_3(t_3)$  and  $I_1(t_3)$ , namely  $I_3(t + 1)$  and  $I_1(t + 1)$ , are available ( $W_{t_3} \cap V \neq \emptyset$ ). This settles the inductive step  $\square$

The complexity of computing the cop number in the hunters and rabbit game, namely the restriction of our game to undirected loopless graphs, is unknown. Nevertheless, by Theorem 5.7, we have that it is NP-hard to compute the cop number in the directed version of hunters and rabbit.

## 6 Tensor products

Given two digraphs  $G$  and  $H$ , the tensor product  $G \times H$  has vertex set  $V(G) \times V(H)$  and contains an arc from  $(x, y)$  to  $(x', y')$  if there is an arc from  $x$  to  $x'$  in  $G$  and an arc from  $y$  to  $y'$  in  $H$ . For a vertex  $z = (x, y)$  of  $G \times H$ ,  $x$  can be interpreted as the  $G$ -label of  $z$  while  $y$  is its  $H$ -label.

We aim to study some properties of  $ca(G \times H) = d(N(G \times H))$ . We still assume that  $\delta^+(G) \geq 1$  and  $\delta^+(H) \geq 1$  to make the game possible on the considered digraphs.

Let  $n(G)$  denote the number of vertices of a digraph  $G$ .

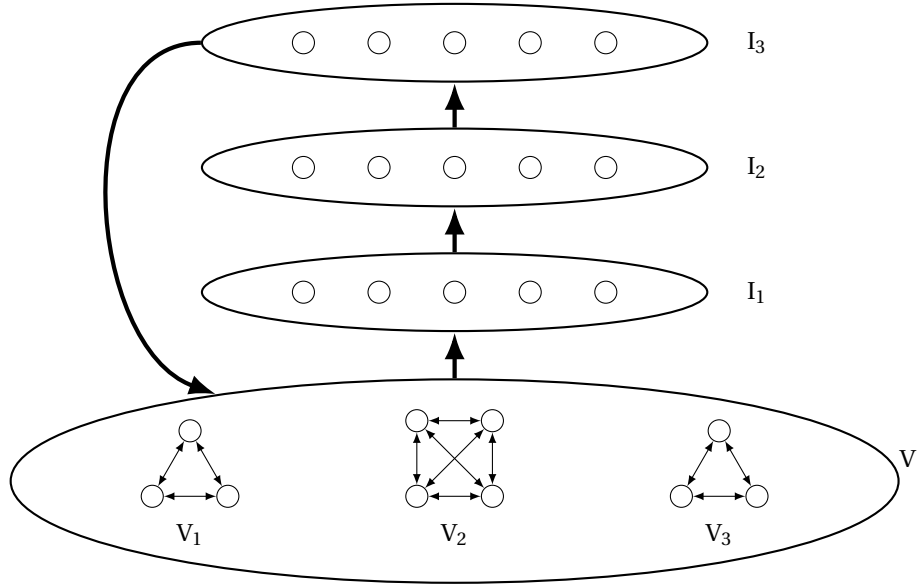


Figure 6: Construction from Theorem 5.7 with partition instance (3, 4, 3). Thick arrows indicate all arcs from one set to the other

**Theorem 6.1.** *Let  $G$  and  $H$  be digraphs such that  $\delta^+(G) \geq 1$  and  $\delta^+(H) \geq 1$ . Then*

$$\frac{ca(G)}{n(G)} \frac{ca(H)}{n(H)} \leq \frac{ca(G \times H)}{n(G \times H)} \leq \min\left(\frac{ca(G)}{n(G)}, \frac{ca(H)}{n(H)}\right).$$

*Proof.* Let us first prove that  $ca(G)ca(H) \leq ca(G \times H)$ . From  $ca(G) = d(N(G))$ , we know that we can build  $ca(G)$  infinitely-long walks in  $G$  that do not meet. Similarly, one can build in  $H$   $ca(H)$  non-meeting infinitely-long walks. For each walk in  $G$  and each walk in  $H$  we can build a walk in  $G \times H$ : the  $i^{th}$  vertex is  $(x_i, y_i)$  where  $x_i$  (resp.  $y_i$ ) is the  $i^{th}$  vertex in the walk through  $G$  (resp.  $H$ ). This provides  $ca(G)ca(H)$  non-meeting infinitely-long walks through  $G \times H$ , showing that  $ca(G)ca(H) \leq ca(G \times H)$ . Let us now prove  $ca(G \times H) \leq \min(ca(G)n(H), ca(H)n(G))$ . We know that  $ca(G)$  attempts are sufficient to capture a robber walking through  $G$ . From Theorem 4.3,  $ca(G)$  vertices of the layered digraph  $\mathcal{L}(G, l)$  where  $l = n(G) - ca(G) + 1$  intersect all  $(s, t)$ -paths. Assume that this set is  $S = \bigcup_{j=1}^l S^j$  where  $S^j \subseteq V(G)^j = \{v^j : v \in V(G)\}$ . Consider now the layered digraph  $\mathcal{L}(G \times H, l = n(G) - ca(G) + 1)$ . The set  $T = \bigcup_{j=1}^l \{(v, u)^j : u \in V(H), v^j \in S^j\}$  is of size  $ca(G)n(H)$ . Any  $(s, t)$ -path in  $\mathcal{L}(G \times H, l)$  induces a path in  $\mathcal{L}(G, l)$  that is necessarily intersected by  $S$ . Since all possible values of the  $H$ -label  $u$  are considered in the  $T$  definition, the set  $T$  intersects all  $(s, t)$ -paths of  $\mathcal{L}(G \times H, l)$  implying that  $ca(G \times H) \leq ca(G)n(H)$ . Similarly, one can show that  $ca(G \times H) \leq ca(H)n(G)$  ending the proof.  $\square$

Since it seems natural that  $ca(G)$  increases when the digraph size is larger, the ratio  $ca(G)/n(G)$  might be more suitable to quantify the difficulty to capture a robber walking through  $G$ . Theorem 6.1 says that by considering the product of two

digraphs, the difficulty cannot exceed the difficulty of either of the two digraphs and stays above the product of the difficulties of the two digraphs.

Note that the lower bound in Theorem 6.1 is reached, for example, if  $ca(G) = n(G)$ , or  $ca(H) = n(H)$  (the lower and upper bound are then equal), or  $ca(G) = 1$  or  $ca(H) = 1$  (in the last case,  $H$  contains exactly one loop that is necessarily reached by the robber after some time, implying that for any walk in  $G \times H$  the  $H$ -label becomes constant after some time).

We know from Theorem 4.3 that  $ct(G \times H, c) \leq n(G)n(H) - c + 1$  when using  $c$  attempts, with  $ca(G \times H) \leq c \leq n(G)n(H)$ . More insights are given below.

**Corollary 6.2.** *Let  $G$  and  $H$  be digraphs with outdegree at least 1. Then,*

$$ct(G \times H, ca(G)n(H)) \leq n(G) - ca(G) + 1 \text{ and}$$

$$ct(G \times H, ca(H)n(G)) \leq n(H) - ca(H) + 1.$$

*Proof.* The proof of the upper bound in Theorem 6.1 directly implies the two inequalities of the corollary.  $\square$

We show an upper bound similar to that of Theorem 6.1 for the cop number  $cn(G \times H)$ .

**Theorem 6.3.** *Let  $G$  and  $H$  be digraphs with outdegree at least 1. Then  $\frac{cn(G \times H)}{n(G \times H)} \leq \min\left(\frac{cn(G)}{n(G)}, \frac{cn(H)}{n(H)}\right)$ .*

*Proof.* Let  $(W_i)_{i \geq 1}$  be a winning cop strategy on  $G$  (resp.  $H$ ) using  $cn(G)$  (resp.  $cn(H)$ ) cops, it is easy to see that the strategy  $(W_i \times V(H))_{i \geq 1}$  (resp.  $(V(G) \times W_i)_{i \geq 1}$ ) is a winning strategy for the cops on  $G \times H$ . The required upper bound then follows immediately.  $\square$

Let  $H_m$  be the digraph obtained from the bioorientation of a complete graph on  $m$  vertices by adding a loop on every vertex.

**Lemma 6.4.** *Let  $G$  be a digraph and  $H_m$  be the digraph defined above. Then  $pw(G \times H_m) \geq m \cdot pw(G)$*

*Proof.* Consider a minimum (wrt separation) ordering of  $V(G \times H_m)$ : if there exist  $g \neq g' \in V(G)$ ,  $h', h'', h''' \in V(H_m)$  such that  $(g', h'')$  is before  $(g, h''')$  and after  $(g, h')$  in the ordering, then we can obtain an ordering of (at most) the same separation by moving  $(g, h)$  right before  $(g, h''')$  in the ordering. This is true because  $(g, h''')$  and  $(g, h)$  have the same out-neighbors ( $H_m$  is vertex symmetric). Therefore, by iteratively performing these moves as long as it is possible, we obtain a minimum ordering  $\sigma$  such that, for every  $g \in V(G)$ , the vertices whose first element is  $g$  are consecutive. The separation of  $\sigma$  is at least  $m \cdot pw(G)$ .  $\square$

Let  $\hat{G}$  be the digraph of Figure 4 and recall that  $pw(\hat{G}) = 2$ ,  $cn(\hat{G}) = 1$ . Let  $G_m = \hat{G} \times H_m$ . Combining Lemma 6.4 and Theorem 6.3 we get a family of digraphs  $G_m$  such that  $cn(G_m) \leq m$  and  $pw(G_m) \geq 2m$ , implying that  $cn(G_m) \leq \frac{pw(G_m)}{2}$ .

## 7 Conclusion

Let us summarize the main contributions of the paper:

- We defined a new variant of the cops and robber game on digraphs where the robber is invisible and slow (one mandatory move along an arc at each time step) while the cops can freely move (helicopter model). The new game can be seen as a more general version of the hunters and rabbit game.
- We proved that it is NP-hard to compute the minimum number of cops required to capture the robber. The problem remains difficult even when capture is required before some constant time limit.
- We showed that the cop number is at most 1 plus the pathwidth ( $pw(D)$ ) of the graph while it can be arbitrarily smaller than  $pw$ . If a monotone capture is required, then at least  $pw$ , and at most  $1 + pw$  cops are required to capture the robber.
- We observed that the cop number is at least the minimum outdegree of any induced subdigraph, a graph parameter that can be easily computed. It is also less than or equal to the size of a minimum feedback vertex set.
- We studied the minimum number of capture attempts  $ca(D)$  required to capture the robber. We observed that this number is easy to compute since it is equal to the dimension of the no-meet matroid which is nothing but the maximum order of a collection of vertex-disjoint cycles. When a time limit  $l$  is considered, the minimum number of capture attempts  $ca(D, l)$  is still easy to compute.
- We proved that the capture time is at most  $n - c + 1$  for a number of attempts  $c \geq ca(D)$ . This implies the main result of the paper: among any set of  $w > ca(D)$  synchronous walks through a digraph  $D$ , there are at least two walks that meet at time at most  $n - w + 2$ .
- We analyzed the cop number, the minimum number of captures  $ca(D)$ , and the capture time on digraph tensor products.

## Acknowledgments

We would like to thank the anonymous reviewers for many valuable comments and suggestions that have led to a substantially improved paper.

## References

- [1] T. V. Abramovskaya, F. V. Fomin, P. A. Golovach, and M. Pilipczuk. How to hunt an invisible rabbit on a graph. *European Journal of Combinatorics*, 52:12–26, 2016.

- [2] J. Bang-Jensen and G. Z. Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.
- [3] J. Barát. Directed path-width and monotonicity in digraph searching. *Graphs and Combinatorics*, 22(1):161–172, 2006.
- [4] W. Ben-Ameur, N. Kushik, A. Maddaloni, J. Neto, and D. Watel. The no-meet matroid. *Discrete Applied Mathematics*, 2022. doi: <https://doi.org/10.1016/j.dam.2022.12.008>.
- [5] D. Berwanger, A. Dawar, P. Hunter, and S. Kreutzer. Dag-width and parity games. In *Proceedings of the 23rd Annual Conference on Theoretical Aspects of Computer Science*, STACS’06, page 524–536, Berlin, Heidelberg, 2006. Springer-Verlag.
- [6] H. Bodlaender, J. Gilbert, H. Hafsteinsson, and T. Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995.
- [7] J. Bolkema and C. Groothuis. Hunting rabbits on the hypercube. *Discrete Mathematics*, 342(2):360–372, 2019.
- [8] A. Bonato. *An Invitation to Pursuit-Evasion Games and Graph Theory*. American Mathematical Society, 2020.
- [9] A. Bonato, P. Golovach, G. Hahn, and J. Kratochvíl. The capture time of a graph. *Discrete Mathematics*, 309(18):5588–5595, 2009.
- [10] S. Brandt, Y. Emek, J. Uitto, and R. Wattenhofer. A tight lower bound for the capture time of the cops and robbers game. *Theoretical Computer Science*, 839:143–163, 2020.
- [11] J. R. Britnell and M. Wildon. Finding a princess in a palace: a pursuit-evasion problem. *Electr. J. Comb.*, 20(1), 2013.
- [12] N. H. Bshouty, L. Higham, and J. Warpechowska-Gruca. Meeting times of random walks on graphs. *Information Processing Letters*, 69(5):259–265, 1999.
- [13] T. Dissaux, F. Fioravantes, H. Gahlawat, and N. Nisse. Recontamination Helps a Lot to Hunt a Rabbit. In J. Leroux, S. Lombardy, and D. Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.
- [14] F. V. Fomin and P. A. Golovach. Graph searching and interval completion. *SIAM Journal on Discrete Mathematics*, 13(4):454–464, 2000.
- [15] F. V. Fomin and D. M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science*, 399(3):236–245, 2008.

- [16] F. V. Fomin, P. A. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theoretical Computer Science*, 411(7):1167–1181, 2010.
- [17] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [18] V. Gruslys and A. Méroneh. Catching a mouse on a tree. *arXiv e-prints*, art. arXiv:1502.06591, Feb. 2015. doi: 10.48550/arXiv.1502.06591.
- [19] R. Janssen. The burning number of directed graphs: bounds and computational complexity. *Theory and Applications of Graphs*, 7(1), 2020.
- [20] N. G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42(6):345–350, 1992.
- [21] W. B. Kinnersley. Bounds on the length of a game of cops and robbers. *Discrete Mathematics*, 341(9):2508–2518, 2018.
- [22] L. M. Kirousis and C. H. Papadimitriou. Interval graphs and searching. *Discrete Mathematics*, 55(2):181–184, 1985.
- [23] M. Mamino. On the computational complexity of a game of cops and robbers. *Theoretical Computer Science*, 477:48–56, 2013.
- [24] J. Obdržálek. Dag-width: Connectivity measure for directed graphs. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, SODA '06*, page 814–821, USA, 2006. Society for Industrial and Applied Mathematics. ISBN 0898716055.
- [25] T. Ohwa. Exact computation for meeting times and infection times of random walks on graphs. *Pacific Journal of Mathematics for Industry*, 7(1):5, 2015.
- [26] A. Quillot. *Jeux et pointes fixes sur les graphes*. PhD thesis, Université Paris VI, 1978.
- [27] P. Seymour and R. Thomas. Graph searching and a min-max theorem for tree-width. *Journal of Combinatorial Theory, Series B*, 58(1):22–33, 1993.
- [28] M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '10*, page 939–948, New York, NY, USA, 2010. Association for Computing Machinery.
- [29] R. Stern. *Multi-Agent Path Finding – An Overview*, pages 96–115. Springer International Publishing, Cham, 2019.
- [30] P. Surynek. An optimization variant of multi-robot path planning is intractable. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1261–1263, Jul. 2010.

- [31] B. Yang and Y. Cao. Digraph searching, directed vertex separation and directed pathwidth. *Discrete Applied Mathematics*, 156(10):1822–1837, 2008.
- [32] J. Yu and S. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1):1443–1449, Jun. 2013.