Vision-based pursuit-evasion in a grid*

Adrian Dumitrescu† Howi Kok‡ Ichiro Suzuki§ Paweł Żyliński¶

October 29, 2009

Abstract

We revisit the problem of pursuit-evasion in a grid introduced by Sugihara and Suzuki in the line-of-sight vision model. Consider an arbitrary evader Z with the maximum speed of 1 who moves (in a continuous way) on the streets and avenues of an $n \times n$ grid G_n. The cunning evader is to be captured by a group of pursuers, possibly only one. The maximum speed of the pursuers is $s \geq 1$; s is a constant for each pursuit-evasion problem considered, but several values for s are studied. We prove several new results (no such algorithms were available for capture using one, two or three pursuers having a constant maximum speed limit):

(i) A randomized algorithm through which one pursuer A with a maximum speed of $s \geq 3$ can capture an arbitrary evader Z in G_n in expected polynomial time. For instance, the expected capture time is $O(n^{1+\log_{6/5} 16}) = O(n^{16.21})$ for $s = 3$, $O(n^{1+\log 12}) = O(n^{4.59})$ for $s = 4$, $O(n^{1+\log 60/13}) = O(n^{3.21})$ for $s = 6$, and it approaches $O(n^3)$ with the further increase of s.

(ii) A randomized algorithm for capturing an arbitrary evader in $O(n^3)$ expected time using two pursuers who can move slightly faster than the evader ($s = 1 + \varepsilon$, for any $\varepsilon > 0$).

(iii) Randomized algorithms for capturing a certain "passive" evader using either a single pursuer who can move slightly faster than the evader ($s = 1 + \varepsilon$, for any $\varepsilon > 0$), or two pursuers having the same maximum speed as the evader ($s = 1$).

(iv) A deterministic algorithm for capturing an arbitrary evader in $O(n^2)$ time, using three pursuers having the same maximum speed as the evader ($s = 1$).

Keywords: pursuit-evasion game, line-of-sight vision model, grids, randomized algorithms

AMS subject classifications: 68W20, 68Q25, 68R10

1 Introduction

An $n \times n$ grid G_n, $n \geq 2$, is the set of points with integer coordinates in $[0, n-1] \times [0, n-1]$ together with their connecting edges viewed as a connected planar set. Alternatively, G_n can be viewed as the union of the following $2n$ line segments: (a) the line segment between $(i, 0)$ and $(i, n-1)$, called *column* i, $0 \leq i \leq n-1$, and (b) the line segment between $(0, j)$ and $(n-1, j)$, called *row* j, $0 \leq j \leq n-1$. A point (x, y) in G_n is called a *vertex* if both x and y are integers.

*An extended abstract of this paper appeared in: Proceedings of the 11th Scandinavian Workshop on Algorithm Theory (SWAT 2008), *Lecture Notes in Computer Science* 5124, (2008), pp. 53–64.

†Dept. of Computer Science, University of Wisconsin-Milwaukee, WI 53201-0784, USA, email: ad@cs.uwm.edu. Supported in part by NSF CAREER grant CCF-0444188.

‡Dept. of Computer Science, University of Wisconsin-Milwaukee, WI 53201-0784, USA.

§Dept. of Computer Science, University of Wisconsin-Milwaukee, WI 53201-0784, USA, email: suzuki@cs.uwm.edu. Supported in part by UWM Research Growth Initiative.

¶Institute of Informatics, University of Gdańsk, 80-952 Gdańsk, Poland, email: zylinski@inf.ug.edu.pl

We consider a vision-based pursuit-evasion problem in G_n in which a group of *pursuers* (*searchers*) are required to search for and capture an *evader* (*fugitive*). Both the pursuers and evader are represented by a (moving) point in G_n (two players can be at the same point at one time). The vision of the players is limited to a straight line of sight (i.e., a row or column): a player at a vertex can see both the corresponding row and column, while one located in the interior of an edge can see only the row or column containing that edge.[1] A player is said to have a *direction detection capability* if he can see in which direction an opponent moves (left or right) when disappearing from the line of sight. A *distance detection capability* is one that allows a player to know either the exact or an approximate distance between his current location and that of an opponent in sight. Generally, we assume that the pursuers have no direction detection capability, and their distance detection capability is limited — a pursuer can tell only whether or not an evader in sight is within distance 1 of his current position, or they have an approximate distance detection capability with a constant relative error.[2] In contrast, the evader may have both direction detection and exact distance detection capabilities. The pursuers can communicate with one another in real time (without delay). The players can also initiate and start executing any movement without delay. The evader may know the algorithm of the pursuers and their initial positions, but he does not know the outcomes of their random choices, in case they use a randomized algorithm.

In the *discrete model*, the moves are restricted to the vertices of G_n and executed at discrete time steps $t = 0, 1, \ldots$, simultaneously, by each player. A move consists of either moving to an adjacent vertex, or staying at the current vertex. The evader is considered captured if he and a pursuer are either at the same grid point at a discrete time step, or if they traverse the same edge from opposite directions in the interval between two consecutive time steps. In this paper, we consider this problem mainly in the *continuous model*, as opposed to the discrete model. In the continuous model, moves are not restricted to discrete time steps or to the vertices of G_n: any move in G_n is allowed within the speed limit constraint, which is 1 for the evader w.l.o.g., and some constant s for the pursuers. The evader is considered captured if there exists a time during the pursuit when his position coincides with the position of one of the pursuers. Here we consider the case $s \geq 1$, and present algorithms for capturing the evader using one, two or three pursuers.

The vision-based continuous pursuit-evasion problem in a grid described above was first introduced by Sugihara and Suzuki [20] as a variant of the well-known *graph search problem* [12, 13, 17], which is essentially the same problem except that it is played in an arbitrary connected graph by "blind" pursuers and an evader having an unbounded speed. In [20] it is shown that it is possible to capture an arbitrary evader using four pursuers having a maximum speed of $s = 1$. Subsequently, Dawes [6] showed that a single pursuer having a speed of n can *locate* (i.e., see) in G_n an arbitrary evader having full knowledge about the pursuer's move, and later Neufeld [14] improved the speed bound to $\lceil \frac{2(n-1)}{3} \rceil + 2$; no direction and distance detection capabilities are needed for the pursuer, since the game ends as soon as he finds the evader. Dawes also discusses a strategy for the pursuer having a speed of at least $(n-1)/k$, $k \geq 1$, to capture the evader after locating him. The strategy requires no direction/distance detection by the pursuer if $k = 1$, but for $k \geq 2$ it requires distance detection sufficient for the pursuer to know the last vertex occupied by the evader when disappearing from the line of sight. A variant of pursuit-evasion in which all players have "full vision" and thus know the positions of the others at all times has been considered in [19]. Other variants of pursuit-evasion in the grid, in an offline setting and no vision, have been introduced in [7] and revisited in [4, 5, 18]. A survey of other known results on the relation between the pursuers' maximum speed and the possibility of capturing an evader in various graphs is given in [8]. Continuous pursuit-evasion problems have also been discussed in a geometric setting [9, 11, 21].

Various discrete pursuit-evasion games in an arbitrary graph, often called "cops and robbers" and "hunter and rabbit", have been considered in the literature. Some assume that both parties have complete knowledge

[1] A player does not block the view of another.

[2] Sections 3 and 4.1 present some results that require the pursuers to have an exact distance detection capability.

about the opponents' positions at all times (which would be called the "full vision" model in our context) [2, 3, 15, 16], but of particular interest to us are those in which both parties have either no or very limited knowledge about the opponent's positions and moves ("limited vision"), and are in fact a closer match to reality [1, 10]. In particular, it has been shown that under the assumption of no vision, by using a randomized algorithm, a single pursuer can catch an evader with a non-zero (1 over a polynomial in n) probability of success on any connected graph G having n vertices in expected $O(n\log(\mathrm{diam}(G)))$ time, where $\mathrm{diam}(G)$ is the diameter of G [1]. If the players have "one-edge visibility" (i.e., they can see each other and hence have complete knowledge about each other's location, if and only if they occupy adjacent vertices), then two pursuers are always sufficient to catch an evader with a non-zero (1 over a polynomial in n) probability of success in any connected graph having n vertices in expected $O(n^5(\log n)^2)$ time [10]. Obviously, with higher vision capabilities, the pursuers may be able to find the evader more easily, but at the same time, the evader will more easily see the pursuers and avoid a close encounter with them. Indeed, in our pursuit-evasion problem with limited straight-line vision along rows and columns of G_n, it is often crucial for the pursuers to "hide" from the evader to increase their chances of capturing him.

Our results. We first present a randomized algorithm through which one pursuer A with a maximum speed of $s \geq 3$ can capture an arbitrary evader Z in G_n in expected polynomial time (Section 3). The expected capture time is $O(n^{1+\log_{6/5}16}) = O(n^{16.21})$ for $s = 3$, $O(n^{1+\log 12}) = O(n^{4.59})$ for $s = 4$, $O(n^{1+\log 60/13}) = O(n^{3.21})$ for $s = 6$, and it approaches $O(n^3)$ with the further increase of s. Next, we present a randomized algorithm through which two pursuers having a speed of $s = 1 + \varepsilon$ can capture an arbitrary evader in $O(n^3)$ expected time (i.e., we only require the pursuers can move slightly faster than the evader). We also present a three-pursuer deterministic algorithm for capturing an arbitrary evader in G_n in $O(n^2)$ time, using three pursuers with a maximum speed of $s = 1$. No such algorithms were known for capture using one, two or three pursuers having a constant maximum speed limit. In particular, the latter result improves upon the four-searcher algorithm of [20], by using one fewer pursuer.

Number of pursuers	s	Evader	Duration of iterative step	Prob. of capture	Expected time to capture
1 (EDD)	4	arbitrary	$O(n)$	$\frac{1}{n^{\log 12}}$	$O(n^{1+\log 12})$
1 (EDD)	≥ 4	arbitrary	$O(n)$	$\frac{1}{n^{\log 1/p(s)}}$	$O(n^{1+\log 1/p(s)})$
1	$1+\varepsilon$	K-passive	$O(n+K)$	$\frac{2}{5n-4}$	$O(n^2+nK+\frac{1}{\varepsilon})$
1 (EDD)	$2+\varepsilon$	K-passive, $K \leq \frac{n}{2}$	$O(n)$	$\frac{1}{5K}$	$O(nK+\frac{1}{\varepsilon})$
2	$1+\varepsilon$	arbitrary	$O(n^2)$	$\frac{1}{n-1}$	$O(n^3)$
2	1	K-passive	$O(n+K)$	$\frac{4}{9n-6}$	$O(n^2+nK)$
1 (EDD)	1	K-passive, $K \leq \frac{n}{2}$	$O(n)$	$\frac{2}{9K+1}$	$O(n^2)$
3	1	arbitrary	$O(n^2)$	1	$O(n^2)$

Table 1: Summary of the main results. EDD denotes the exact distance detection capability, and s denotes the maximum speed of the pursuer(s). For 2nd row in the table: $p(s) = \frac{s^2-4s+1}{4(s-1)(s-3)}$.

Furthermore, under the additional assumption that the evader is K-*passive* for some known K, that is, he will stop moving after not seeing any pursuer for K time units, the expected capture time can be reduced, with even a smaller maximum speed requirement for the pursuers. Namely, we show that it is possible to capture a K-passive evader using a single pursuer having a maximum speed of $s = 1 + \varepsilon$, in expected $O(n^2 + nK + \frac{1}{\varepsilon})$ time. With two pursuers, we only need $s = 1$ for both, and the expected capture time becomes $O(n^2 + nK)$. If $K \le \frac{n}{2}$, further improvements are possible, provided that the pursuers have the exact distance detection capability. See Table 1 for a summary of these results.

The paper is organized as follows. After some preparatory results in Section 2, we present probabilistic algorithms for one and two pursuers under various assumptions (Sections 3 and 4). In Section 5 we present a three-pursuer deterministic algorithm for capturing an arbitrary evader. Concluding remarks are found in Section 6.

2 Preliminaries

In this section, we present useful schemes for pursuers to locate and possibly capture the evader. We also introduce the concept of K-passiveness that determines the length of period in which an evader can remain active after seeing a pursuer. In the rest of the paper, we denote the pursuers by A, B, $\ldots$ and the evader by Z. For measuring distances in G_n, we use the L_1-shortest path metric, e.g., the distance between $(0, 2/3)$ and $(1, 2/3)$ is $5/3$. $B(o, r)$ denotes the ball of radius r centered at a point o, and $|I|$ stands for the length of an interval I on the line.

2.1 Searching for Z

A first goal in the process of capturing Z is seeing Z.

Lemma 1 *Using a randomized algorithm, one pursuer A with a maximum speed of $s \ge 1$ can locate (see) Z in $O(n)$ expected time.*

Proof. A executes the following procedure to search for Z.

Procedure `Search1`

0. Set $W := 2n$.
1. A goes to $(0, 0)$ from its current location at maximum speed s.
2. Time is reset to 0. A selects a waiting time $w \in [0, W]$ uniformly at random, and waits for time w at $(0, 0)$.
3. Uniformly at random, A selects one of the two axis directions $x+$ or $y+$, and starts moving from $(0, 0)$ in that direction at maximum speed s (towards $(n-1, 0)$ or resp. $(0, n-1)$). If A has not seen Z, repeat the algorithm from step 2 (by symmetry, the step can be iterated by starting at any of the four corners of G_n instead of $(0, 0)$).

Analysis. Assume first that $s > 1$. Consider the time interval $I = [n, 2n] \subset [0, W]$. Let h (resp. v) be the total time in I during which Z is visible along a horizontal line (resp. vertical line) in G_n. Obviously $h + v \ge |I|$ (when Z is at integer grid points, he is visible along both a horizontal and a vertical line). We will show that

$$\text{Prob}(A \text{ sees } Z \text{ in one iteration}) \ge \frac{I}{2W} = \frac{1}{4}.$$

Some intuition: assume that Z is visible along a vertical (resp. horizontal) line at coordinate $x = z$ (resp. $y = z$) at time $t \in I$. First, observe that if A starts Step 3 from $(0, 0)$ at $t_0 = t - z/s$ in the direction $x+$ (resp. $y+$), then A would see Z at t as indicated above. Secondly, observe that A can see Z at most one time during Step 3 (because $s > 1$). The next two claims translate in the fact that disjoint intervals in which Z is seen during the time interval I generate disjoint time intervals in $[0, 2n]$ in which corresponding As can start. They characterize I as a good time-interval included in $[0, W]$ (see also the proofs of Claim 3 and Claim 4 in Section 3).

Claim 1 *If $t \in I$, and $z \in [0, n-1]$, then $t_0 := t - z/s \in [0, 2n] = [0, W]$.*

Proof. Upper bound: Since $t \leq 2n$, we have $t_0 = t - z/s \leq t \leq 2n$. For the lower bound:

$$t_0 = t - \frac{z}{s} = \frac{t-z}{s} + \frac{(s-1)t}{s} \geq 0,$$

where $t \geq z$ and $\frac{s-1}{s} \geq 0$ follow from the maximum unit speed assumption for Z and the maximum speed $s \geq 1$ for A. □

Claim 2 *Suppose A sees Z along a vertical (resp. horizontal) line at moments $t_1 \leq t_2$, $t_1, t_2 \in I$, so that Z is at $x = z_1$ (resp. $y = z_1$) at t_1, and at $x = z_2$ (resp. $y = z_2$) at t_2. Then $t_1 = t_2$ must hold.*

Proof. The corresponding starting time for A in Step 3 would be

$$t_1 - \frac{z_1}{s} = t_2 - \frac{z_2}{s}.$$

This readily implies $z_2 - z_1 = s(t_2 - t_1)$, which leads to a contradiction since $|z_2 - z_1| \leq t_2 - t_1$, by the maximum unit speed assumption for Z, unless $t_1 = t_2$ and $z_1 = z_2$. □

By conditioning on the two possible choices, and by implicitly using the above two claims, we get:

$$\mathrm{Prob}(A \text{ sees } Z \text{ in one iteration}) \geq \frac{1}{2} \cdot \frac{h+v}{W} \geq \frac{1}{2} \cdot \frac{|I|}{W} = \frac{n}{4n} = \frac{1}{4}.$$

Since the probability of success in one iteration of the algorithm is bounded from below by a constant, the expected number of repetitions is also constant, and consequently the entire procedure takes $O(n)$ expected time (one iteration takes n/s time).

The case $s = 1$ requires a slightly more careful argument because Claim 2 does not hold as is; however, the set of exceptions has measure zero in $[0, 2n]$, which then leads to the same bound. This concludes the proof of the lemma. □

2.2 Chasing

We say that pursuer A with the maximum speed $s \geq 1$ *chases* Z if he continuously moves towards Z at a speed of s, after seeing Z within distance at most s. Notice that chasing Z by a pursuer forces Z to continue to move forward to avoid an immediate capture, and — because of the assumption of no direction-detection — A may not know temporarily where Z is, when Z turns left or right at a vertex v during a chase. However, since A is initially within distance s of Z, A will see Z again within distance 1 when he reaches v, thus A will be able to continue to chase.

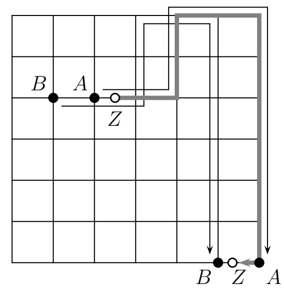

Figure 1: Z will eventually be on the edge between A and B.

Lemma 2

(i) *If $s = 1 + \varepsilon$, then a single pursuer can capture Z within $\frac{s}{\varepsilon} = O(\frac{1}{\varepsilon})$ time after he starts chasing Z.*

(ii) *If $s = 1$, then two pursuers can capture Z within $O(n^2)$ time after one of them starts chasing Z.*

Proof. For the first claim, note that the pursuer will catch up with Z within $\frac{s}{\varepsilon} = O(\frac{1}{\varepsilon})$ time units, even he loses sight of him during the first s time units. For the second claim, suppose pursuer A has started chasing Z. Pursuer B first forms a "tandem" formation with A, say with B to the west of A at distance 1, and thereafter maintains that formation.[3] Once this is accomplished, Z is captured within $O(n^2)$ time units when he attempts to cross an edge westward, with A and B on the east and west endpoints of the edge, respectively, as is shown in Fig. 1. To form a tandem formation, B first catches up with A in $O(n)$ time units (i.e., at a same position) as in the discrete, full visibility case [19], moves with A till they reach a vertex, and then remains stationary for one unit time until A reaches an adjacent vertex. Recall that A and B can communicate with each other without delay throughout this procedure. □

2.3 K-passiveness and guessing

For any integer $K \geq 1$, evader Z is said to be K-*passive* if he can move only for K time units after seeing a pursuer; thereafter such Z becomes stationary, until he sees a pursuer again. A 1-passive evader is analogous to a "reactive rabbit" considered in [10] that can move (in a discrete graph model) only when a hunter is in sight, that is, a hunter is adjacent to a rabbit in the graph (one-edge visibility). An arbitrary evader may be thus considered as ∞-passive.

As expected, K-passiveness for small K can make the job of the pursuers much easier. For instance with a 1-passive evader Z, a single pursuer A with a maximum speed of $s \geq 1$ can start chasing Z in $O(n)$ time steps (a capture then follows by Lemma 2, if either $s > 1$, or $s = 1$ and there are two pursuers): If A does not see Z initially for one unit of time, then Z must be stationary so A first finds Z by moving north along column 0 and east along row $n-1$. As soon as he sees Z, he simply moves towards Z (within the row or column in which they both lie). This forces Z to eventually turn left or right at some vertex v and then come to a halt within distance 1 because of 1-passiveness (or return to the same row/column at a smaller distance from A). Although A does not know at which vertex and in which direction Z has turned (recall that A has no direction detection capability, and has only a limited distance detection capability), by continuing to move in the direction in which he saw Z for the last time, A eventually reaches v, finds Z within distance 1, and starts chasing Z.

[3] We use *east, west, north,* and *south* to refer to the four axis directions in the understood manner.

It sometimes happens that the pursuers have not seen a K-passive Z for K time units, and hence they know that he is stationary. Actually, we often let the pursuers attempt to "hide" from a K-passive Z till he becomes stationary. Once this happens, the pursuers can guess the location of Z, approach and start chasing him[4] with a probability of success of $\Omega(\frac{1}{n})$. Specifically, we have the following lemma.

Lemma 3 *Assume that the pursuer(s) having maximum speed $s \geq 1$, currently located in column 0, know(s) that a K-passive Z is stationary (somewhere out of their sight in G_n). Then*:

(i) *With probability at least $\frac{2}{5n-4}$, a single pursuer can start chasing Z within $O(n)$ time.*

(ii) *With probability at least $\frac{4}{9n-6}$, two pursuers can start chasing Z within $O(n)$ time.*

Proof. Let us prove the first claim. Let $(0, y)$ be the position of pursuer A. Since Z is not visible and hence not in column 0, for the purpose of guessing where Z is, we partition the edges in G_n except those in column 0 into at most $2n - 2 + \lceil \frac{n-1}{2} \rceil$ sections, as described below.

1. The horizontal edges are partitioned into at most $2n-2$ disjoint disconnected sections, where for each i, $0 \leq i \leq n-2$, the edges between columns i and $i+1$ strictly to the north of level y (excluding their west endpoints) form a section, and those between columns i and $i+1$ strictly to the south of level y (excluding their west endpoints) form another. The number of sections is $n-1$ if $y = 0$ or $n-1$, and $2n-2$ otherwise. See Fig. 2(a).

2. The vertical edges in columns 1 through $n-1$ are partitioned into $\lceil \frac{n-1}{2} \rceil$ disjoint disconnected sections, where all edges lying between rows 0 and 2 (excluding their endpoints) form a section, all edges lying between rows 2 and 4 (excluding their endpoints) form another section, and so on. See Fig. 2(b).

Observe that vertices in G_n are part of the sections composed of horizontal edges. It is easy to verify that for every section r, it is possible for A to approach and examine r without being seen by Z stationary in r, so that he will be within distance 1 of Z. See Fig. 2(c-d). Thus the probability that A successfully starts chasing Z is at least $\frac{1}{2n-2+\lceil \frac{n-1}{2} \rceil} \geq \frac{2}{5n-4}$.

The proof of the second claim with two pursuers A and B in column 0 is similar. The horizontal edges are partitioned exactly in the same manner as in the single-pursuer case above, using the location of one of the searchers, say A. The vertical edges not in column 0 are partitioned into $\lceil \frac{n-1}{4} \rceil$ sections by grouping those lying between rows 0 and 4 (excluding their endpoints) into a section, those lying between rows 4 and 8 (excluding their endpoints) into another, and so on. When searching for Z, an approach to a section of horizontal edges is done by A exactly as before. To examine a section r of vertical edges, pursuers A and B approach r in a coordinated manner as illustrated in Fig. 2(e). This shows that the probability that the pursuers correctly guess the section in which Z is located and start chasing him is at least $\frac{1}{2n-2+\lceil \frac{n-1}{4} \rceil} \geq \frac{4}{9n-6}$. □

2.4 Hiding

In many of our randomized algorithms, pursuers use variations of the following general scheme to "hide" in column i, so that they can start chasing Z with probability $\Omega(\frac{1}{n})$ if Z appears in column i later. The general idea is for a pursuer, say A, to choose one of the $n-1$ edges in column i uniformly at random, say edge e, and hide in the interior of e without being seen by Z. If subsequently Z reaches column i at some vertex

[4] For convenience, we say "pursuers start chasing Z" to mean "one of the pursuers starts chasing Z".

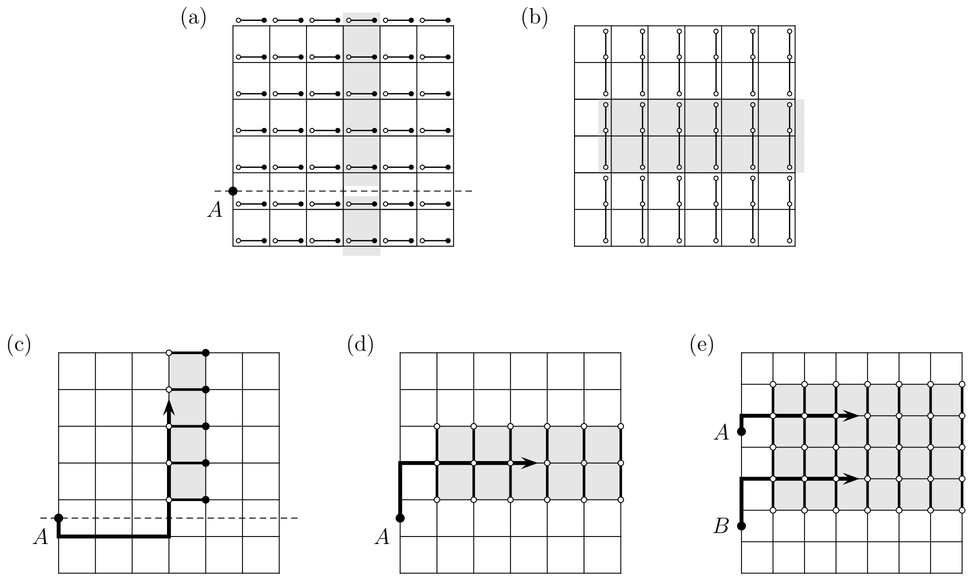

Figure 2: $n = 7$. (a) Partition of the horizontal edges into sections. (b) Partition of the vertical edges into sections. (c) A path of A to approach a section of horizontal edges. (d) A path of A to approach a section of vertical edges. (e) Paths of A and B to approach a section of vertical edges.

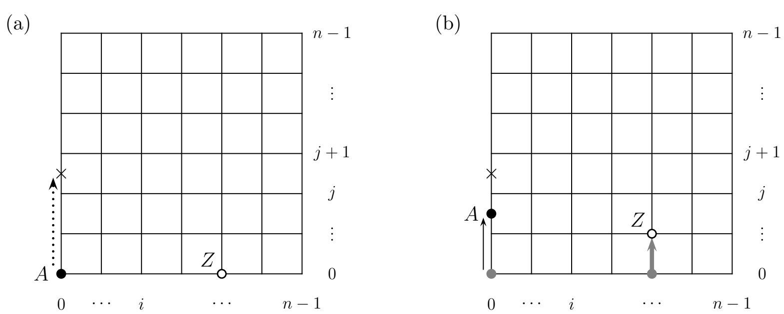

Figure 3: Procedure `Hide1`: (a) A chooses a target position and (b) moves towards it at speed $s > 1$.

v, then with probability at least $\frac{1}{n-1}$, v will be an endpoint of e, and thus A will be within distance 1 of Z, which will allow A to start chasing Z at that moment.

Specifically, assume that a single pursuer A, with maximum speed $s > 1$, is at vertex $(0, 0)$ and sees Z in row 0 at time t at distance > 1. A uses the following procedure `Hide1` to hide in column 0. When it is finished, we say that column 0 is *guarded* by A. See Fig. 3.

Procedure `Hide1`

> A chooses one of the $n - 1$ edges in column 0 uniformly at random, and immediately moves straight to some interior point of that edge at speed s.

Lemma 4 *Suppose that a single pursuer A, with maximum speed $s > 1$, is at vertex $(0, 0)$ and sees Z in row 0 at time t at distance > 1. Let A start procedure* `Hide1` *at time t and hide in column 0. When Z reaches column 0 for the first time after t (if ever), A is within distance 1 of Z with probability at least $\frac{1}{n-1}$.*

Proof. Let e be the edge in column 0 that A chooses. Clearly, if Z does not move from its current position until A reaches an interior point point of e, then Z gains no knowledge about e and hence, when Z reaches column 0 later at some vertex v, with probability at least $\frac{1}{n-1}$, v is an endpoint of e, hence A is within distance 1 of Z [5]. The only possibility for Z to gain some knowledge about e before reaching column 0, is to move quickly to some row $k > 0$ no later than when A would appear there if e was to the north of row k, and see if A indeed appears there (i.e., crosses row k) — if A does, then e is to the north of row k; otherwise e is to the south of row k. However, it is not possible for Z to do this because A moves faster than Z; for every $k > 0$, A can reach $(0, k)$ at time $t + \frac{k}{s} < t + k$, as $s > 1$, while Z cannot reach row k at least until time $t + k$. □

We can adapt procedure `Hide1` for any column and, by interchanging the rows and columns, any row, to guard it by hiding a pursuer in it. As we will see later in Section 4, it is also possible to modify the above scheme so that two pursuers, each with a maximum speed of $s = 1$, hide in column 0 without letting Z know where they are. Note that the direction detection capability of Z does not help him to gain any knowledge about the edge in which A hides in the above scenario, since A always moves north from $(0, 0)$. However, if A, currently at vertex (i, j), $0 \le i \le n - 1$, $1 \le j \le n - 2$, sees Z in row j and wishes to hide in one of the edges in column i using a similar strategy, then Z can see in which direction (north or south) A moves from (i, j) and gain some knowledge about the edge A has chosen. There are two ways to cope with this problem.

`Hide2`: If A has a maximum speed of $s = 2 + \epsilon$, $\epsilon > 0$, then A first moves towards Z at speed s within row j, forcing Z to disappear from row j. The first time A reaches a vertex, say (l, j), with Z not in sight, A moves to one of the edges in column l at speed s.

`Hide3`: If there is another pursuer B (where both pursuers have a maximum speed of $s = 1 + \epsilon$), then A asks B to move towards Z within row j to force Z to leave row j, and hides in column i immediately after Z disappears from row j.

It can be verified by similar arguments as in the proof of Lemma 4, that by using either `Hide2` or `Hide3`, A can reach any row $k \neq j$ before Z can, hence Z gains no knowledge about A's choice. That is, A succeeds in hiding.

3 One-pursuer randomized algorithms

In the first part of this section, we show how to capture an arbitrary evader with one pursuer A in expected polynomial time, provided the maximum speed s of A is about three times the maximum speed of Z (which is 1). We present specific results for $s \ge 3$. The method essentially breaks down for s approaching $s_0 = \frac{3+\sqrt{5}}{2} \approx 2.62$, with the increase of the expected capture time, as this becomes unbounded as a polynomial in n. In this section we assume the players (in particular, A) have the exact distance detection capability. However, in the end, we point out that similar results can be obtained under a weaker approximate distance detection capability for the players. In fact, such results hold even if Z has the exact distance detection capability, while A has only an approximate one.

Theorem 1 *Using a randomized algorithm, one pursuer A with a maximum speed of $s \ge 3$ and the exact distance detection capability can capture an arbitrary Z in G_n in expected polynomial time. More precisely*:

[5] In the scheme presented, the probability is $\frac{1}{n-1}$ if v is either $(0, 0)$ or $(0, n - 1)$, and $\frac{2}{n-1}$ otherwise. It is possible to improve the probability to $\frac{2}{n}$ for even n by letting A exclude every other edge in column 0 when choosing an edge randomly. Similarly, for odd n the probability can be increased to $\frac{2}{n+1}$. We adopted a simpler scheme for clarity of presentation.

(i) *Let* $p(s) = \frac{s^2-4s+1}{4(s-1)(s-3)}$. *For* $s \geq 4$, *the expected capture time is* $O(n^{1+\log 1/p(s)})$.

Specifically, the expected capture time is $O(n^{1+\log 12}) = O(n^{4.59})$ *for* $s = 4$, $O(n^{1+\log 60/13}) = O(n^{3.21})$ *for* $s = 6$, *and it approaches* $O(n^3)$ *with the further increase of* s *(*$s \geq 6$*).*

(ii) *For* $s \in [3.2, 4)$, *the expected capture time*[6] *is* $O(n^{1+\log_{s-2} 4(s-1)})$.

(iii) *For* $s \in [3, 3.2)$, *the expected capture time is* $O(n^{1+\log_{6/5} 16}) = O(n^{16.21})$.

Proof. Let us start with the case $s = 4$ (for simplicity of exposition). The algorithm for A is composed of *phases*. One phase is similar in spirit with the search algorithm described in the proof of Lemma 1. A succeeds in capturing as soon as one phase is successful. Each phase is composed of several (a logarithmic number) of *rounds*. If each round in the current phase is successful, the current phase is declared successful. After each successful round the distance between A and Z is reduced by a constant factor (this distance is measured on the line, since after each successful round, A succeeds in seeing Z). The time-lengths of the rounds decrease exponentially.

Assume A sees Z at some moment at distance d. The first round is successful, if A succeeds in seeing Z again at a smaller distance, say $d' \leq d/2$. The second round is successful, if A succeeds in seeing Z again at a smaller distance, say $d'' \leq d'/2$, and so on, until A gets within distance $\leq s = 4$, after which he will capture Z. We will impose the condition that the distance reduction is bounded from below by a constant fraction larger than 1 (say, 2) after each successful round; in other words $d'/d \leq 1/2$, and so on. We will show that the probability that each round is successful is bounded from below by another constant (say, $1/12$). Putting these together will ensure that A can capture Z in expected polynomial time.

One phase Parameter: $s = 4$.

1. A performs a walk in G_n until he sees Z, using procedure `Search1`. By Lemma 1, this takes $O(n)$ expected time.

2. A repeatedly executes one round (details below) until the current round terminates in failure or the distance between A and Z is less than $s = 4$ (this latter case happens after at most $O(\log n)$ successful rounds, and this makes current phase successful). If the current round terminates in failure, A starts a new phase. If the current phase is successful, A captures Z.

End of phase

One round Parameters: $s = 4$, $x = 2k$, $W = k/4$.

0. Let $d = x = 2k$ be the distance between A and Z at the start of the current round ($t = 0$). W.l.o.g. assume that A and Z are on the same row. Let $W := k/4$. Let $o = (x_0, y_0)$ be the initial position of Z at $t = 0$. (Refer to Fig. 4.)

1. A moves at maximum speed s towards o. As explained below, we can assume that A does not see Z during this time interval, namely $[0, k/2]$. Note that when A reaches o at $t = k/2$, Z is somewhere in the ball $B(o, k/2)$.

2. A selects a waiting time $w \in [0, W]$ uniformly at random, and waits for time w at o.

[6] This bound appeared incorrectly in the extended abstract.

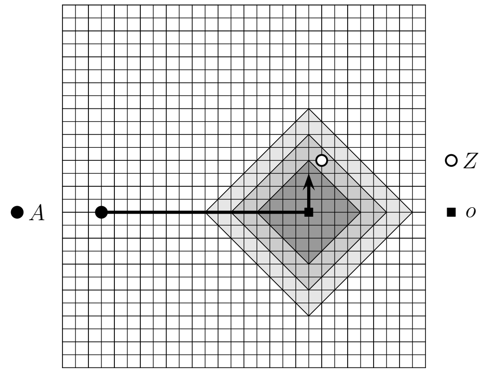

Figure 4: One round of A's algorithm for $s = 4$; here $k = 8$, $x = 2k = 16$, $W = k/4 = 2$. The three concentric balls around o, $B(o, k/2)$, $B(o, 3k/4)$ and $B(o, k)$, are shown.

3. A selects one of the four axis directions $x+$, $x-$, $y+$, $y-$, and starts moving from o in that direction at maximum speed s. If A hits the boundary of G_n (when $B(o, k)$ is not entirely contained in G_n), it stays there until time k. Note that A reaches the boundary of the ball $B(o, k)$ or the boundary of G_n latest at time $t = k$ (by our choice of parameters); while Z is also confined to the same ball $B(o, k)$ in the time interval $[0, k]$. If A sees Z during this last segment of his move, the current round is successful, and otherwise it is not.

End of round

We first observe that in a successful round, the distance reduction is at least 2: $\frac{d'}{d} \le \frac{k}{d} = \frac{k}{2k} = \frac{1}{2}$, as A and Z are within distance k when A sees Z in $B(o, k)$. For one round: let o be the *last* position of Z when A sees Z; w.l.o.g. o is a grid point of G_n, because once A sees Z (say, along a row), A will force Z out of this row, and this happens at an integer grid point of G_n. The time is reset to 0 when o is chosen. Consider the ball $B(o, k)$ of radius k centered at o (the position of Z at time $t = 0$ when A sees Z). If it happens that A sees Z before reaching o, the time is reset, a new origin is chosen, and the current round (not a new one) is restarted from the smaller current distance between A and Z.

Consider the time interval $I = [2k/3, 3k/4]$. Note that $I \subset [k/2, k/2 + W]$. Let h^+ (resp. h^-) be the total time in I during which Z is visible along a horizontal line (row) with y-coordinate $\ge y_0$ (resp. $< y_0$). Similarly, let v^+ (resp. v^-) be the total time in I during which Z is visible along a vertical line (column) with x-coordinate $\ge x_0$ (resp. $< x_0$). Obviously, $h^+ + h^- + v^+ + v^- \ge |I|$ (when Z is at integer grid points, he is visible along *both* a horizontal and a vertical line). We will say that the current round is *successful* if A sees Z in the time interval $[k/2, k]$.

Assume that Z is visible along a vertical (resp. horizontal) line at coordinate $x = x_0 + z$ (resp. $y = y_0 + z$) at time $t \in I$. W.l.o.g. we can assume that $z \ge 0$. Note that if A starts its last segment (Step 3 of the round) from o at $t_0 = t - z/4$ in the direction $x+$ (resp. $y+$), then A would see Z at t as indicated above. We also need to argue that A can see Z at most one time during his last segment of move; to put it differently, if Z is visible by A along the same direction (vertical or horizontal) at two different moments $t_1 < t_2$, $t_1, t_2 \in I$, the corresponding As must have different starting times. This translates in the fact that disjoint intervals in which Z is seen during the time interval I generate disjoint time intervals in $[k/2, k/2 + W]$ in which corresponding As can start.

Specifically we prove the following two claims regarding the interval I we have chosen. In general, we say that $I \subset [k/2, k/2 + W]$ is a *good interval* if it satisfies these two claims.

Claim 3 *If* $t \in I = [2k/3, 3k/4]$, *then* $t_0 := t - z/4 \in [k/2, 3k/4] = [k/2, k/2 + W]$.

Proof. Upper bound: Since $t \leq 3k/4$, we have $t_0 = t - z/4 \leq t \leq 3k/4$. For the lower bound:

$$t_0 = t - z/4 = \frac{t-z}{4} + \frac{3t}{4} \geq \frac{3}{4} \cdot \frac{2k}{3} = \frac{k}{2},$$

where $z \leq t$ follows from the maximum unit speed assumption for Z. □

Claim 4 *Suppose* A *sees* Z *along a vertical* (*resp. horizontal*) *line at moments* $t_1 \leq t_2$, $t_1, t_2 \in I$, *so that* Z *is at* $x = x_0 + z_1$ (*resp.* $y = y_0 + z_1$) *at* t_1, *and* Z *is at* $x = x_0 + z_2$ (*resp.* $y = y_0 + z_2$) *at* t_2. *Then* $t_1 = t_2$.

Proof. Similarly as in the proof of Lemma 1, the corresponding starting time for A in his last segment of move would be

$$t_1 - \frac{z_1}{4} = t_2 - \frac{z_2}{4}.$$

This readily implies $z_2 - z_1 = 4(t_2 - t_1)$, which leads to a contradiction since $|z_2 - z_1| \leq t_2 - t_1$, by the maximum unit speed assumption for Z, unless $t_1 = t_2$ and $z_1 = z_2$. □

We now show that the probability that any given round is successful is at least $1/12$. By conditioning on the four possible choices (axis directions) followed by A in his last segment, and by implicitly using Claim 3 and Claim 4, we get:

$$\text{Prob(success in one round)} \geq \frac{\frac{1}{4}h^+ + \frac{1}{4}h^- + \frac{1}{4}v^+ + \frac{1}{4}v^-}{W} \geq \frac{|I|/4}{W} = \frac{|I|/4}{k/4} = \frac{|I|}{k} = \frac{1}{12}.$$

Observe that the bound remains valid even if A hits the boundary of G_n during the last segment of his move (Step 3), since we can imagine that A continues his move beyond this boundary.

Since the initial distance between A and Z (after Step 1 of a phase) is $x \leq n$, after at most $\log n$ successful rounds, the distance between A and Z becomes less than 4 (recall, the distance reduction after a successful round is at least 2 : ($d'/d \leq 1/2$). The current phase is then successful since A can chase and then capture Z within another $4/3$ time by Lemma 2.

$$\text{Prob(success in one phase)} \geq \left(\frac{1}{12}\right)^{\log n} = \frac{1}{n^{\log 12}}.$$

The execution time for one phase is bounded by

$$O(n) + \frac{1}{2}\left(n + \frac{n}{2} + \frac{n}{4} + \dots\right) = O(n).$$

It follows that the expected number of phases until a successful one occurs is $O(n^{\log 12})$, and the expected capture time is consequently $O(n^{1+\log 12}) = O(n^{4.59})$, as claimed. This concludes the analysis for the case $s = 4$.

We now show how to extend this result for (constant) $s \geq 4$, and set the parameters, so as to decrease the expected capture time. The algorithm for A is otherwise the same. For simplicity, we require the distance reduction in a successful round to be at least 2, as in case $s = 4$. Let $W = ak$ for some $a \in (0, 1)$ satisfy the equation

$$\frac{2k}{s} + W + \frac{k}{s} = k.$$

This ensures that A reaches the boundary of $B(o,k)$ by the time $t = k$, while Z is confined to the ball $B(o,k)$ on the time interval $[0,k]$. The above can be rewritten as

$$\frac{3}{s} + a = 1, \tag{1}$$

which yields $a = 1 - \frac{3}{s}$. We now determine a *good* interval $I \subset [\frac{2k}{s}, \frac{2k}{s} + W]$. That is, I is an interval for which analogous claims to Claim 3 and Claim 4 are satisfied. Set $I = [\frac{2k}{s-1}, (\frac{2}{s} + a)k] = [\frac{2}{s-1}k, \frac{s-1}{s}k]$. Indeed, on one hand, we verify that if $t \in I$, then $t_0 = t - \frac{z}{s} \geq \frac{2k}{s}$. Recall that at time $2k/s$, A has reached o. We have

$$t_0 = t - \frac{z}{s} = \frac{t-z}{s} + \left(1 - \frac{1}{s}\right) t \geq 0 + \frac{s-1}{s} \cdot \frac{2k}{s-1} = \frac{2k}{s},$$

as in Claim 4. On the other hand, if $t \in I$, then

$$t - \frac{z}{s} \leq \frac{2k}{s} + ak,$$

as in Claim 3. Thus I is a good interval. Then

$$\text{Prob(success in one round)} \geq \frac{|I|}{4W} = \frac{\frac{s-1}{s} - \frac{2}{s-1}}{4(1 - \frac{3}{s})} = \frac{s^2 - 4s + 1}{4(s-1)(s-3)} := p(s).$$

Observe that our previous case $s = 4$ is only a special case of this setting. Indeed, for $s = 4$, we get

$$\text{Prob(success in one round)} \geq p(4) = \frac{1}{4 \cdot 3} = \frac{1}{12}.$$

For instance, if $s = 6$, we get

$$\text{Prob(success in one round)} \geq p(6) = \frac{1}{4} \cdot \frac{13}{15} = \frac{13}{60},$$

and this results in an expected capture time of $O(n^{1+\log 60/13}) = O(n^{3.21})$. It is also easy to observe that the expected capture time is a decreasing function of s, therefore it is bounded by $O(n^{3.21})$ for $s \geq 6$, and it approaches $O(n^3)$ with the increase of s. Indeed,

$$\lim_{s\to\infty} p(s) = \lim_{s\to\infty} \frac{s^2 - 4s + 1}{4(s-1)(s-3)} = \frac{1}{4}.$$

The general upper bound on the expected capture time we obtain by this method for a given $s \geq 4$ is $O(n^{1+\log 1/p(s)})$, with $p(s)$ defined previously. Observe also that this bound is inapplicable for instance when s is close to 3. It can be checked that $|I| > 0$ requires

$$a > 2\left(\frac{1}{s-1} - \frac{1}{s}\right) = \frac{2}{(s-1)s}.$$

Using $a = 1 - 3/s$, yields $s^2 - 4s + 1 > 0$, and further $s > 2 + \sqrt{3} \approx 3.73$.

Now, we show how to extend our result for the interval $s \in [3,4)$. Our method can be further pushed for values $s < 3$ (provided $s > s_0 = \frac{3+\sqrt{5}}{2} \approx 2.62$), however, the bound on the expected capture time becomes prohibitive already for $s = 3$ (see below).

Instead of 2 we now only require a (modest) distance reduction of $x > 1$ per round (for x close to 1 in the limit). This means we slightly change notation and denote the initial distance between A and Z by xk (for $x > 1$). The number of necessary successful rounds remains $O(\log n)$, and furthermore, the execution time for one phase remains linear, since for $x > 1$, we have $\sum_{i \geq 0} nx^{-i} = O(n)$. Equation (1) is now

$$\frac{x}{s} + a + \frac{1}{s} = 1, \tag{2}$$

which yields $a = 1 - \frac{x+1}{s}$. For this setting, it can be checked that $I = [\frac{x}{s-1}k, \frac{s-1}{s}k]$ is a good interval. (Observe that $I \subset [\frac{x}{s}k, \frac{x}{s}k + W]$.) The condition $|I| > 0$ requires $(s-1)^2 - sx > 0$. We observe in passing the limit of this method, when setting $x = 1$: the quadratic equation $(s-1)^2 - s = 0$ has solutions $s_{1,2} = \frac{3 \pm \sqrt{5}}{2}$, and the larger of the two (which is relevant here) imposes $s > \frac{3+\sqrt{5}}{2} \approx 2.62$.

Given $s \in [3, 4)$, we set

$$x(s) = \begin{cases} 6/5 & \text{if } s \in [3, 3.2) \\ s - 2 & \text{if } s \in [3.2, 4) \end{cases}.$$

We have

$$\text{Prob(success in one round)} \geq \frac{|I|}{4W} = \frac{1}{4a}\left(\frac{s-1}{s} - \frac{x}{s-1}\right) = \frac{(s-1)^2 - sx}{4(s-1)(s-(x+1))}.$$

For $s \in [3, 3.2)$ and $x(s) = 6/5$, this is bounded from below as follows:

$$\text{Prob(success in one round)} \geq \frac{(s-1)^2 - 1.2s}{4(s-1)(s-2.2)} = \frac{1}{4}\left(1 - \frac{1.2}{(s-1)(s-2.2)}\right) \geq \frac{1}{4} \cdot \left(1 - \frac{3}{4}\right) = \frac{1}{16}.$$

The corresponding distance reduction is $x(s) = 6/5$, which results in an expected capture time of $O(n^{1+\log_{6/5} 16}) = O(n^{16.21})$. (This bound is attained for $s = 3$, while for larger s in this interval, the expected capture time is smaller.)

For $s \in [3.2, 4)$ and $x(s) = s - 2$, this is bounded from below as follows:

$$\text{Prob(success in one round)} \geq \frac{(s-1)^2 - (s-2)s}{4(s-1)} = \frac{1}{4(s-1)}.$$

The corresponding distance reduction is $x(s) = s - 2$, which results in a reduced expected capture time of $O(n^{1+\log_{s-2} 4(s-1)})$. □

Remark. The following assumption of an approximate distance detection capability for the players is a natural one. If the players (A and Z) are visible to each other at some distance d, the distance $\widetilde{d}$ observed by some player satisfies

$$1 - \rho \leq \frac{\widetilde{d}}{d} \leq 1 + \rho,$$

for a small constant ρ (e.g., $\rho = 1/10$ or $\rho = 1/100$). Of course, the distances observed by A and Z may differ. The same algorithm of A for capturing Z can be used, so that similar results hold under this assumption of approximate distance detection capability as well. Essentially, A proceeds according to the estimated distance perceived, with the effect that the the probability of success per round and the distance reduction per round are slightly reduced. We omit the calculations.

3.1 One-pursuer randomized algorithms for a K-passive Z

Recall that an evader is called K-*passive*, if he will stop moving after not seeing any pursuer for K time units. First, we show that for any $K \geq 1$, a single pursuer A with a limited distance detection capability can capture a K-passive Z in G_n in expected time $O(n^2 + nK + \frac{1}{\varepsilon})$, provided he can move slightly faster than Z, i.e., $s = 1 + \varepsilon$ for an arbitrary small $\varepsilon > 0$. We assume that A knows the value of K. The algorithm we give here is considerably simpler than the algorithms we present for two pursuers for capturing Z under weaker assumptions (in Section 4).

The algorithm works as follows. A goes to $(0, 0)$ and waits for Z to appear in row 0 or column 0 for up to K time units. If Z appears in row 0 within K time units (w.l.o.g. at distance greater than 1; otherwise A immediately starts chasing Z), then A immediately hides in column 0 using procedure `Hide1`. Once this is done, if Z appears in column 0 within the next K time units, then A will be within distance 1 of Z with probability at least $\frac{1}{n-1}$ (by Lemma 4), and then can start chasing Z if this happens. On the other hand, if Z does not appear in column 0 for K time units, either while A waits at $(0, 0)$ in the beginning or after A hides in column 0, then A knows K is stationary. Then A guesses Z's location, approaches him, and starts chasing him if he is indeed there. The probability of success is at least $\frac{2}{5n-4}$ by Lemma 3. The total time so far is $O(n + K)$, and if A successfully starts chasing Z, then a capture occurs in additional $O(\frac{1}{\varepsilon})$ time units by Lemma 2. The case in which Z appears in row 0 is handled in a similar manner. To summarize:

Theorem 2 *Using a randomized algorithm* (*described above*)*, with probability at least* $\frac{2}{5n-4}$*, a single pursuer with maximum speed of* $s = 1+\varepsilon$ *can start chasing a* K*-passive* Z *within* $O(n+K)$ *time. The expected time to capture* Z *by repeating this process is* $O(n^2 + nK + \frac{1}{\varepsilon})$.

In Theorem 2 we only assumed that the pursuer has a limited distance detection capability that allows him to know only whether or not Z in sight is within distance 1 of his location. However, if the pursuer has the exact distance detection capability, then for the case $K \leq \frac{n}{2}$, it is possible to improve the probability for capturing a K-passive evader from $\Omega(\frac{1}{n})$ to $\Omega(\frac{1}{K})$, by increasing the maximum speed of the pursuer to $s = 2 + \epsilon$. The improvement is based upon the following observation: if K is small, then it is to the pursuer's advantage to find Z first and *then* hide, to improve the chances of guessing Z's final location.

Theorem 3 *With probability at least* $\frac{1}{5K}$*, a single pursuer with maximum speed of* $s = 2 + \varepsilon$*,* $\varepsilon > 0$*, and the exact distance detection capability can start chasing a* K*-passive* Z*,* $K \leq \frac{n}{2}$*, within* $O(n)$ *time. The expected time to capture* Z *is* $O(nK + \frac{1}{\varepsilon})$.

Proof. We improve the chances of guessing Z's final location from $\Omega(\frac{1}{n})$ to $\Omega(\frac{1}{K})$, by modifying the hiding strategy for A, described before Theorem 2, in the following manner, so that A first finds Z and then hides.

STEP 1. A goes to $(0, 0)$ and waits for Z to appear in row 0 or column 0 for up to K time units. During the K time units:

a. If Z appears within distance $2 + \varepsilon$ from A, then A immediately starts chasing Z.

b. If Z appears at a distance greater than $2 + \varepsilon$ from A, then A immediately starts STEP 2.

c. If Z does not appear in row 0 nor column 0 (this means Z is stationary), then A first locates Z by moving north along column 0 and east along row $n - 1$; observe that A is at a vertex at the moment he finds (sees) Z. If Z is within distance $2 + \varepsilon$ from A, then A starts chasing Z, else A continues with STEP 2.

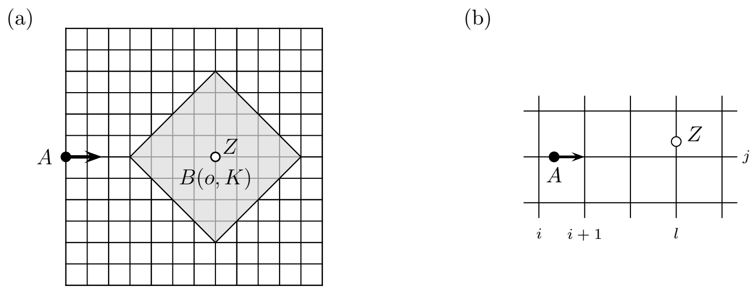

Figure 5: (a) When A sees Z, he moves towards Z until either the distance d between the two is $s = 2 + \varepsilon$ — and then A starts chasing — or Z disappears from row j. (b) Z disappeared from row j at vertex (l, j), and A has not seen Z on his way from point $p = (i + r, j)$, $r \in [0, 1)$ towards vertex $(i + 1, j)$.

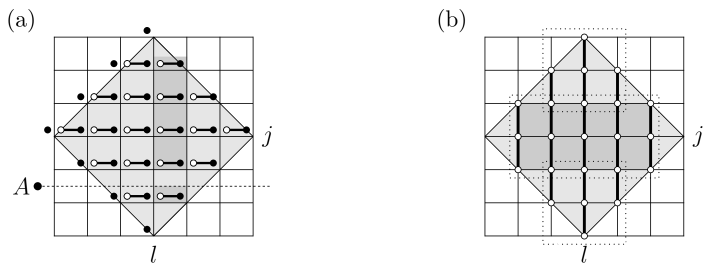

Figure 6: Partition of the edges that intersect $B((l, j), K)$ into at most $5K$ sections; here we have $K = 3$. (a) Partition of the horizontal edges and points into at $1 + 1 + 2 + 2 + 2 + 1 + 1 = 10 \leq 4 \cdot 3$ sections. (b) Partition of the vertical edges into 3 sections.

STEP 2. W.l.o.g. assume that A sees Z in row j, and thus A is located at $(0, j)$. (Notice that $(0, j)$ also includes $(0, 0)$, and the case when A sees Z in column i is handled in a similar manner.) Now, to succeed in hiding, A uses a modification of `Hide2`: with speed $2 + \varepsilon$, A moves towards Z on row j until either the distance d between the two is $s = 2+\varepsilon$ — and then A starts chasing — or Z disappears from row j (which must happen to avoid chasing); see Fig. 5(a). Let $p = (i + r, j)$, $r \in [0, 1)$, be the first point such that (i) Z disappears from row j at vertex (l, j), $i + 1 < l$, when A is at p, and (ii) Z does not reenter row j until A reaches vertex $(i + 1, j)$ while moving towards Z; see Fig. 5(b). When A reaches $(i + 1, j)$, he memorizes the distance $d := l - i$, and hides in column $i + 1$ using the following strategy.

- If $d > K$, then A hides in column $i + 1$ at a location immediately to the north or south of vertex $v = (i + 1, j)$. Since v is the only vertex in column $i + 1$ that Z could possibly reach in the next K time units, A can start chasing Z if Z appears in column $i + 1$.
- If $1 < d \leq K$, then A hides in column $i + 1$ with the following modification. Observe that in the next K time units, Z can reach column $i + 1$ only at those vertices within distance $K - d + 1$ of $(i + 1, j)$. Thus A randomly selects an edge in column $i + 1$ from among those edges both of whose endpoints are within distance $K - d + 1$ of v, and then moves to the midpoint of that edge. Since there are $2(K - d + 1) < 2K$ such edges, A can start chasing Z with probability at least $\frac{1}{2K}$ if Z appears in column $i + 1$.

It easy to see that A succeeds in hiding by similar arguments as in the proof of Lemma 4.

So w.l.o.g. assume now that A is hiding in column $i + 1$ at $(i + 1, y)$, $j - K < y < j + K$, in the interior of some edge e. Note that a K-passive Z, last seen at vertex (l, j) by a pursuer, must be stationary at some point inside the ball $B((l, j), K)$ after K time units. For the purpose of guessing, as in the proof of Lemma 3, we partition the (at most $4K^2 + 4K - 1$) edges of G_n that intersect $B((l, j), K)$ into at most $5K$ disjoint disconnected sections as follows. (See Fig. 6(a-b).)

a) The horizontal segments of $B((l, j), K)$ are first divided into horizontal edges (excluding their west endpoints) and points — see Fig. 6(a). Next, for each $l - K \leq i \leq l + K$, the edges/points between columns i and $i+1$ strictly to the north of level y form a section, and those between columns i and $i+1$ strictly to the south of level y form another. The number of sections is at most $2 + 2 \cdot (2K - 1) = 4K$. (Fig. 6(a))

b) The vertical segments of $B((l, j), K)$ are first divided into vertical edges (excluding their endpoints). Next, these vertical edges are partitioned into K disjoint disconnected sections, where all edges lying between rows $j - K$ and $j - K + 2$ (excluding their endpoints) form a section, all edges lying between rows $j - K + 2$ and $j - K + 4$ (excluding their endpoints) form another section, and so on — see Fig. 6(b).

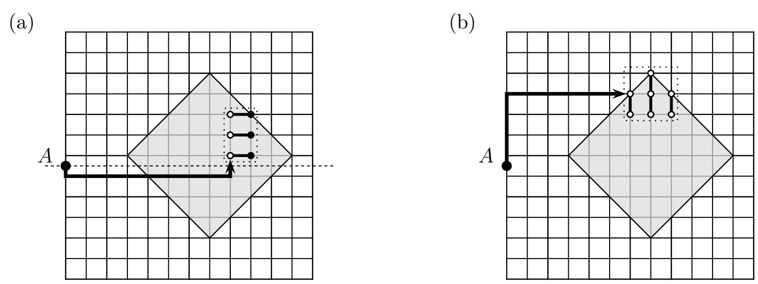

Figure 7: (a) A path of A from his hiding place to approach a section of horizontal edges. (b) A path of A to approach a section of vertical edges.

Now, if Z does not appear in the column/row that A guards in the next K time units after hiding, then A randomly chooses a section in the partition of ball $B((l, j), K)$ described above, approaches it along a suitable path (see Fig. 7), and starts chasing Z if he is there, with probability at least $\frac{1}{5K}$. Otherwise, if Z crosses the guarded column/row, then A starts chasing with probability at least $\frac{1}{2K}$. $\square$

Observe that if we assume no direction detection capability for Z, then there is no need for A to first force Z to disappear from the row/column they are in, as A can immediately hide in a column/row without Z knowing which direction A goes. Therefore, with no direction detection capability for Z, we can reduce A's maximum speed down to $1 + \varepsilon$, and the above approach continues to work. Consequently, we get

Corollary 1 *Assume that Z has no direction detection capability. Then, with probability at least $\frac{1}{5K}$, a single pursuer with maximum speed of $s = 1 + \varepsilon$, $\varepsilon > 0$, and the exact distance detection capability can start chasing a K-passive Z, $K \leq \frac{n}{2}$, within $O(n)$ time. The expected time to capture Z is $O(nK + \frac{1}{\varepsilon})$.*

4 Two-pursuer randomized algorithms

The one-pursuer algorithms presented in the previous section require either the pursuer's maximum speed about 3 (recall from Section 3 that $s_0 = \frac{3+\sqrt{5}}{2} \approx 2.62$), or two assumptions: (1) the evader is K-passive

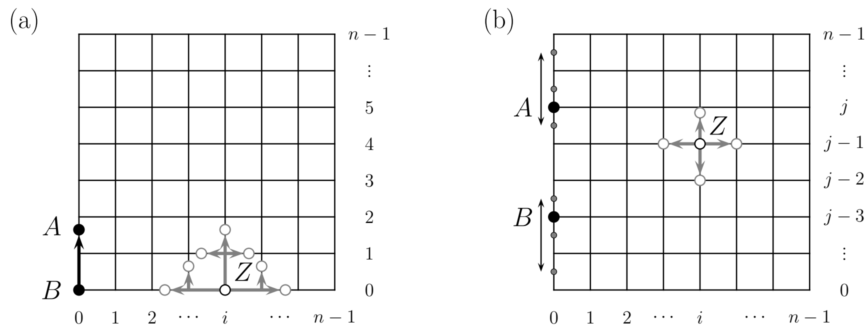

Figure 8: Procedure `Hide4`: (a) Movement of A and B in Step 2. (b) Possible positions of A and B chosen in Step 3.

for some K, and (2) the pursuer has the maximum speed of $s = 1 + \varepsilon$. In this section, we show that it is possible to capture the evader in G_n in polynomial time even if we drop one of the two assumptions (1)-(2), provided that we have two pursuers.

4.1 Capturing a K-passive evader

First, we consider the case in which evader Z is K-passive for some K known to the pursuers A and B whose maximum speed is $s = 1$. The basic idea is similar to that in the one-pursuer algorithms discussed in the previous section. Pursuers A and B attempt to hide from Z in column 0 (or row 0), in such a way that either (1) Z does not see them for K time units and becomes stationary, or (2) within $O(n + K)$ time units Z encounters A or B within distance 1 with probability $\Omega(\frac{1}{n})$. In the latter case one of the pursuers starts chasing Z and the capture follows within $O(n^2)$ time by Lemma 2. The difficulty here is that the pursuers with a maximum speed of $s = 1$ cannot move faster than Z, and hence they cannot use procedure `Hide1` to hide. However, we overcome this issue by hiding *both* A and B in column/row 0 without letting Z know their exact locations. Specifically, the pursuers execute the following procedure `Hide4` to hide in column 0.

Procedure `Hide4`

1. A and B go to $(0, 0)$ and stay there until they see Z in row 0 or column 0. If they do not see Z for K time units, then Z is stationary, and hence they guess the location of Z in G_n, approach him, and start chasing him with probability at least $\frac{4}{9n-6}$ by Lemma 3. So suppose they see Z in row 0. (The case in which they see Z in column 0 is handled in a symmetric manner, with rows and columns interchanged.) If Z is within distance 1, then they immediately start chasing Z. Otherwise, go to Step 2.

2. A immediately leaves $(0, 0)$, moves north in column 0 at a speed of 1, and stops as soon as it reaches the first vertex $(0, j)$, $1 \leq j \leq n - 2$, such that A does not see Z in row j; otherwise, if A sees Z in row j for every j, $1 \leq j \leq n - 2$, then it stops at $(0, n - 1)$. Concurrently, B stays at $(0, 0)$ until A reaches $(0, 3)$, and thereafter moves north at a speed of 1 and maintains the distance of 3 from A. B stops as soon as A stops. (If A stops at $(0, 1)$, $(0, 2)$ or $(0, 3)$, then B is still at $(0, 0)$.) See Fig. 8(a).

3. See Fig. 8(b). Suppose that A and B are at vertices $(0, j)$ and $(0, j')$, respectively, where $j' = 0$ if $j \leq 3$, and $j' = j - 3$ if $4 \leq j \leq n - 1$. A chooses one of the edges in column 0 to the north of row $j - 1$ uniformly at random, and moves straight to the midpoint of that edge at a speed of 1. Concurrently, B chooses one of the edges in column 0 to the south of row $j' + 1$ uniformly at random,

and moves straight to the midpoint of that edge at a speed of 1. (As is shown in the proof of Lemma 5, by maintaining a distance of 3, A and B can together "guard" column 0.)

4. A and B stay at their respective locations in column 0 until one of the following situations occurs.

   (a) Z appears in column 0 at some vertex at distance greater than 1 from A and B (this can occur before A and B arrive at their respective locations chosen in the previous step): in this case `Hide4` ends in failure.

   (b) Z appears in column 0 at some vertex within distance 1 from A or B: in this case the pursuers start chasing Z.

   (c) K time units have elapsed since A or B saw Z for the last time: since Z is stationary, the pursuers guess the location of Z, approach him, and start chasing him with probability at least $\frac{4}{9n-6}$ by Lemma 3.

Lemma 5 *Using* `Hide4`*, with probability at least* $\frac{4}{9n-6}$*, two pursuers* A *and* B *with a maximum speed of* $s = 1$ *can start chasing a* K*-passive* Z *in* $O(n + K)$ *time.*

Proof. In Section 2, we already showed that the pursuers successfully start chasing Z with probability at least $\frac{4}{9n-6}$ once Z becomes stationary. Suppose Z does not become stationary, and assume that Step 3 starts at time t with A and B at vertices $(0, j)$ and $(0, j')$, respectively.

Let us consider the case in which $4 \leq j \leq n-2$ and $j' = j-3$. By the minimality of j, A saw Z in rows $1, 2, \ldots, j-1$ while moving north at a speed of 1 but he did not in row j. Thus it must be the case that Z also moved north at a speed of 1 along some column $i \geq 2$ simultaneously with A, at least until he reached vertex $(i, j-1)$ at time $t-1$, but he did not reach vertex (i, j) at time t. (The case $i = 1$ cannot happen since that would imply Z was within distance of 1 of the pursuers in row 0.) Hence at time t, Z must be strictly below row j and strictly above row j' (indeed, Z is on or above row $j-2 = j'+1$). Furthermore, since $i \geq 2$, Z is at distance at least 1 from column 0 at time t. Therefore, while in Step 3, A and B are moving respectively towards the midpoints of the edges they have chosen, Z cannot gain any knowledge about A and B's destinations unless he first moves to column 0. However, before Z reaches column 0 at some vertex $v = (0, y)$ necessarily at time $t+1$ or later, A or B have reached either $(0, y - \frac{1}{2})$ or $(0, y + \frac{1}{2})$ (i.e., the midpoints of the edges incident on v) with probability at least $\frac{1}{n-2}$. (Note that every edge in column 0 except the one between vertices $(0, j-2)$ and $(0, j-1)$ can be chosen by A or B with probability at least $\frac{1}{n-2}$.) Thus with probability at least $\frac{1}{n-2}$, A or B will be within distance 1 of Z when Z reaches column 0 and then starts chasing Z.

An argument similar to the above can be used to show that, if $j \leq 3$ or $j = n-1$, then chasing starts with probability at least $\frac{1}{n-1}$ when Z reaches column 0.[7] We omit the details, and only mention that in both of these cases, although Z gains knowledge about the position of one pursuer — if $j = n-1$ then A always hides at $(0, n - \frac{3}{2})$, and if $j \leq 3$ then B always hides at $(0, \frac{1}{2})$ — Z's uncertainty about the position of the other pursuer ensures that column 0 is guarded (by both pursuers).

Therefore, in any execution of `Hide4`, either (1) Z becomes stationary and one of the pursuers starts chasing him with probability at least $\frac{4}{9n-6}$, or (2) Z does not become stationary and one of the pursuers starts chasing him with probability at least $\frac{1}{n-1}$. Finally, it is straightforward to verify that `Hide4` does not run for more than $O(n + K)$ time. □

[7] It is possible to improve the probability by excluding (more or less) every other edge from consideration in Step 3. For instance, if $n = 8$, $j = 5$ and $j' = 2$, then we can let each of A and B choose an edge out of two and obtain a probability of $\frac{1}{2}$. However, these improvements do not affect the overall success probability of `Hide4`.

Theorem 4 *With probability at least $\frac{4}{9n-6}$, two pursuers with a maximum speed of $s = 1$ can start chasing a K-passive Z within $O(n+K)$ time. The expected time to capture Z by repeating the process is $O(n^2+nK)$.*

Proof. The result follows from Lemma 5 and the observation that, by Lemma 2, a capture occurs within $O(n^2)$ time once chasing starts. □

Analogously, as was done for the one-pursuer algorithm, by allowing the exact distance detection capability for pursuers, we can improve the probability of chasing for $K \leq \frac{n}{2}$ in the two-pursuer algorithm, still keeping the maximum speed equal to 1.

Theorem 5 *With probability at least $\frac{2}{9K+1}$, two pursuers with a maximum speed of $s = 1$ and the exact distance detection capability can start chasing a K-passive Z, $K \leq \frac{n}{2}$, within $O(n)$ time. The expected time to capture Z by repeating the process is $O(n^2)$.*

Proof. As the proof is similar to the one-pursuer case, we only present a sketch. Again, the idea is to let the pursuers locate Z before hiding and memorize his location $o = (x, y)$, so that if Z becomes stationary, then they can start chasing him with a probability of $\Omega(\frac{1}{K})$ by considering only those edges that intersect $B(o, K)$ when guessing the location of Z. As before, for the purpose of guessing we partition such edges into a total of at most $\frac{9K+1}{2}$ disjoint sections following the idea in the proof of Lemma 3. With two pursuers available, we have at most $4K$ sections of horizontal edges and $\lceil \frac{K}{2} \rceil$ sections of vertical edges. Locating a stationary Z is straightforward with A and B moving together along column 0 and row $n-1$. However, since their maximum speed is $s = 1$, when they see Z (from a vertex, of course) they hide using a scheme that is a variation of Steps 2 and 3 of `Hide4`. We give only an outline — For instance, to hide in column 0 (when they see Z along some row), A and B move in opposite directions in column 0 until they are distance 3 apart, and thereafter maintain that distance until either (1) neither pursuer sees Z (along a row) or (2) one hits the grid boundary. Then they each randomly select an edge appropriately and move to its midpoint. The reader can verify that it is possible to do this in such a way that Z gains no knowledge about the pursuers' target positions. We omit the details. □

4.2 Capturing an arbitrary evader

In this section, we show that for an arbitrary small $\varepsilon > 0$, two pursuers A and B with a maximum speed of $s = 1 + \varepsilon$ can capture an arbitrary Z with probability at least $\frac{1}{n-1}$. (In the discussion below we assume $\varepsilon < 1$, however, our approach can be easily modified for arbitrary $\varepsilon > 0$.) The overall strategy of the pursuers is to first guard a column or row, and then "advance" it towards Z until Z is forced to appear in a guarded column/row. At that moment, with probability at least $\frac{1}{n-1}$, the pursuers guarding the column/row are within distance from Z small enough to start chasing.

First, the pursuers guard a column or row. Note that they cannot use procedure `Hide3` until one of them sees Z from a position near a vertex. So the approach in which a pursuer attempts to first locate Z and then hide using `Hide3`, does not always work (unless the pursuer has a greater maximum speed of $s > \frac{3}{2}$, as the reader can verify), because Z may become visible to him only when he is not near any vertex. However, taking advantage of having now two pursuers A and B, this difficulty can be overcome as follows. Pursuers A and B move to $(0, 0)$. While A stays at $(0, 0)$, B searches for Z using the randomized procedure `Search1`. It is easy to check that either A or B will see Z from a vertex in $O(n)$ expected time: if Z appears in column or row 0, A sees him from vertex $(0, 0)$; otherwise, if Z does not appear, B sees him from the vertex he is at on his path on column 0 or row 0. Once this happens, the relevant pursuer successfully hides using `Hide3`.

For convenience of explanation, at this moment we rotate the grid and/or rename the pursuers if necessary, so that w.l.o.g. we assume that pursuer A is guarding some column i at $(i, m+1-\frac{\varepsilon}{2})$ for some $0 \le m \le n-2$ (recall that we assumed $\varepsilon < 1$), and Z lies to the east of column i. Our next objective is to advance the guarded column from i to $i+1$, by either moving A to column $i+1$ without revealing to Z the edge in which he is hiding, or by hiding B in column $i+1$. We use the following procedure `Advance` to achieve this goal (see Fig. 9).

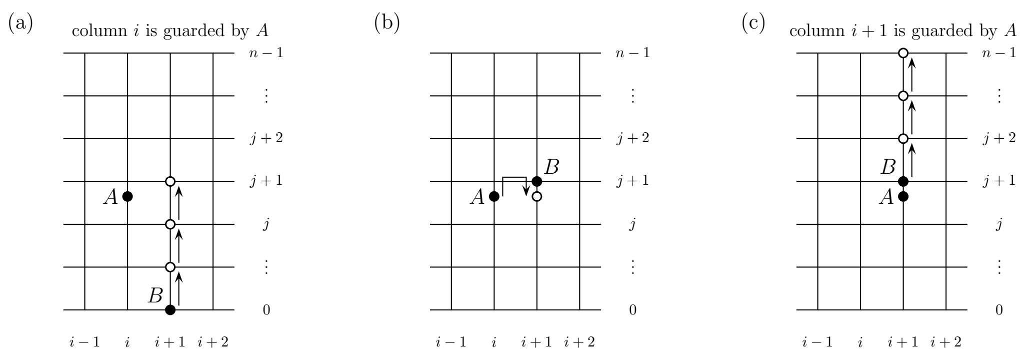

Figure 9: Procedure `Advance`, Step 3: (a) A is at $(i, m+1-\frac{\varepsilon}{2})$ and remains stationary until B reaches $(i+1, j+1) = (i+1, m+1)$, (b) A moves to column $i+1$ while B stays at $(i+1, m+1)$ for one time unit, and (c) B continues his move towards $(i+1, n-1)$.

Procedure `Advance`

1. /* At this moment, A is hiding at $(i, m+1-\frac{\varepsilon}{2})$ for some $0 \le m \le n-2$, B is elsewhere in the grid, and Z is to the east of column i. */

   B goes from its current position to vertex $(i+1, n-1)$ along any of the shortest paths.

   Termination Conditions for Step 1

   If, while moving, B sees Z within distance 1, then B starts chasing Z. If Z enters column i before B reaches $(i+1, n-1)$, then with probability at least $\frac{1}{n-1}$ A is within distance 1 of Z and starts chasing Z; if A is not with distance 1 of Z, then `Advance` ends in failure.

2. B moves south from vertex $(i+1, n-1)$ to $(i+1, 0)$ at unit speed.

   /* This action forces Z to leave the region between columns i and $i+1$ before Step 3 starts, if he is in that region at the end of Step 1. */

   For convenience, we reset now the time to $t = 0$.

3. For $j = 0, 1, \ldots, n-2$:

   3.1 In the time interval $[2j, 2j+1]$, A remains stationary, and B moves north from $(i+1, j)$ to $(i+1, j+1)$ at a speed of 1.

   3.2 In the time interval $[2j+1, 2j+2]$, B stays at $(i+1, j+1)$. If $j = m$ (i.e., A is presently at $(i, j+1-\frac{\varepsilon}{2}) = (i, m+1-\frac{\varepsilon}{2})$), then A moves north to $(i, m+1)$, east to $(i+1, m+1)$, and then south to $(i+1, m+1-\frac{\varepsilon}{2})$, at a speed of s.

Termination Conditions for Steps 2 and 3

As soon as one of the following holds, the above iteration is terminated and the specified action is taken.

(a) If A or B sees Z within distance 1, then he starts chasing Z immediately. This condition has the highest priority.

(b) If B sees Z to the east on some row, then B hides in column $i+1$ and guards it using `Hide3`, by randomly selecting an edge in that column and moving to its interior point at distance $\frac{\varepsilon}{2}$ from the northern endpoint.

(c) Z appears in column $i+1$ when A is not in column $i+1$. Then A is either located at point $(i, m+1-\frac{\varepsilon}{2})$ in column i for some $0 \leq m \leq n-2$, or currently moving to $(i+1, m+1-\frac{\varepsilon}{2})$ in Step 3.2. In either case, B tells A to move to vertex $(i+1, m+1)$ via row $m+1$ at speed s. When A reaches vertex $(i+1, m+1)$ within one time unit, with probability at least $\frac{1}{n-1}$ A is within distance 1 of Z and starts chasing Z. (If B happens to be within distance 1 of Z, then B starts chasing Z by termination condition (a).)

(d) Z appears in column $i+1$ when A is already in column $i+1$. In this case, with probability at least $\frac{1}{n-1}$, A is within distance 1 of Z and starts chasing Z. (Again, if B happens to be within distance 1 of Z, then B starts chasing Z by termination condition (a).)

(e) If Z appears in column $i+1$ but chasing does not start as described above, then `Advance` ends in failure.

The basic idea of Step 3 in `Advance` is as follows. Observe that Z can gain some knowledge about A's hiding location — without entering column i — in three ways:

1. Z sees A in row $j+1$ in the time interval $[2j+1, 2j+2]$ and knows that A's hiding place in column i was $(i, j+1-\frac{\varepsilon}{2})$. Hence A's new hiding place in column $i+1$ will be $(i+1, j+1-\frac{\varepsilon}{2})$.

2. Z observes that A does *not* appear in row $j+1$ in the time interval $[2j+1, 2j+2]$ and knows that A's hiding place in column i is *not* $(i, j+1-\frac{\varepsilon}{2})$.

3. Z enters column $i+1$, and either sees A or does not see A, to the south of B who is currently at $(i+1, y)$. In the former case, Z knows A's hiding place, and in the latter case, Z knows that A hides at a point to the north of level $\lceil y \rceil - 1$.

Thus, in both Cases 1 and 2, for every j, $0 \leq j \leq n-2$, Z has to be on row $j+1$ sometime in $[2j+1, 2j+2]$ to know whether A was hiding at $(i, j+1-\frac{\varepsilon}{2})$ in column i. Hence, by placing B at $(i+1, j+1)$ in the interval $[2j+1, 2j+2]$ for every j, we ensure that the guarded column is advanced from i to $i+1$ (by B) whenever Z attempts to gain knowledge about A's hiding location in Cases 1 and 2. We handle Case 3 by placing A within distance $\frac{\varepsilon}{2}$ of a vertex in column i, so that A can start chasing Z with probability at least $\frac{1}{n-1}$ when Z appears at any vertex in column $i+1$.

Lemma 6 *Assume that* `Advance` *is executed with A hiding in column i and Z to the east of column i. Then within $O(n)$ time, either chasing starts with probability at least $\frac{1}{n-1}$, or one of the pursuers hides in column $i+1$ with Z to the east of column $i+1$.*

Proof. As is already mentioned in the termination condition for Step 1, if in Step 1 pursuer B sees Z within distance 1, then B starts chasing Z. Also, if Z enters column i before B reaches $(i+1, n-1)$, then A starts chasing Z with probability at least $\frac{1}{n-1}$. So assume that B starts executing Step 2; observe that column i is still guarded by A. We consider three cases depending on the position of Z at the beginning of Step 2.

(i) Z is in column $i+1$ at the beginning of Step 2. In this case, either B is within distance 1 of Z and starts chasing Z (condition (a)) or, A moves to column $i+1$ within one unit time and, with probability at least $\frac{1}{n-1}$, starts chasing Z (condition (c)).

(ii) Z is located between columns i and $i+1$ at the beginning of Step 2. If B approaches Z to within distance 1 while moving along column $i+1$, then B starts chasing Z (condition (a)). Otherwise, Z must enter either column i or column $i+1$ — in the former case, with probability at least $\frac{1}{n-1}$ A is within distance 1 of Z and starts chasing Z (condition (a)); in the latter case, A moves to column $i+1$ within one unit time and, with probability at least $\frac{1}{n-1}$, starts chasing Z (condition (c)).

(iii) Z is to the east to column $i+1$ at the beginning of Step 2. Suppose chasing does not start by condition (a). If Z never appears in column $i+1$ and Step 3 is completed, or if `Advance` terminates because of condition (b), then clearly A or B hides in and guards column $i+1$ with Z to the east of column $i+1$. So assume otherwise, and hence, Z enters column $i+1$ (from the east) during the execution of Step 2 or 3. Then, as explained in termination conditions (c) and (d) for Steps 2 and 3, chasing starts with probability at least $\frac{1}{n-1}$.

□

Observe that if Z successfully escapes to the area to the west to column i during the execution of `Advance`, which results in failure of the current capture attempt, then A and B will immediately notice this fact.

Theorem 6 *With probability at least $\frac{1}{n-1}$, two pursuers with a maximum speed of $s = 1+\varepsilon$ can start chasing an arbitrary evader Z within $O(n^2)$ time. The expected time to capture Z by repeating this process is $O(n^3)$.*

Proof. By Lemma 6, the initial step combining the modified procedures `Search1` and `Hide3`, followed by repeated executions of procedure `Advance` (with suitable renaming of the pursuers) ensures that within $O(n^2)$ time, either (1) chasing starts with probability at least $\frac{1}{n-1}$, or (2) a pursuer hides in column $n-1$ with Z in the empty area to the east of column $n-1$. Since the latter is impossible, it must be the case that chasing starts with probability at least $\frac{1}{n-1}$. A capture then follows within $O(\min\{1/\varepsilon, n^2\}) = O(n^2)$ time by Lemma 2. □

5 A three-pursuer deterministic algorithm

In this section, we show that three pursuers A, B and C with a maximum speed of $s = 1$ can capture an arbitrary evader Z. Taking into account Lemma 2, our goal is to let one of the three pursuers start chasing Z. To get into this favorable situation, the pursuers construct a "moving trap" for Z. The pursuers start by executing the following action sequence (see Fig. 10).

Procedure `Search2`

1. Initially A, B and C are at $(0,1)$, $(2,0)$ and $(1,1)$, respectively.

2. For $i = 1, 2, \ldots, n-2$:

   2.1 C moves north to $(i, n-1)$, south to $(i, 1)$, and east to $(i+1, 1)$.

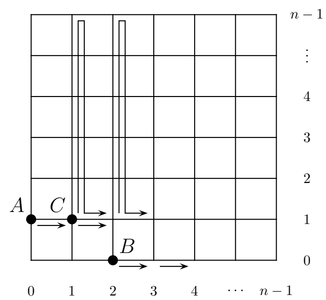

Figure 10: Traversal of G_n by A, B and C.

2.2 A and B simultaneously move east to $(i, 1)$ and $(i+2, 0)$, respectively. (This part is not executed when $i = n - 2$.)

We first need some definitions. Suppose A and B are at intersections $(i-1, j+1)$ and $(i+1, j)$, respectively, for some integers $2 \le i \le n-2$ and $0 \le j \le n-2$. See Fig. 11(a). The set of points $(x, y) \in G_n$ such that either (1) $x = i - 1$ and $y > j + 2$ or (2) $i - 1 < x \le i + 1$ and $y > j + 1$ is called a *trap*, and is denoted by TRAP(i, j). Symmetrically, if A and B are at intersections $(i-1, j)$ and $(i+1, j+1)$, respectively, then we define TRAP(i, j) to be the set of points (x, y) such that either (1) $i - 1 \le x < i + 1$ and $y > j + 1$ or (2) $x = i + 1$ and $y > j + 2$. See Fig. 11(b). Let TRAPPED(i, j) be the condition that Z is in TRAP(i, j) and the pursuers know this fact. The following lemma is similar to Lemma 4 of [20] (we omit the proof).

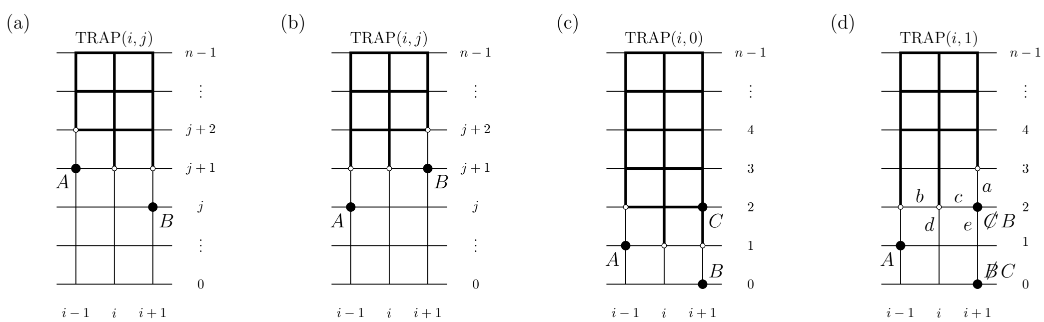

Figure 11: (a-b) TRAP(i, j). (c) Z is in TRAP$(i, 0)$. (d) TRAP$(i, 1)$: B and C swap their names.

Lemma 7 *If the above action sequence is executed, then within $O(n^2)$ time either Z is chased by a pursuer or* TRAPPED$(i, 0)$ *holds for some $1 \le i \le n-2$.*

The pursuers terminate the execution of the above action sequence as soon as TRAPPED$(i, 0)$ holds for some $1 \le i \le n-2$. Note that at this moment, A and B are at intersections $(i-1, 1)$ and $(i+1, 0)$, respectively. (Refer to Fig. 11 with $j = 0$.) Thereafter, A and B move east and west as necessary to keep Z in a trap (or start chasing Z if they can). Specifically:

1. If Z appears at $(i-1, 2)$, $(i, 1)$ or $(i+1, 1)$, then A or B can start chasing Z.
2. If Z appears at $(i-1, l)$ for some $3 \le l \le n-1$ at time t, then A and B move west at t and reach intersections $(i-2, 1)$ and $(i, 0)$, respectively, at $t+1$. Clearly Z is in TRAP$(i-1, 0)$ at $t+1$ and the pursuers know this fact, and hence TRAPPED$(i-1, 0)$ holds at $t+1$.

3. Similarly, if Z appears at $(i+1, l)$ for some $2 \le l \le n-1$ at time t, then A and B move east and reach $(i, 1)$ and $(i+2, 0)$, respectively, at $t+1$, so that TRAPPED$(i+1, 0)$ holds at $t+1$.

While A and B keep Z inside a trap as described above, C moves to row 2 and then moves within row 2 until it is on the same column as B. See Fig. 11(c), where A, B and C are at $(i-1, 1)$, $(i+1, 0)$ and $(i+1, 2)$, respectively. When this happens, say at time t, C and B swap their names and roles, as shown in Fig. 11(d). Then A and the new B determine a new trap TRAP$(i, 1)$ which is smaller than the previous TRAP$(i, 0)$. The purpose of this operation is to make the trap smaller so that the pursuers will get closer to Z.

Note that Z may or may not be in TRAP$(i, 1)$ after the swap at time t. Since Z is in TRAP$(i, 0)$ before the swap, if it is not in TRAP$(i, 1)$ after the swap, then it must be on one of the five edges a, b, c, d and e shown in Fig. 11(d). If Z in on a, c or e at time t, then B knows this fact and he can start chasing Z at t. If B sees Z at t to the west at distance greater than 1, then (since Z is in TRAP$(i, 0)$) Z must be on b, and hence both A and B move towards b and one of them can start chasing Z by time $t+1$. Whether Z is on d can be determined by moving C to $(i, 1)$: If C finds Z on d at $t+1$, then A and B move towards d and one of them can start chasing Z by $t+2$; if C does not see Z on d at $t+1$, then the pursuers know that Z is in TRAP$(i, 1)$, and hence TRAPPED$(i, 1)$ holds at $t+1$ (or earlier if Z becomes visible to A or B before $t+1$).

In summary, either one of the pursuers starts chasing Z by $t+2$, or the pursuers find that Z is in TRAP$(i, 1)$ by $t+1$. If Z is found to be in TRAP$(i, 1)$ by $t+1$, then the pursuers can repeat the same operation, i.e., A and B keep Z within a trap while C moves to row 3 and then to the same column as A. If this process is repeated, then eventually one of the pursuers must start chasing Z since TRAP$(i, n-2)$ is empty for any i. Also, since the time needed between two consecutive swaps is $O(n)$, chasing must start within $O(n^2)$ time. We thus obtain the following result.

Theorem 7 *Using a deterministic algorithm, three pursuers with a maximum speed of $s = 1$ can capture an arbitrary evader Z within $O(n^2)$ time.*

6 Concluding remarks

In most of our results in this paper we have only assumed a limited distance detection capability for the pursuers — they can tell only whether or not Z in sight is within distance 1 of their locations, or have an approximate distance detection capability with constant ratio. This feature is desirable in practical applications. In addition, for some results requiring exact distance detection capability (such as the one-pursuer randomized algorithm for capturing an arbitrary Z), we pointed out that similar results hold under the assumption of approximate distance detection capability. Our algorithms are applicable in many scenarios with autonomous robots in locating and capturing a hostile or uncontrollable robot moving on the ground in a grid-like city environment, or in a contaminated environment not suitable for human intervention. A similar pursuit-evasion problem has recently been considered in a 3D grid [23].

Some questions remain for future study. The maximum speed requirements and the capture times in our algorithms can be probably further reduced. Particularly interesting questions are: Can a single pursuer with a maximum speed of $s = 1 + \varepsilon$ capture an (arbitrary) evader in G_n in a polynomial (in n) number of steps? Is it possible to capture an arbitrary evader in polynomial time by using two pursuers with a maximum speed $s = 1$? Or by using a small group of pursuers with a maximum speed $s < 1$?

It is also natural to consider similar questions for the case of "incomplete" grids, i.e., arbitrary connected subgraphs of G_n. If the maximum speed of the pursuers is $s = 1$, then for any $k \ge 1$, a tree-like environment (similar in spirit to those found in [22]) can be constructed as an incomplete grid for which k pursuers, using a deterministic capture algorithm, are necessary (and sufficient) to capture a fugitive; see Fig. 12. For randomized capture algorithms, however, the situation might be different.

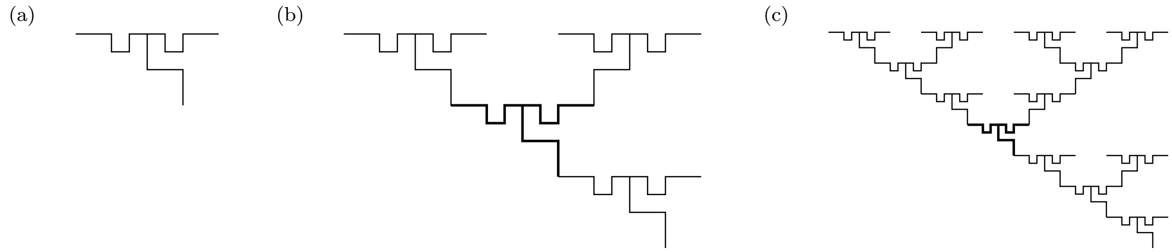

Figure 12: Tree-like grids $G(1)$, $G(2)$ and $G(3)$ that require 1, 2 and 3 deterministic pursuers, respectively.

It is worth noting that two deterministic pursuers suffice (for capturing Z) in the discrete model, despite the fact that all players have the same "speed" of one edge per step. It is obvious that one pursuer is not enough even in the discrete model.

Theorem 8 *In the discrete model, with no direction and exact distance detection capabilities for pursuers, two pursuers can capture Z in $O(n^2)$ time steps using a deterministic algorithm.*

Proof. A and B always maintain a tandem formation in which A is exactly one edge to the west of B (see Fig. 13).

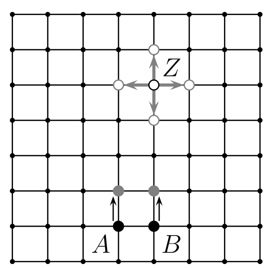

Figure 13: Possible moves for Z as A and B move towards him in a tandem formation in the discrete model.

A and B start by moving in tandem along the south boundary of the grid (row 0) eastward from the west end. When one of them sees Z, they move north in tandem, and continue to do so until one of them starts chasing Z or they lose sight of Z. In the latter case, Z must be in a row to the north of A and B, and depending on whether Z was visible to A or B in the previous step, he is either on the column immediately to the west of A or immediately to the east of B. Then A and B repeatedly move in the direction of Z (either west or east) within their current row, until one of them sees Z again. They then repeat the same procedure, by first moving north towards Z. It is easy to see that Z is forced to remain to the north of A and B, and chasing starts in $O(n^2)$ time steps. Arguing as in the proof of Lemma 2 for the continuous model, one can show that the pursuers in the tandem formation can capture Z within $O(n^2)$ steps after chasing starts. □

Acknowledgment. We wish to thank Amol Mali and Kazuo Sugihara for interesting discussions on these topics.

References

[1] M. Adler, H. Räcke, N. Sivadasan, C. Sohler, B. Vöcking, Randomized pursuit-evasion in graphs, *Combinatorics, Probability and Computing* 12(3) (2003), 225–244.

[2] M. Aigner, M. Fromme, A game of cops and robbers, *Discrete Applied Mathematics* 8 (1984), 1–12.

[3] T. Andreae, Note on a pursuit game played on graphs, *Discrete Applied Mathematics* 9 (1984), 111–115.

[4] F. Berger, A. Gilbers, A. Grüne, R. Klein, How many lions are needed to clear a grid?, *Algorithms* 2(3) (2009), 1069–1086.

[5] P. Brass, K. D. Kim, H.-S. Na, Ch.-S. Shin, Escaping offline searchers and isoperimetric theorems, *Computational Geometry: Theory and Applications* 42(2) (2009), 19–126.

[6] R. W. Dawes, Some pursuit-evasion problems on grids, *Information Processing Letters* 43 (1992), 241–247.

[7] A. Dumitrescu, I. Suzuki, P. Żyliński, Offline variants of the "Lion and Man" problem – Some problems and techniques for measuring crowdedness and for safe path planning –, *Theoretical Computer Science* 399(3) (2008), 220–235.

[8] F. V. Fomin, D. Thilikos, An annotated bibliography on guaranteed graph searching, *Theoretical Computer Science* 399(3) (2008), 236–245.

[9] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, R. Motwani, Visibility-based pursuit-evasion in a polygonal environment, *International Journal of Computational Geometry and Applications* 9(4/5) (1999), 471–493.

[10] V. Isler, S. Kannan, S. Khanna, Randomized pursuit-evasion with local visibility, *SIAM Journal on Discrete Mathematics* 20(1) (2006), 26–41.

[11] T. Kameda, M. Yamashita, I. Suzuki, On-line polygon search by a seven-state boundary 1-searcher, *IEEE Transactions on Robotics* 22(3) (2006), 446-460.

[12] L. M. Kirousis, C. H. Papadimitriou, Searching and pebbling, *Theoretical Computer Science* 47 (1986), 205–218.

[13] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, C. H. Papadimitriou, The complexity of searching a graph, *Journal of the ACM* 35(1) (1986), 18–44.

[14] S. W. Neufeld, A pursuit-evasion problem on a grid, *Information Processing Letters* 58 (1996), 5–9.

[15] S. W. Neufeld, R. Nowakowski, A game of cops and robbers played on product of graphs, *Discrete Mathematics* 186 (1998), 253–268.

[16] N. Nisse, K. Suchan, Fast robber in planar graphs, in *Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science (WG), Lecture Notes in Computer Science* 5344 (2008), 312–323.

[17] T. D. Parsons, Pursuit-evasion in a graph, *Lecture Notes in Mathematics* 642 (1976), 426–441.

[18] E. Reingold, L. Alonso, Lower bounds for cops and robber pursuit, manuscript (2008).

[19] K. Sugihara, I. Suzuki, On a pursuit-evasion problem related to motion coordination of mobile robots, in *Proceddings of the 21st Hawaii International Conference on System Sciences*, Kailua-Kona, Hawaii (1988), 218–226.

[20] K. Sugihara, I. Suzuki, Optimal algorithms for a pursuit-evasion problem in grids, *SIAM Journal on Discrete Mathematics* 2(1) (1989), 126–143.

[21] I. Suzuki, M. Yamashita, Searching for a mobile intruder in a polygonal region, *SIAM Journal on Computing* 21 (1992), 868–888.

[22] I. Suzuki, M. Yamashita, H. Umemoto, T. Kameda, Bushiness and a tight worst-case upper bound on the search number of a simple polygon, *Information Processing Letters* 66 (1998), 49–52.

[23] I. Suzuki, P. Żyliński, Capturing an evader in a building — Randomized and deterministic algorithms for mobile robots, *IEEE Robotics and Automation Magazine* 15(2) (2008), 16–26.