

The capture time of the hypercube

Anthony Bonato William B. Kinnersley Paweł Prałat*

Department of Mathematics
Ryerson University
Toronto, ON, Canada

{abonato, wkinners, pralat}@ryerson.ca

Przemysław Gordinowicz

Institute of Mathematics
Technical University of Lodz
Łódź, Poland

pgordin@p.lodz.pl

Submitted: Nov 24, 2012; Accepted: Apr 19, 2013; Published: Apr 30, 2013

Mathematics Subject Classifications: 05C57, 05C80

Abstract

In the game of Cops and Robbers, the capture time of a graph is the minimum number of moves needed by the cops to capture the robber, assuming optimal play. We prove that the capture time of the n -dimensional hypercube is $\Theta(n \ln n)$. Our methods include a novel randomized strategy for the players, which involves the analysis of the coupon-collector problem.

Keywords: Cops and Robbers, hypercube, coupon-collector problem

1 Introduction

The game of Cops and Robbers (defined, along with all the standard notation, at the end of this section) is usually studied in the context of the minimum number of cops needed to have a winning strategy, or *cop number*. The cop number is often a challenging graph parameter to analyze, and establishing upper bounds for this parameter is the focus of Meyniel's conjecture: the cop number of a connected n -vertex graph is $O(\sqrt{n})$. For additional background on Cops and Robbers and Meyniel's conjecture, see the book [3].

Our focus in the present article is not on the number of cops needed, but rather, how long it takes them to win. To be more precise, the *length* of a game is the number of rounds

*Supported by grants from NSERC and Ryerson

it takes (not including the initial or 0th round) to capture the robber (the degenerate case is the game played on K_1 , which has length 0). We say that a play of the game with $c(G)$ cops is *optimal* if its length is the minimum over all possible strategies for the cops, assuming the robber is trying to evade capture for as long as possible (here $c(G)$ denotes the cop number of G). There may be many optimal plays possible (for example, on the path P_4 with four vertices, the cop may start on either of the two vertices in the centre), but the length of an optimal game is an invariant of G . When k cops play on a graph with $k \geq c(G)$, we denote this invariant $\text{capt}_k(G)$, which we call the *k-capture time* of G . In the case $k = c(G)$, we just write $\text{capt}(G)$ and refer to this as the *capture time* of G . Note that $\text{capt}_k(G)$ is monotonically non-increasing with k . The capture time parameters may be viewed as temporal counterparts to the cop number, and were introduced in [2].

To date, little is known for exact values or even bounds on the capture times of various graphs. In [2], the authors proved that if G is cop-win (that is, has cop number 1) of order $n \geq 5$, then $\text{capt}(G) \leq n - 3$. By considering small-order cop-win graphs, the bound was improved to $\text{capt}(G) \leq n - 4$ for $n \geq 7$ in [5]. Examples were given of planar cop-win graphs in both [2, 5] which prove that the bound of $n - 4$ is optimal. Beyond the case $k = 1$, a recent paper of Mehrabian [9] investigates the 2-capture time of grids (that is, Cartesian products of paths). It was shown in [9] that if G is the Cartesian product of two trees, then $\text{capt}(G) = \lfloor \text{diam}(G)/2 \rfloor$; hence, the 2-capture time of an $m \times n$ grid is $\lfloor \frac{m+n}{2} \rfloor - 1$. Other recent works on capture time where a visible or an invisible robber moves via a random walk are [6, 7].

The question of determining the capture time of higher dimensional grids was left open in [9]. Perhaps the simplest such graphs are the hypercubes, written Q_n , which may be viewed as the n -fold Cartesian products of K_2 . In the present paper we derive the asymptotic order of the capture time of the hypercube. It was established in [8] that the cop number of a Cartesian product of n trees is $\lceil \frac{n+1}{2} \rceil$; in particular, $c(Q_n) = \lceil \frac{n+1}{2} \rceil$. We note that the cop numbers of Cartesian and other graph products were investigated first in [10].

Our main result is the following theorem, whose proof is deferred to the next sections.

Theorem 1. *For n an integer, we have that*

$$\text{capt}(Q_n) = \Theta(n \ln n).$$

We consider separately the upper and lower bounds on $\text{capt}(Q_n)$; see Sections 2 and 3, respectively. The techniques used for the lower bound are especially novel, as we assume the robber moves randomly (essentially, as a random walk in the graph). Although a priori this may be counterintuitive (an optimal strategy would be for the robber to move away from the cops, which fails to occur with a given probability in a random strategy), we show that with positive probability the robber has a strategy to survive long enough to achieve the lower bound; the analysis involves consideration of the coupon-collector problem and a tail inequality on the expected amount of time needed to collect the coupons.

1.1 Definitions and notation

We consider only finite, reflexive, undirected graphs in the paper. For background on graph theory, the reader is directed to [13].

The game of *Cops and Robbers* was independently introduced in [11, 12] and the cop number was introduced in [1]. The game is played on a reflexive graph; that is, vertices each have at least one loop. Multiple edges are allowed, but make no difference to the play of the game, so we always assume there is exactly one edge between adjacent vertices. There are two players consisting of a set of *cops* and a single *robber*. The game is played over a sequence of discrete time-steps or *turns*, with the cops going first on turn 0 and then playing on alternate time-steps. A *round* of the game is a cop move together with the subsequent robber move. The cops and robber occupy vertices; for simplicity, we often identify the player with the vertex they occupy. We refer to the set of cops as C and the robber as R . When a player is ready to move in a round they must move to a neighbouring vertex. Because of the loops, players can *pass*, or remain on their own vertices. Observe that any subset of C may move in a given round. The cops win if after some finite number of rounds, one of them can occupy the same vertex as the robber (in a reflexive graph, this is equivalent to the cop landing on the robber). This is called a *capture*. The robber wins if he can evade capture indefinitely. A *winning strategy for the cops* is a set of rules that if followed, result in a win for the cops. A *winning strategy for the robber* is defined analogously.

If we place a cop at each vertex, then the cops are guaranteed to win. Therefore, the minimum number of cops required to win in a graph G is a well-defined positive integer, named the *cop number* (or *copnumber*) of the graph G . We write $c(G)$ for the cop number of a graph G .

We use the notation $\lg n$ for the logarithm of n in base 2, and $\ln n$ for the logarithm in the natural base.

2 Upper bound

We establish an upper bound on the capture time of the hypercube by using a strategy for the cops which is analogous to the one described in [8] (although the reader need not be familiar with that strategy to understand ours). We obtain our result as a special case of a more general upper bound.

We begin with a lemma, which concerns a notion closely related to the capture time. The *eccentricity* of a vertex u in a connected graph G is the maximum distance between u and another vertex of G . The *radius* of G , written $\text{rad}(G)$, is the minimum eccentricity of a vertex of G . A *central vertex* in a graph of radius r is one whose eccentricity is r .

Lemma 2. *Let T_1 and T_2 be trees, and consider the game played in $T_1 \square T_2$ with a single cop. If the robber moves at least $\text{rad}(T_1 \square T_2)$ times in the course of the game, then the cop can capture him (and can do so immediately after the robber's last move).*

Proof. We use the cop strategy suggested in [8]. The cop starts on a central vertex in $T_1 \square T_2$. On the cop's turn, let (u, v) denote the cop's position and (u', v') the robber's. Let $d_1 = \text{dist}_{T_1}(u, u')$, let $d_2 = \text{dist}_{T_2}(v, v')$, and let $d = d_1 + d_2$. If d is even, then the cop stands still. Otherwise, if $d_1 > d_2$, then the cop approaches the robber by changing the first coordinate of his position; if $d_2 > d_1$, then the cop approaches by changing his second coordinate.

By the cop's strategy, he might move on his first turn, but afterward moves only after the robber moves. By symmetry, we may suppose $d_1 \geq d_2$ on the cop's first turn. So long as $d_1 > d_2$, the cop will change only the first coordinate of his position until, after some cop move, $d_1 = d_2$. After this point in the game, the cop's strategy ensures that $d_1 = d_2$ after each of the cop's turns, and hence, if the robber ever agrees with the cop in either coordinate, he will be captured on the cop's subsequent turn. Consequently, when we restrict attention to the first coordinates of the cop's positions (that is, when we consider the projection onto T_1), the cop follows a path in T_1 . Likewise, the cop follows a path in the projection onto T_2 . Thus, the cop follows a shortest path in $T_1 \square T_2$ to the vertex at which he eventually captures the robber. Hence, the cop moves no more than $\text{rad}(T_1 \square T_2)$ times before capturing the robber. \square

Note that Lemma 2 does not actually bound the capture time of $T_1 \square T_2$. In fact one cop cannot, in general, capture a robber on the Cartesian product of two trees. However, avoiding a single cop requires the robber to remain still infinitely often. The point of Lemma 2 is that if the robber is forced to move often enough, then he can be captured. Moreover, in this case we cannot bound the length of the game, since the robber can evade capture arbitrarily long; rather, Lemma 2 bounds only the number of times the cop must himself move to a new vertex. We make use of this (rather technical) lemma in the proof of the following result.

Theorem 3. *For every integer n , and all trees T_0, T_1, \dots, T_{n-1} , we have that*

$$\text{capt}(T_0 \square T_1 \square \dots \square T_{n-1}) \leq \left(\sum_{i=0}^{n-1} \text{rad}(T_i) \right) \lceil \lg n \rceil - \left\lfloor \frac{n-1}{2} \right\rfloor + 1.$$

Before proving Theorem 3, we introduce some notation. We view the vertices of the product $T_0 \square T_1 \square \dots \square T_{n-1}$ as n -tuples with coordinates indexed by $0, 1, \dots, n-1$. Now consider the binary representations of the coordinate indices, each using $\lceil \lg n \rceil$ bits (where we allow leading zeros). Let s be a binary sequence of length at most $\lceil \lg n \rceil$. We denote by $\text{start}(s)$ the set of coordinate indices whose binary representations start with s ; that is, $\text{start}(s)$ equals

$$\{i \in \mathbb{Z} : 0 \leq i \leq n-1 \text{ and } s \text{ is a prefix of the } \lceil \lg n \rceil\text{-bit binary representation of } i\}.$$

Let \circ denote the concatenation operator on binary strings.

Proof of Theorem 3. When $n \leq 2$ the desired bound was proved in [9], so suppose $n \geq 3$. Let $k = c(T_0 \square T_1 \square \dots \square T_{n-1}) = \lceil \frac{n+1}{2} \rceil$, and label the k cops C_0, C_1, \dots, C_{k-1} . Each cop's strategy has two steps.

Step 1: The cop attempts to occupy a vertex that disagrees with the robber's vertex in only one or two coordinates.

More precisely, for $0 \leq i \leq k - 2$, cop C_i seeks to agree with the robber in all coordinates except $2i$ and $2i + 1$. If n is odd, then C_{k-1} wants to agree with the robber in all coordinates except $n - 1$; if instead n is even, then C_{k-1} wants to agree in all coordinates except $n - 2$ and $n - 1$, just like cop C_{k-2} . Since the specific details of this step are more complex, we postpone them until after we discuss Step 2.

Step 2: The cop tries to actually capture the robber.

Suppose first that n is odd. For $1 \leq i \leq k - 2$, cop C_i initially agrees with the robber in all coordinates except $2i$ and $2i + 1$ (this was the goal of Step 1). If the robber stands still, then so does C_i . Suppose instead that the robber changes coordinate j of his position. If $j \notin \{2i, 2i + 1\}$, then C_i also changes coordinate j , in order to maintain his agreement with the robber. Finally, if $j \in \{2i, 2i + 1\}$, then C_i also changes one of these coordinates, in accordance with the strategy on $T_{2i} \square T_{2i+1}$ described in Lemma 2. (That is, C_i considers the projections of both players' positions onto coordinates $2i$ and $2i + 1$, and plays as he would on this subgraph.) Cop C_{k-1} plays as described above, except that when the robber stands still, he plays in accordance with an optimal strategy on T_{n-1} (just as he does when the robber changes coordinate $n - 1$).

When n is even, cop C_i plays just as above for $0 \leq i \leq k - 3$. Cops C_{k-2} and C_{k-1} play similarly, but with some small changes. If the robber changes coordinate j for $j \notin \{n - 2, n - 1\}$, then C_{k-2} and C_{k-1} both change coordinate j to maintain their agreement with the robber. Otherwise, (including when the robber stands still) C_{k-2} and C_{k-1} together play in accordance with an optimal strategy on $T_{n-2} \square T_{n-1}$.

Now, for the completion of Step 1, the cops move as follows. For $0 \leq i \leq n - 1$, let v_i be a central vertex in T_i ; initially all k cops start at vertex (v_1, v_2, \dots, v_n) . Each cop approaches the robber in $\lceil \lg n - 1 \rceil$ "phases". During each phase, each cop deems certain coordinates *active*; active coordinates remain active throughout all subsequent phases. A coordinate that is not active is *inactive*. The cop aims to occupy a vertex that agrees with the robber's vertex in every active coordinate; once the cop has done so, he proceeds to the next phase (and chooses additional active coordinates). On the robber's turn, he either stands still or changes exactly one coordinate of his position. If he changes an active coordinate in which he and the cop previously agreed, then the cop changes that same coordinate; in all other cases, the cop approaches the robber in the direction of the first active coordinate in which he and the robber disagree. (Since each T_i is a tree, the cop's move is uniquely determined.) Thus, once the cop and robber agree in an active coordinate, the cop maintains this agreement for the remainder of the game.

We next specify how the cops choose active coordinates in Step 1. Fix a cop C_i with $0 \leq i \leq k - 2$ (we return to cop C_{k-1} later). Recall that C_i seeks to eventually agree with the robber in all coordinates other than $2i$ and $2i + 1$; consequently, he will never deem these coordinates active. During phase j , for $1 \leq j \leq \lceil \lg n \rceil - 1$, a coordinate is active for C_i precisely when it differs from $2i$ in one of the first j bits. (In some cases, a cop has the

same active coordinates in phase k as in phase $k - 1$; in such cases, the cop finishes phase k immediately, and proceeds directly to phase $k + 1$.) For example, when $n = 11$, there are three phases and six cops; all cops' active coordinates in each phase are presented in Table 1. Note that in phase $\lceil \lg n \rceil - 1$, cop C_i 's only remaining inactive coordinates are $2i$ and $2i + 1$. When n is odd, cop C_{k-1} uses the same approach; in phase $\lceil \lg n \rceil - 1$, only coordinate $n - 1$ remains inactive. When n is even, C_{k-1} moves identically to C_{k-2} throughout Step 1.

Cop	Phase 1	Phase 2	Phase 3
C_0	$\{8, 9, 10\}$	$\{4, 5, 6, 7, 8, 9, 10\}$	$\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$
C_1	$\{8, 9, 10\}$	$\{4, 5, 6, 7, 8, 9, 10\}$	$\{0, 1, 4, 5, 6, 7, 8, 9, 10\}$
C_2	$\{8, 9, 10\}$	$\{0, 1, 2, 3, 8, 9, 10\}$	$\{0, 1, 2, 3, 6, 7, 8, 9, 10\}$
C_3	$\{8, 9, 10\}$	$\{0, 1, 2, 3, 8, 9, 10\}$	$\{0, 1, 2, 3, 4, 5, 8, 9, 10\}$
C_4	$\{0, 1, 2, 3, 4, 5, 6, 7\}$	$\{0, 1, 2, 3, 4, 5, 6, 7\}$	$\{0, 1, 2, 3, 4, 5, 6, 7, 10\}$
C_5	$\{0, 1, 2, 3, 4, 5, 6, 7\}$	$\{0, 1, 2, 3, 4, 5, 6, 7\}$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Table 1: Active coordinates during play on Q_{11} .

To bound the number of rounds needed to complete Step 1, we group the cops into *squads*. A squad is a collection of cops having the same inactive coordinates. For example, from Table 1 in Phase 1 we have one squad formed by C_0, C_1, C_2 , and C_3 , and the second squad formed by C_4 and C_5 . For $1 \leq i \leq k$ and $1 \leq j \leq \lceil \lg n \rceil - 1$, coordinate x is inactive for cop C_i during phase j precisely when x agrees with i in the first j bits. Thus, we may identify each possible squad by a binary sequence of length at most $\lceil \lg n \rceil - 1$; at any given point in the game, for a binary sequence s , let S_s denote the squad of cops having inactive coordinates $\text{start}(s)$.

Initially we have two nonempty squads, S_0 and S_1 , and all cops begin in phase 1. Since all cops begin at the same vertex, and cops in the same squad move identically, all members of a squad complete their phases simultaneously. When the cops in S_0 finish phase 1, they all leave S_0 and join either S_{00} or S_{01} . Likewise, when S_{01} finishes phase 2, it splits into S_{010} and S_{011} . In general, for s a binary string with length j , such that $\text{start}(s) \neq \emptyset$, squad S_s is nonempty only while its members are in phase j . Once the squad completes phase j (for $j < \lceil \lg n \rceil - 1$), it splits into squads S_{s0} and S_{s1} . (Note that one of S_{s0} or S_{s1} could remain empty. For example, on Q_{11} , squad S_{11} is empty throughout the game; this is due to the fact that $\text{start}(11) = \emptyset$.)

Fix a binary string s of length j (for $1 \leq j \leq \lceil \lg n \rceil - 1$) such that $\text{start}(s) \neq \emptyset$. Let \hat{s} be s with the last bit removed, and let s' be s with the last bit flipped. Squad S_s becomes nonempty when the members of squad $S_{\hat{s}}$ finish phase $j - 1$ (or, in the case $j = 1$, at the beginning of the game). At this point, S_s splits into S_s and $S_{s'}$. Thus, at the beginning of phase j , each member of S_s agrees with the robber in all but at most $|\text{start}(s')|$ active coordinates. Moreover, for $i \in \text{start}(s')$, each cop in S_s must make no more than $\text{rad}(T_i)$ moves in the direction of coordinate i before he agrees with the robber in that coordinate. Hence, the members of S_s finish phase j after at most N turns on which the robber either sits still or moves in the direction of some coordinate in $\text{start}(s)$,

where $N = \sum_{i \in \text{start}(s')} \text{rad}(T_i)$.

At any point in the game, the inactive coordinates (across all nonempty squads) partition $\{0, 1, \dots, n-1\}$: this is clear at the beginning of the game, and is maintained under the splitting of squads. Hence, on each robber move, at least one cop makes progress toward finishing either Step 1 or Step 2. The number of rounds in which some cop makes progress in Step 1 is at most $\sum_s \sum_{i \in \text{start}(s')} \text{rad}(T_i)$, where the sum is over all binary strings s with length at most $\lceil \lg n \rceil - 1$. However, this upper bound can be simplified as follows:

$$\begin{aligned} \sum_s \sum_{i \in \text{start}(s')} \text{rad}(T_i) &= \sum_{j=1}^{\lceil \lg n \rceil - 1} \sum_{\substack{s \text{ has} \\ \text{length } j}} \sum_{i \in \text{start}(s')} \text{rad}(T_i) \\ &= \sum_{j=1}^{\lceil \lg n \rceil - 1} \sum_{i=0}^{n-1} \text{rad}(T_i) = \left(\sum_{i=0}^{n-1} \text{rad}(T_i) \right) (\lceil \lg n \rceil - 1). \end{aligned}$$

Once cop C_i has started Step 2, for $1 \leq i \leq k-2$, the robber can move in the direction of coordinates $2i$ and $2i+1$ no more than $\text{rad}(T_{2i}) + \text{rad}(T_{2i+1}) - 1$ times before C_i captures him (by Lemma 2). Likewise, if n is odd, then once cop C_{k-1} has entered Step 2, the robber can spend no more than $\text{rad}(T_{n-1}) - 1$ turns either standing still or moving in the direction of coordinate $n-1$; if n is even, then once cops C_{k-2} and C_{k-1} have entered Step 2, the robber can spend no more than $\lfloor \text{diam}(T_{n-2} \square T_{n-1})/2 \rfloor$ turns either standing still or moving in the direction of coordinates $n-2$ and $n-1$. (This follows from the strategy given by Mehrabian [9] for the 2-capture time of Cartesian products of two trees, with the caveat that the cops here require one extra turn to reach the initial positions used in Mehrabian's strategy.) In either case, the number of rounds during which some cop makes progress in Step 2 is bounded above by $\sum_{i=0}^{n-1} \text{rad}(T_i) - \lfloor (n-1)/2 \rfloor + 1$. Thus, we have that

$$\begin{aligned} \text{capt}(T_0 \square T_1 \square \dots \square T_{n-1}) &\leq \left(\sum_{i=0}^{n-1} \text{rad}(T_i) \right) (\lceil \lg n \rceil - 1) + \sum_{i=0}^{n-1} \text{rad}(T_i) - \lfloor (n-1)/2 \rfloor + 1 \\ &= \left(\sum_{i=0}^{n-1} \text{rad}(T_i) \right) \lceil \lg n \rceil - \left\lfloor \frac{n-1}{2} \right\rfloor + 1, \end{aligned}$$

as claimed. □

When $T_0 = T_1 = \dots = T_{n-1} = K_2$, we have the following special case of Theorem 3.

Corollary 4. *For n an integer, we have that*

$$\text{capt}(Q_n) \leq n \lceil \lg n \rceil - \left\lfloor \frac{n-1}{2} \right\rfloor + 1 = (1 + o(1))n \lg n.$$

Before concluding this section, we remark that the ideas in the proof of Theorem 3 can be applied in more generality. Instead of playing the game on $T_0 \square T_1 \square \dots \square T_{n-1}$

for trees T_i , suppose we instead play on $G_0 \square G_1 \square \cdots \square G_{n-1}$ for arbitrary graphs G_i . In the proof of Theorem 3, we had (essentially) $n/2$ cops, each of which sought to agree with the robber on all but two coordinates. In the game on $G_0 \square G_1 \square \cdots \square G_{n-1}$, we can use n teams of cops, with each team seeking to agree with the robber on all but one coordinate. Suppose the team assigned to coordinate i contains k_i cops, where $k_i \geq c(G_i)$, for $0 \leq i \leq n-1$. Once this team finishes Step 1, whenever the robber moves in the direction of coordinate i (or stands still), the cops can change that coordinate according to a strategy realizing $\text{capt}_{k_i}(G_i)$. Eventually, some team captures the robber. These modifications to the argument of Theorem 3 yield the following:

Theorem 5. *For graphs G_0, G_1, \dots, G_{n-1} and integers k_0, k_1, \dots, k_{n-1} with $k_i \geq c(G_i)$, we have*

$$\text{capt}_K(G_0 \square G_1 \square \cdots \square G_{n-1}) \leq \left(\sum_{i=0}^{n-1} \text{rad}(G_i) \right) \lceil \lg n \rceil + \sum_{i=0}^{n-1} (\text{capt}_{k_i}(G_i) - 1) + 1,$$

where $K = k_0 + k_1 + \cdots + k_{n-1}$.

Note that when each G_i is a tree, this result is weaker than Theorem 3 both in terms of the bound on the capture time and in terms of the number of cops used.

3 Lower bound

We next show that $\text{capt}(Q_n) \geq (1 - o(1))\frac{1}{2}n \ln n$. Our proof of this bound involves two strategies, one for the cops and one for the robber. On each turn, a robber using the *random strategy* moves to an adjacent vertex chosen uniformly at random. A cop using the *greedy strategy* moves one step closer to the robber on each turn; when there are multiple ways to do this, the cop chooses between them arbitrarily. (For both strategies, the players may choose their starting locations “intelligently”.) As it turns out, the greedy strategy is in some sense the optimal response to the random strategy; we formalize this intuition below. The probability of an event A is denoted by $\Pr[A]$, and the expected value of a random variable X is denoted by $\mathbb{E}[X]$.

Lemma 6. *Fix T a positive integer. Consider a cop C attempting to capture a robber R using the random strategy on the cube Q_n , and suppose R has the next move. For $d \leq n$, define*

$$p_d = \max \Pr [C \text{ catches } R \text{ in under } T \text{ rounds} \mid C \text{ and } R \text{ are distance } d \text{ apart}],$$

where the maximum is taken over all possible strategies for C . Then the following inequalities hold for $1 \leq k \leq n/2$:

$$(i) \quad p_{2k-2} \geq p_{2k}$$

$$(ii) \quad p_{2k} \geq p_{2k-1}$$

Proof. We prove both claims simultaneously by induction on T . The claims are easily verified when $T = 1$, so fix $T > 1$. For $d \leq n$, define

$$p'_d = \max \Pr [\text{C catches R in under } T - 1 \text{ rounds} \mid \text{C and R are distance } d \text{ apart}],$$

where the maximum is again taken over all strategies for C. Note that always $p'_d \leq p_d$.

For the proof of item (i), by the induction hypothesis, $p'_{2k-2} \geq p'_{2k} \geq p'_{2k-1}$ and $p'_{2k} \geq p'_{2k+2} \geq p'_{2k+1}$. Suppose C is at distance $2k$ from R. Let d be the distance between C and R after one additional round of the game; since the distance between the players cannot change by more than 2 in a single round, $2k - 2 \leq d \leq 2k + 2$. Thus, $p_{2k} \leq p'_{2k-2} \leq p_{2k-2}$.

For item (ii), suppose C and R begin distance $2k$ apart. If C plays greedily, then after one additional round, the distance between the players is $2k - 2$ with probability $2k/n$ and $2k$ with probability $(n - 2k)/n$. Hence, $p_{2k} \geq \frac{2k}{n} p'_{2k-2} + \frac{n-2k}{n} p'_{2k}$.

Suppose instead that C and R begin distance $2k - 1$ apart. If R moves closer to C, then after C's subsequent move, the distance between the players is one of $2k - 3$, $2k - 2$, and $2k - 1$. If instead R moves farther from C, then after C's move, the distance is one of $2k - 1$, $2k$, and $2k + 1$. By the induction hypothesis, $p'_{2k-2} \geq p'_{2k-3}$ and $p'_{2k-2} \geq p'_{2k} \geq p'_{2k-1}$, so $p'_{2k-2} = \max\{p'_{2k-1}, p'_{2k-2}, p'_{2k-3}\}$. Similarly, $p'_{2k} = \max\{p'_{2k-1}, p'_{2k}, p'_{2k+1}\}$. Therefore,

$$p_{2k-1} \leq \frac{2k-1}{n} p'_{2k-2} + \frac{n-2k+1}{n} p'_{2k} \leq \frac{2k}{n} p'_{2k-2} + \frac{n-2k}{n} p'_{2k} \leq p_{2k},$$

which completes the proof. \square

Lemma 6 shows that, against a random robber, the cop should keep the distance between the players even and, subject to that, minimize the distance. We have, therefore, that the following cop strategy is optimal against a random robber: if the robber is an even distance away, sit still; otherwise, play greedily. The lemma shows that this strategy is optimal in that it maximizes the probability of capturing the robber “quickly”, but in fact this optimality can be proved in a stronger sense. A more detailed coupling argument shows that for any cop strategy, there is a greedy strategy that captures the robber no less quickly, assuming identical sequences of robber moves. Since we need only the weaker property established in the lemma, we omit the proof of this stronger claim.

A single greedy cop will eventually capture a random robber. In this scenario, the capturing process behaves similarly to the well-known “coupon-collector” process. In the coupon-collector problem, there are m types of coupons, and the aim is to collect at least one coupon of each type. One coupon is obtained in each round, and the type of coupon is chosen uniformly at random. It is well-known that the expected number of rounds needed to collect all m types of coupon is $(1 + o(1))m \ln m$.

When all but i coupons have already been collected, the probability of obtaining a new coupon on the next turn is i/m , so the number of rounds needed to collect a new coupon is geometrically distributed with probability of success i/m . As we will show, similar behaviour arises when a greedy cop plays against a random robber on Q_n . In our analysis of $\text{capt}(Q_n)$, we will need to bound the probability that the actual capturing time is significantly less than its expectation. For this we use a slight generalization of a result

by Doerr (see Theorem 1.24 in [4]). The proof of this generalization is quite similar to that of Doerr's original result, but we include the full proof for completeness.

Lemma 7. *Consider the coupon-collector process with m coupons in total, of which all but m_0 have already been collected. Let X be a random variable denoting the number of rounds needed to collect the remaining m_0 coupons. For $\varepsilon > 0$, we have that*

$$\Pr[X < (1 - \varepsilon)(m - 1) \ln m] \leq \exp(-m^{-1+\varepsilon}m_0).$$

Proof. Let $T = (1 - \varepsilon)(m - 1) \ln m$. For the i th uncollected coupon, where $i \in [m_0]$, let X_i be an indicator random variable for the event that this type of coupon is collected within T rounds. It is straightforward to see that

$$\begin{aligned} \Pr[X_i = 1] &= 1 - \left(1 - \frac{1}{m}\right)^T \leq 1 - \exp(-T/(m - 1)) \\ &= 1 - \exp(-(1 - \varepsilon) \ln m) = 1 - m^{-1+\varepsilon}, \end{aligned}$$

where the inequality above uses the standard inequality $(1 - 1/m) \geq \exp(-1/(m - 1))$, which is valid for $m > 1$. Fix $I \subseteq [m_0]$ and $j \in [m_0] \setminus I$. Now

$$\begin{aligned} \Pr[X_i = 1 \text{ for all } i \in I] &= \Pr[X_i = 1 \text{ for all } i \in I \mid X_j = 1] \cdot \Pr[X_j = 1] \\ &\quad + \Pr[X_i = 1 \text{ for all } i \in I \mid X_j = 0] \cdot \Pr[X_j = 0]. \end{aligned}$$

But since

$$\Pr[X_i = 1 \text{ for all } i \in I \mid X_j = 0] \geq \Pr[X_i = 1 \text{ for all } i \in I],$$

it follows that

$$\Pr[X_i = 1 \text{ for all } i \in I \mid X_j = 1] \leq \Pr[X_i = 1 \text{ for all } i \in I].$$

Thus,

$$\Pr[X_i = 1 \text{ for all } i \in I \cup \{j\}] \leq \Pr[X_i = 1 \text{ for all } i \in I] \cdot \Pr[X_j = 1].$$

Now induction yields

$$\Pr[X_i = 1 \text{ for all } i \in [m_0]] \leq \prod_{i=1}^{m_0} \Pr[X_i = 1] \leq (1 - m^{-1+\varepsilon})^{m_0} \leq \exp(-m^{-1+\varepsilon}m_0),$$

where the last inequality follows because $(1 + x) \leq \exp(x)$ for all x . □

We are now ready to prove our lower bound on $\text{capt}(Q_n)$, and the proof of this result concludes the proof of Theorem 1. We state the lower bound in greater generality.

Theorem 8. *Fix a positive constant d . On Q_n , a robber can escape capture against n^d cops for at least $(1 - o(1))\frac{1}{2}n \ln n$ rounds.*

Proof. Let $T = \frac{1}{2}(n - 1) \ln n$ and let

$$\varepsilon = \varepsilon(n) = \frac{\ln((4d + 1) \ln n)}{\ln n} = \frac{O(\ln \ln n)}{\ln n} = o(1).$$

Fix a cop strategy. We claim that for sufficiently large n , a random robber evades capture for at least $(1 - \varepsilon)T$ rounds with positive probability; it would then follow that some particular set of random choices, and hence some particular deterministic strategy, allows the robber to survive this long. (Note that we are not concerned with the expected length of the game, but only with the probability that the robber evades capture for “long enough”.)

For the initial placement, the robber aims to choose a starting vertex that is relatively far from *all* cops. The number of vertices of Q_n within distance k of any given cop is $\sum_{i=0}^k \binom{n}{i}$, so the number of vertices within distance k of any of the n^d cops is at most

$$n^d \sum_{i=0}^k \binom{n}{i} \leq n^d (k + 1) \binom{n}{k} \leq 2n^d k \left(\frac{ne}{k}\right)^k \leq n^{d+1} ((4e)^{1/4})^n < n^{d+1} 1.85^n,$$

when $k = n/4$. For sufficiently large n we have $n^d \sum_{i=0}^k \binom{n}{i} < 2^n$, so the robber can choose an initial position at least distance $n/4 + 1$ from every cop. (In fact more careful analysis permits $k = (1 - o(1))n/2$, but using $n/4 + 1$ simplifies later computations without affecting the asymptotics of the final result.)

To analyze the effectiveness of the robber’s strategy, we use the following approach. First, we focus on a single cop. We determine the probability that this cop captures the robber within $(1 - \varepsilon)T$ rounds. We then apply the union bound to obtain an upper bound on the probability that any of the cops can capture the robber within $(1 - \varepsilon)T$ rounds. If this probability is less than 1, then with positive probability the robber evades capture. The theorem then follows, since the robber can survive for at least $(1 - \varepsilon)T$ rounds against any fixed cop strategy.

Suppose cop C is at distance $2k$ from the robber (on the robber’s turn). By Lemma 6, we may assume C uses a greedy strategy. On any given round, the robber moves toward C with probability $2k/n$. Thus, after the robber’s move and C’s response, the distance between the two is $2k - 2$ with probability $2k/n$ and $2k$, otherwise. Letting X_i denote the number of rounds for which the cop remains at distance $2i$ from the robber, we see that X_i is geometrically distributed with probability of success $2i/n$; the number of rounds needed for C to catch the robber is at most $\sum_{i=1}^k X_i$. Hence, this process is equivalent to the coupon-collector process with $n/2$ coupons, if we suppose that all but k coupons have already been collected. Since the robber begins at least distance $n/4 + 1$ from C, we may (again by Lemma 6) take $k = \lceil n/8 \rceil$ to obtain a lower bound on the expected length of the process:

$$\sum_{i=1}^{\lceil n/8 \rceil} \mathbb{E}[X_i] = \sum_{i=1}^{\lceil n/8 \rceil} \frac{n}{2i} = \frac{n}{2} \sum_{i=1}^{\lceil n/8 \rceil} \frac{1}{i},$$

which tends to $\frac{n}{2}(\ln n - \ln 8 + \gamma)$, where $\gamma \approx 0.557$ denotes the Euler-Mascheroni constant.

By Lemma 7, the probability that C captures the robber in under $(1 - \varepsilon)T$ rounds is at most $\exp(-(n/2)^{-1+\varepsilon}n/8)$, which is asymptotically $\exp(-(n/2)^\varepsilon/4)$. By the union bound, the probability that any of the cops captures the robber in under $(1 - \varepsilon)T$ rounds is thus, at most $(1 + o(1))n^d \exp(-(n/2)^\varepsilon/4)$. Since

$$\frac{(n/2)^\varepsilon}{4} = \frac{1}{4} \exp\left(\varepsilon\left(\ln n + O(1)\right)\right) = (1 + o(1))\frac{1}{4}(4d + 1) \ln n > \left(d + \frac{1}{5}\right) \ln n,$$

this probability tends to 0 as n tends to infinity. Hence, with probability approaching 1 as $n \rightarrow \infty$, the robber escapes capture for at least $(1 - \varepsilon)T$ rounds. \square

It remains open to compute the exact value of the capture time of the hypercube Q_n , although Theorem 1 derives it up to a constant factor. We conjecture that $\text{capt}(Q_n) = (1 + o(1))\frac{1}{2}n \ln n$.

References

- [1] M. Aigner, M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics* **8** (1984) 1–12.
- [2] A. Bonato, G. Hahn, P.A. Golovach, J. Kratochvíl. The capture time of a graph. *Discrete Mathematics* **309** (2009) 5588–5595.
- [3] A. Bonato, R.J. Nowakowski. *The Game of Cops and Robbers on Graphs*. American Mathematical Society, Providence, Rhode Island, 2011.
- [4] B. Doerr. Analyzing Randomized Search Heuristics: Tools from Probability Theory. *Series on Theoretical Computer Science* Vol. 1, 1–20, World Scientific, 2011.
- [5] T. Gavenčiak. Cop-win graphs with maximal capture-time. *Discrete Mathematics* **310** (2010) 1557–1563.
- [6] A. Kehagias, D. Mitsche, P. Prałat. Cops and invisible robbers: the cost of drunkenness. *Theoretical Computer Science* **481** (2013), 100–120.
- [7] A. Kehagias, P. Prałat. Some remarks on cops and drunk robbers. *Theoretical Computer Science* **463** (2012), 133–147.
- [8] M. Maamoun, H. Meyniel. On a game of policemen and robber. *Discrete Applied Mathematics* **17** (1987) 307–309.
- [9] A. Merhrabian. The capture time of grids. *Discrete Mathematics* **311** (2011) 102–105.
- [10] S. Neufeld, R. Nowakowski. A game of cops and robbers played on products of graphs. *Discrete Mathematics* **186** (1998) 253–268.
- [11] R.J. Nowakowski, P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics* **43** (1983) 235–239.
- [12] A. Quilliot. Jeux et pointes fixes sur les graphes. Thèse de 3ème cycle, Université de Paris VI, 1978, 131–145.
- [13] D.B. West. *Introduction to Graph Theory*, 2nd edition. Prentice Hall, 2001