

Line Search for an Oblivious Moving Target

Jared Coleman ✉ 🏠 

University of Southern California, USA

Evangelos Kranakis ✉ 🏠 

Carleton University, Ottawa, Ontario, Canada

Danny Krizanc ✉ 🏠

Wesleyan University, Middletown CT, USA

Oscar Morales-Ponce ✉ 🏠 

California State University, Long Beach, CA, USA

Abstract

Consider search on an infinite line involving an autonomous robot starting at the origin of the line and an oblivious moving target at initial distance $d \geq 1$ from it. The robot can change direction and move anywhere on the line with constant maximum speed 1 while the target is also moving on the line with constant speed $v > 0$ but is unable to change its speed or direction. The goal is for the robot to catch up to the target in as little time as possible.

The classic case where $v = 0$ and the target's initial distance d is unknown to the robot is the well-studied “cow-path problem”. Alpert and Gal [2] gave an optimal algorithm for the case where a target with unknown initial distance d is moving *away* from the robot with a known speed $v < 1$. In this paper we design and analyze search algorithms for the remaining possible knowledge situations, namely, when d and v are known, when v is known but d is unknown, when d is known but v is unknown, and when both v and d are unknown. Furthermore, for each of these knowledge models we consider separately the case where the target is moving away from the origin and the case where it is moving toward the origin. We design algorithms and analyze competitive ratios for all eight cases above. The resulting competitive ratios are shown to be optimal when the target is moving towards the origin as well as when v is known and the target is moving away from the origin.

2012 ACM Subject Classification Theory of computation → Online algorithms; Theory of computation → Adversary models

Keywords and phrases Infinite Line, Knowledge, Oblivious, Robot, Search, Search-Time, Speed, Target

Funding *Evangelos Kranakis*: Research supported in part by NSERC Discovery grant.

1 Introduction

Search is important to many areas of computer science and mathematics and has received the attention of numerous studies. In the simplest search scenario, one is interested in the optimal trajectory of a single autonomous mobile agent (also referred to simply as a robot) tasked with finding a target placed at an unknown location on the infinite line. The line search problem is to give an algorithm for the agent so as to minimize the competitive ratio defined as the supremum over all possible target locations of the ratio of the time the agent takes to find the target and the time it would take if the target's initial position was known to the robot ahead of time. This classic problem has led to many variations (see [2] for more on its history).

In this paper we consider an extension of the line search problem involving an autonomous robot and an oblivious moving target. The search is again performed on an infinite line and concerns an autonomous robot starting at the origin of the line but differs from the previously studied case in that the search is for a *moving* target whose speed and direction are not necessarily known to the searching robot. The robot starts at the origin and the target at an arbitrary distance d from the origin. The target is moving with constant speed and is oblivious in that it cannot change its speed and/or direction of movement. We consider and analyze several alternative knowledge-based scenarios in which the target's speed and initial distance from the origin may be known or unknown to the searching robot. The case where a target with unknown initial distance from the origin is moving away from the origin was solved by Alpern and Gal [2]. As far as we are aware, these are the first results for the remaining cases.

1.1 Notation and terminology

On the infinite real line, consider an autonomous robot which is initially placed at the origin whose maximum speed is 1 and an oblivious robot (also referred to as the moving target) initially placed at a distance d to the right or left of the origin and moving with constant speed $v > 0$. As is usually done in linear search and in order to avoid trivial considerations on the competitive ratio by adversarially placing the target very close to the robot, we assume that d is not smaller than the unit distance, i.e., $d \geq 1$.

The target may be moving away from or toward the origin. If it is moving away, we assume its speed is strictly less than 1 as otherwise the problem can not be solved. Further, we assume that the autonomous robot knows the direction the target is moving (away from or toward the origin). The search is completed as soon as the robot and target are co-located.

The movement of the autonomous robot is determined by a trajectory which is defined as a continuous function $t \rightarrow f(t)$, with $f(t)$ denoting the location of the robot at time t . Moreover, it is true that $|f(t) - f(t')| \leq u|t - t'|$, for all t, t' , where u is the speed of the agent (be that the searching robot or the oblivious target). The autonomous robot can move with its own constant speed and during the traversal of its trajectory it may stop and/or change direction instantaneously and at any time as specified by the search algorithm.

A search strategy is a sequence of movements followed by the robot. The competitive ratio of a search strategy X , denoted CR_X , is defined as the supremum over all possible initial target locations and speeds of the ratio of the time the agent takes to find the target and the time it would take if the target's initial position was known to the robot ahead of time. The competitive ratio of a certain type of search problem is the infimum of CR_X taken over all possible strategies X for this problem. By abuse of notation we may drop mention of X when this is easily implied from the context.

Our goal in this paper is to prove bounds on the competitive ratios of algorithms under four different knowledge models:

1. **FullKnowledge**: The robot knows both the target's speed v and its initial distance d .
2. **NoDistance**: The robot knows the target's speed v but not its initial distance d .
3. **NoSpeed**: The robot knows the target's initial distance d , but not its speed v .
4. **NoKnowledge**: The robot knows neither the target's speed v nor its initial distance d .

For all knowledge models, the robot does not know the target's initial position. We study each of the above knowledge models for the case where the target is moving toward the origin (**Toward**) and where it is moving away (**Away**) from the origin. In each case, we assume the robot knows the direction of travel of the target.

1.2 Related Work

Several research papers have considered the search problem for a robot searching for a static (fixed) target placed at an unknown location on the real line, see [3, 13]. The problem was first independently considered in a stochastic setting by Bellman and Beck in the 1960's (cf. [4, 5] as well as [3, 13]). In a deterministic setting it is now well known that the optimal trajectory for this single agent search uses a doubling strategy whose trajectory attains a competitive ratio of 9. Linear search has attracted much attention and been the focus of books including [1, 2, 14].

The case of a moving target appears to have been first considered by McCabe [12]. In that paper, the problem of searching for an oblivious target that follows a Bernoulli random walk on the integers is considered. For the case of a deterministic oblivious searcher, the only result we are aware of is found in Alpern and Gal [2]. There they consider the case where the target is moving away from the origin at a constant speed $v < 1$ which is known to the searching robot. Only the initial distance of the target is unknown. They give an algorithm with optimal competitive ratio for this case.

Our problem is reminiscent of the problem of catching a fugitive in a given domain which is generally referred to as the cops and robbers problem [6]. The main difference is that in those problems, the target (robber) is itself an autonomous agent. As a result, the techniques considered there do not apply to our case.

Our problem is also related to rendezvous (of two robots) on an infinite line but it differs because in our case only one of the robots is autonomous while the other is oblivious. Related studies on the infinite line include rendezvous with asymmetric clocks [8] and asynchronous deterministic rendezvous [11]. More recent work on linear search concerns searching for a static target by a group of cooperating robots, some of which may have suffered either crash [9] or Byzantine [7] faults.

1.3 Results of the paper

In all situations considered it is unknown to the robot whether the target is initially to the left or to the right of the origin. We analyze the competitive ratio in four situations which reflect what knowledge the robot has about the target. We present results on the **FullKnowledge** model (the robot knows v and d) in Section 2, the **NoDistance** model (the robot knows v but not d) in Section 3, the **NoSpeed** model (the robot knows d but not v) in Section 4, and the **NoKnowledge** model (the robot knows neither v nor d) in Section 5. For each of these models we study separately the case when the target is moving away or toward the origin (this knowledge being available to the robot). The results are summarized in Table 1. We conclude with a summary and additional open problems.

Knowledge	Movement	Competitive Ratio	Section
v, d	Away	$CR = 1 + \frac{2}{1-v}$	2.1
	Toward	$CR = 1 + \frac{2}{1+v}$ if $v < 1$ $CR = 1 + \frac{1}{v}$ otherwise	2.2
v	Away	$CR = 1 + 8 \frac{1+v}{(1-v)^2}$	3.1 [2]
	Toward	$CR = 1 + \frac{1}{v}$ if $v \geq \frac{1}{3}$ $CR = 1 + 8 \frac{1-v}{(1+v)^2}$ otherwise	3.2
d	Away	$CR \leq 5$ if $v \leq \frac{1}{2}$ $CR \leq 1 + 16 \frac{(\log \frac{1}{1-v})^2}{(1-v)^4}$ otherwise	4.1
	Toward	$CR = 3$	4.2
\emptyset	Away	$CR \leq 1 + \frac{16}{d} [\log \log (\max(d, \frac{1}{1-v})) + 3] \cdot \max(d, \frac{1}{1-v})^8 \cdot \log^2 [\max(d, \frac{1}{1-v})]$	5.1
	Toward	$CR = 1 + \frac{1}{v}$	5.2

■ **Table 1** Table of competitive ratio bounds proven for each knowledge model for cases with the target moving away from or towards the origin with speed v and initial distance d from the robot which is moving with speed 1. Equalities indicate that tight upper and lower bounds are proven.

2 The FullKnowledge Model

We first study the model where the robot knows the target's speed v and its initial distance from the origin d .

2.1 The FullKnowledge/Away Model

For the case when the target is moving away from the origin, clearly if $v \geq 1$ then the robot can never catch the target. Thus, for this model (and all other **Away** models), we assume $v < 1$. In this section, we will analyze an algorithm where the robot chooses a direction and moves for time $\frac{d}{1-v}$. If the robot does not find the target after moving for time $\frac{d}{1-v}$ in one direction, then it changes direction and continues moving until it does.

■ **Algorithm 1** Online Algorithm for FullKnowledge/Away Model

-
- 1: **input:** target speed v and initial distance d
 - 2: choose any direction and go for time $\frac{d}{1-v}$
 - 3: **if** target not found **then**
 - 4: change direction and go until target is found
-

► **Theorem 1.** *For the FullKnowledge/Away model, Algorithm 1 has an optimal competitive ratio of*

$$1 + \frac{2}{1-v}. \quad (1)$$

Proof. By Algorithm 1, the robot goes in one direction for a time $\frac{d}{1-v}$. Observe that if the robot does not encounter the target after this amount of time, it must be on the opposite side of the origin (in the other direction). At the time the robot changes direction, its distance to the target will be equal to $\frac{d}{1-v} + d + \frac{dv}{1-v} = \frac{2d}{1-v}$. Thus, the total time required until the robot catches up to the target is at most

$$\frac{d}{1-v} + \frac{2d}{(1-v)^2}.$$

Clearly then, the competitive ratio is at most

$$\frac{\frac{d}{1-v} + \frac{2d}{(1-v)^2}}{\frac{d}{1-v}} = 1 + \frac{2}{1-v}$$

which is as claimed in Equation (1) above.

Optimality follows from the fact that regardless of which direction the robot chooses to travel, the adversary can place the target in the opposite direction. Moreover, for the robot to catch up to the target it must visit one of the points $\pm \frac{d}{1-v}$. If the robot visits location $\frac{d}{1-v}$ to the right (resp. left) the adversary places the target on the left (resp. right). Therefore the completion time will be at least $\frac{d}{1-v} + \frac{2d}{(1-v)^2}$. This shows the upper bound is tight and completes the proof of Theorem 1. ◀

2.2 The FullKnowledge/Toward Model

Consider the following algorithm which is similar to Algorithm 1.

■ **Algorithm 2** Online Algorithm for the FullKnowledge/Toward Model

-
- 1: **input:** target speed v and initial distance d
 - 2: choose any direction and go for time $\frac{d}{1+v}$
 - 3: **if** target not found **then**
 - 4: change direction and go until target is found
-

▶ **Theorem 2.** *For the FullKnowledge/Toward model, Algorithm 2 has competitive ratio at most*

$$1 + \frac{2}{1+v}. \tag{2}$$

Proof. The robot goes in one direction for a time $\frac{d}{1+v}$. If the robot finds the target in this time, the algorithm is clearly optimal. If, however, the robot does not find the target, then it must be on the opposite side of the origin (in the other direction). If this is the case, then by time $\frac{d}{1+v}$ the target has moved a distance $\frac{dv}{1+v}$ and is at distance $d - \frac{dv}{1+v} = \frac{d}{1+v}$ from the origin. Therefore at the time the robot changes direction, the distance between robot and target is $\frac{2d}{1+v}$. Thus, the robot will encounter the target in additional time $\frac{2d}{(1+v)^2}$. It follows that the total time required for the robot to meet the target is $\frac{d}{1+v} + \frac{2d}{(1+v)^2}$ and the resulting competitive ratio satisfies

$$CR \leq \frac{\frac{d}{1+v} + \frac{2d}{(1+v)^2}}{\frac{d}{v+1}} = 1 + \frac{2}{1+v}.$$

This completes the proof of Theorem 2. ◀

▶ **Theorem 3.** *For the FullKnowledge/Toward model, the competitive ratio of any online algorithm is at least $1 + \frac{2}{1+v}$, provided that $v < 1$. In particular, Algorithm 2 is optimal for $v < 1$.*

Proof. Consider any algorithm for a robot starting at the origin to meet a target initially placed at an unknown location distance d from the origin. For any point at distance a from the origin, the target takes exactly $\frac{d-a}{v}$ time to reach a . Then, let t denote the time the robot first passes through a point at distance a from the origin. If $t < \frac{d-a}{v}$, then the robot cannot

6 Line Search for an Oblivious Moving Target

know whether the target is on the same or opposite side of the origin. On the other hand, if $t \geq \frac{d-a}{v}$ and it has not encountered the target, then the target must be on the opposite side of the origin. Thus, given a trajectory, let $\pm a$ be the first point such that the robot is at position $\pm a$ at time exactly $\frac{d-a}{v}$. Clearly such a point must exist for any trajectory since the target is moving toward the origin. Then whichever side of the origin the robot is on, consider the instance where the target started on the opposite side. Clearly then, the robot takes an additional time at least $\frac{2a}{1+v}$ to reach the target. Thus, the competitive ratio is given by:

$$\frac{\frac{d-a}{v} + \frac{2a}{1+v}}{\frac{d}{1+v}} = \frac{(d-a)(1+v) + 2av}{vd} = 1 + \frac{1}{v} + \frac{a(v-1)}{dv}. \quad (3)$$

Observe that whenever $v < 1$, the right-hand side of Equation (3) satisfies

$$1 + \frac{1}{v} + \frac{a(v-1)}{dv} \geq 1 + \frac{1}{v} + \frac{\frac{d}{1+v}(v-1)}{dv} = 1 + \frac{2}{1+v}$$

which completes the proof of Theorem 3. ◀

With Theorem 3 proved, we know Algorithm 2 is optimal for any value of v between 0 and 1, but what about when $v > 1$? In this case, we'll prove the following algorithm is optimal: the robot waits at the origin forever. We call this algorithm “the waiting algorithm”.

► **Theorem 4.** *Whenever the target is moving toward the origin with speed $v \geq 1$, the waiting algorithm has an optimal competitive ratio of $1 + \frac{1}{v}$.*

Proof. Clearly the algorithm takes exactly time d/v to complete and so the upper bound follows trivially. For the lower bound, we build upon the proof of Theorem 3. It simply remains to consider Equation (3) for $v > 1$. In this case, the right-hand side of Equation (3) is increasing with respect to $a \geq 0$, so

$$1 + \frac{1}{v} + \frac{a(v-1)}{dv} \geq 1 + \frac{1}{v}.$$

This completes the proof of Theorem 4. ◀

► **Remark 5.** Observe that the waiting algorithm makes no use of the target's speed or initial distance and therefore, as long as the target is moving toward the origin, applies directly to the other knowledge models.

3 The NoDistance Model

In this section we assume that the robot knows v but not d . Consider the following zig-zag algorithm with “expansion ratio” $a > 0$ (with the value of a to be determined).

Algorithm 3 Online Algorithm for NoDistance/Away and NoDistance/Toward Models

```

1: input: target speed  $v$  and expansion ratio  $a$ 
2:  $i \leftarrow 0$ 
3: while target not found do
4:   if at origin then
5:      $d \leftarrow (-a)^i$ 
6:      $i \leftarrow i + 1$ 
7:   else if at  $d$  then
8:      $d \leftarrow 0$ 
9:   move toward  $d$ 

```

3.1 The NoDistance/Away Model

The following result was shown by Alpern and Gal [2].

► **Theorem 6.** *For the NoDistance/Away model, Algorithm 3 with $a = 2\frac{1+v}{1-v}$ has an optimal competitive ratio of*

$$1 + 8 \frac{1+v}{(1-v)^2}.$$

3.2 The NoDistance/Toward Model

Recall first the statement made in Remark 5, that the optimality of the waiting algorithm (which makes no use of any knowledge of d) holds for any d as long as $v \geq 1$. Thus, we need only consider scenarios where $0 \leq v < 1$. As we will see, however, when the target is moving toward the origin, the waiting algorithm is optimal for far slower targets! In general, since the target is moving toward the origin, the robot need not search ever-increasing distances away from the origin (i.e. execute Algorithm 3 with an expansion ratio $a > 1$). We call any algorithm which involves the robot *never* traveling further than some finite distance from the origin (in one or both directions) a *contracting algorithm*. Note that Algorithm 3 for $0 < a \leq 1$ is a contracting algorithm and $a = 0$ is exactly the waiting algorithm. We'll start by showing that any contracting algorithm cannot have a better competitive ratio than the waiting algorithm:

► **Theorem 7.** *The competitive ratio of Algorithm 3 for any $0 \leq a \leq 1$ is $1 + \frac{1}{v}$.*

Proof. Let d' be the finite distance further than which the robot will never travel in at least one direction. Then consider the scenario where the target is initially a distance $d = c \cdot d' \gg d'$ from the origin in the same direction. Then the competitive ratio is at least

$$\sup_c \frac{\frac{cd' - d'}{v}}{\frac{cd'}{v+1}} = \sup_c \frac{c-1}{c} \frac{1+v}{v} = \lim_{c \rightarrow \infty} \frac{c-1}{c} \left(1 + \frac{1}{v}\right) = 1 + \frac{1}{v}$$

which proves Theorem 7. ◀

By Theorem 7, any algorithm which hopes to out-perform the waiting algorithm must be expanding. Now we show that the following hybrid algorithm, Algorithm 4, is optimal.

■ **Algorithm 4** Wait or Zig-Zag Search Algorithm for NoDistance/Toward model

-
- 1: **input:** target speed v
 - 2: **if** $v \geq \frac{1}{3}$ **then**
 - 3: execute waiting algorithm
 - 4: **else**
 - 5: execute Algorithm 3 with $a = 2\frac{1-v}{1+v}$
-

► **Theorem 8.** *For the NoDistance/Toward model, the competitive ratio of Algorithm 4 is at most*

$$\begin{cases} 1 + \frac{1}{v} & \text{if } v \geq \frac{1}{3} \\ 1 + 8\frac{1-v}{(1+v)^2} & \text{if } v < \frac{1}{3} \end{cases} \quad (4)$$

Proof. The first case is trivial: the competitive ratio of the waiting algorithm is exactly $1 + \frac{1}{v}$ by Theorem 4. The second case, however, is a bit more complicated. First, observe that if $v < \frac{1}{3}$ then a must be less than 3. Indeed, consider the scenario where the robot “just misses” the target on the very first round of the algorithm (after traveling a distance 1 in some direction and then turning around). Then the competitive ratio of the algorithm is

$$1 + \frac{2a + 2}{1 + v}$$

which is greater than $1 + 8\frac{1-v}{(1+v)^2}$ for any $a > 3$ and $v > 0$:

$$1 + \frac{2a + 2}{1 + v} > 1 + \frac{8}{1 + v} > 1 + \frac{8}{1 + v} \cdot \frac{1 - v}{1 + v}$$

since $\frac{1-v}{1+v} < 1$.

Now, consider the round k when the robot catches up to the target and observe that

$$a^{k-2} < d - \left(2 \sum_{i=0}^{k-3} a^i + a^{k-2} \right) v = d - \left(2 \frac{a^{k-2} - 1}{a - 1} + a^{k-2} \right) v$$

since otherwise, the robot would have caught up to the target in round $k - 2$. This yields the following inequality which will prove useful in analyzing the competitive ratio below:

$$\begin{aligned} a^{k-2} &< d - \left(2 \frac{a^{k-2}}{a - 1} + a^{k-2} \right) v \\ &\leq d \frac{a - 1}{a - 1 + v(a + 1)} + \frac{v}{a - 1 + v(a + 1)} \\ &\leq d \frac{a - 1}{a - 1 + v(a + 1)} + \frac{1}{4a - 2} \end{aligned} \quad (5)$$

Observe the worst competitive ratio, then, is given by the situation where the robot “just misses” the target on the $(k - 2)^{\text{th}}$ round and catches up to it only on round k . It follows the competitive ratio of Algorithm 4 is

$$\frac{2 \sum_{i=0}^{k-3} a^i + a^{k-2} + \frac{2(a^{k-2} + a^{k-1})}{1+v}}{\frac{d}{1-v}} \leq \frac{2 \frac{a^{k-2}-1}{a-1} + a^{k-2} + \frac{2(a^{k-2} + a^{k-1})}{1+v}}{\frac{d}{1-v}}$$

which, by Inequality (5) (and by substituting each a^{k-2} with the right-hand side of Inequality (5)), is less than or equal to

$$CR \leq 1 + \frac{1}{2} \left[\frac{1}{d} \left(\frac{a-3}{a-1} \cdot \frac{v(5-7a)}{1-3a+2a^2} \right) + \frac{4a^2}{(a-1)+v(a+1)} \right] \quad (6)$$

$$\begin{aligned} &\leq \lim_{d \rightarrow \infty} 1 + \frac{1}{2} \left[\frac{1}{d} \left(\frac{a-3}{a-1} \cdot \frac{v(5-7a)}{1-3a+2a^2} \right) + \frac{4a^2}{(a-1)+v(a+1)} \right] \\ &= 1 + \frac{2a^2}{(a-1)+v(a+1)} \end{aligned} \quad (7)$$

which follows since the right-hand side of Inequality (6) is increasing with respect to d on $1 < a \leq 3$. Finally, the right-hand side of Inequality (7) is minimized at $a = 2\frac{v-1}{v+1}$ with a value of $1 + 8\frac{1-v}{(1+v)^2}$, which proves Theorem 8. \blacktriangleleft

Now we show that Algorithm 4 is optimal by proving a tight lower bound on the competitive ratio for any online algorithm. Our proof is based on techniques developed in [10]. Let $X(t)$ denote the robot's position at time t according to a given strategy.

► **Theorem 9.** *For the NoDistance/Toward model, any strategy X has a competitive ratio of at least*

$$\begin{cases} 1 + \frac{1}{v} & \text{if } v \geq \frac{1}{3} \\ 1 + 8\frac{1-v}{(1+v)^2} & \text{otherwise} \end{cases}$$

Proof. Let $\beta_t = \inf_{t' > t} \frac{t'}{|X(t')|}$. Clearly, then, if $t_1 \leq t_2$ then $\beta_{t_1} \leq \beta_{t_2}$. Furthermore, $\beta_t \geq 1$ for all t since the maximum speed of the robot is 1. Now let $\beta = \lim_{t \rightarrow \infty} \beta_t$. By definition of the limit infimum, there must exist a finite time t such that $\beta \leq \frac{t'}{|X(t')|}$ for all $t' \geq t$ and thus there must exist a time $t_1 > \frac{\beta(\beta+1)}{\beta-1}t$ such that the robot reaches a point (without loss of generality, on the right side of the origin) $X(t_1) = \frac{t_1}{\beta+\epsilon_1}$ for any arbitrarily small $\epsilon_1 > 0$. Consider such a time and observe that, by construction, the robot could not have reached any point to the left of $x_0 = -\frac{t_1 - X(t_1)}{1+\beta}$ after time $t_0 = \frac{\beta(t_1 - X(t_1))}{1+\beta}$ since $x_0 \leq -t$ and $t_0 > t$ (see Figure 1). Now, consider a target starting at initial position p (to be determined) moving at speed $v > 0$ toward a robot which starts at the origin and has a speed of 1. Thus, by placing the target at a starting location so that the farthest *right* the robot could have reached is $x_0 - \epsilon_0$ for any arbitrarily small ϵ_0 , the robot *can not* have reached the target by time t_1 . Such a target has an initial position of

$$p = -\frac{(1+\beta v)(t_1 - X(t_1))}{1+\beta} - \epsilon_0$$

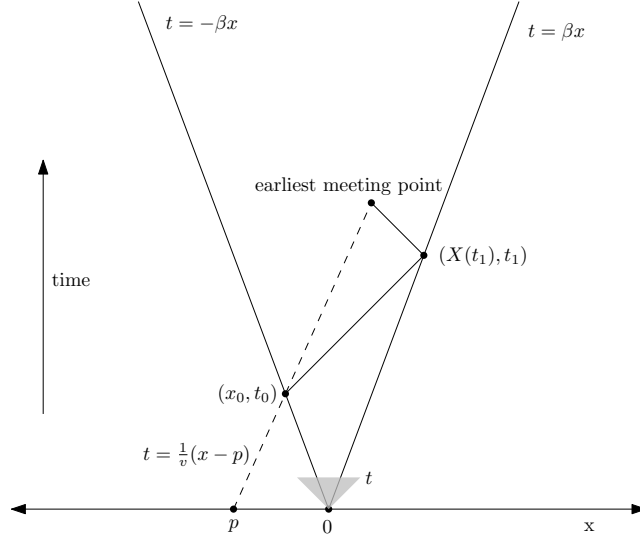
and follows the trajectory

$$X_{\text{target}}(t) = vt + p \quad (8)$$

where $X_{\text{target}}(t)$ denotes the robot's position at time t . Observe also, if the robot moves directly toward the target after t_1 , then its trajectory after time t_1 is given by

$$X(t) = X(t_1) + t_1 - t \quad (9)$$

Thus, the earliest time the robot could possibly encounter the target can be computed by finding the intersection between the robot trajectory (Equation (9)) and the target's



■ **Figure 1** The cone-bounded trajectory of the robot and worst-case placement p of the target. The small gray triangle is to remind the reader that, by the definition of β , the robot trajectory is only guaranteed to be contained by the cone after some finite time t . Thus, in order to maximize the competitive ratio, we (as the adversary) should place the target so that its trajectory does not intersect (x_0, t_0) or the gray triangle.

trajectory (Equation (8)) and solving for t :

$$\begin{aligned}
 vt + p &= X(t_1) + t_1 - t \\
 t &= \frac{X(t_1) + t_1 - p}{1 + v}.
 \end{aligned} \tag{10}$$

Then the competitive ratio (Equation (10) divided by $-p/(1 + v)$, the optimal search time) is

$$\begin{aligned}
 CR &\geq \sup_{\epsilon_0, \epsilon_1} \frac{(X(t_1) + t_1 - p)/(1 + v)}{-p/(1 + v)} = \sup_{\epsilon_0, \epsilon_1} \frac{X(t_1) + t_1 - p}{-p} = \sup_{\epsilon_0, \epsilon_1} \left[1 - \frac{X(t_1) + t_1}{p} \right] \\
 &= \sup_{\epsilon_1} \left[1 + \frac{(1 + \beta)(t_1 + X(t_1))}{(1 + \beta v)(t_1 - X(t_1))} \right]
 \end{aligned} \tag{11}$$

$$= 1 + \frac{(1 + \beta)^2}{(1 + \beta v)(\beta - 1)} \tag{12}$$

where Inequality (11) follows since $p = -\frac{(1 + \beta v)(t_1 - X(t_1))}{1 + \beta} - \epsilon_0$ for arbitrarily small $\epsilon_0 > 0$ and Inequality (12) follows since $X(t_1) = \frac{t_1}{\beta + \epsilon_1}$ for arbitrarily small $\epsilon_1 > 0$. Finally, observe that if $v < \frac{1}{3}$, then the right-hand side of Equality (12) has a single minimum of $1 + 8\frac{1-v}{(1+v)^2}$ at $\beta = \frac{v-3}{3v-1}$. On the other hand, if $v \geq \frac{1}{3}$, then the right-hand side of Equality (12) is decreasing with respect to β and thus the competitive ratio satisfies

$$CR \geq \lim_{\beta \rightarrow \infty} \left[1 + \frac{(1 + \beta)^2}{(1 + \beta v)(\beta - 1)} \right] = 1 + \frac{1}{v}.$$

4 The NoSpeed Model

In this section we assume that the robot knows d but not v .

4.1 The NoSpeed/Away

For this model, it is clear that the robot cannot execute an algorithm like Algorithm 1 since no upper bound on the target's speed is known. Note that, if any upper bound $\hat{v} < 1$ on the target's speed were known, the robot could execute Algorithm 1 by assuming the target speed to be equal to \hat{v} , resulting in a competitive ratio of at most $1 + \frac{2}{1-\hat{v}}$. Since the target speed v is unknown (and potentially very close to 1), however, we propose another strategy. Consider a monotone increasing non-negative integer sequence $\{f_i : i \geq 0\}$ such that $f_0 = 1$ and $f_i < f_{i+1}$, for all $i \geq 0$. The idea of the algorithm is to search for the target by making a guess about its speed in rounds as follows. We start from the origin and alternate searching right and left. On the i -th round, we use the guess $v_i = 1 - 2^{-f_i}$ and search the necessary distance away from the origin such that, if the target's speed is less than or equal to v_i and the target's initial position is in the same direction from the origin that the robot moves in round i , then the target will be found in round i . Otherwise, we can conclude that either the target is moving with a speed greater than v_i or else it is on the opposite side of the origin. In this case the robot returns to the origin and repeats the algorithm in the opposite direction.

Later in the analysis we will show how to select the integer sequence $\{f_i : i \geq 0\}$ so as to obtain bounds on the competitive ratio. The algorithm explained above is formalized as Algorithm 5.

■ **Algorithm 5** Online Algorithm for NoSpeed/Away Model

```

1: input: target initial distance  $d$ 
2:         integer sequence  $\{f_i : i \geq 0\}$  such that  $f_i < f_{i+1}$ , for  $i \geq 0$  and  $f_0 = 1$ ;
3:  $t \leftarrow 0$ 
4: for  $i \leftarrow 0, 1, 2, \dots$  until target found do
5:    $v_i \leftarrow 1 - 2^{-f_i}$ 
6:    $x_i \leftarrow (-1)^i \cdot \frac{d+tv_i}{1-v_i}$ 
7:   move to  $x_i$  and back to the origin
8:    $t \leftarrow t + |x_i|$ 

```

To compensate for the fact that the starting speed of the robot in the algorithm is $v_0 = 1 - 2^{-1} = 1/2$ we first need to consider the case $v \leq \frac{1}{2}$.

► **Lemma 10.** *For the NoSpeed/Away model, if the unknown speed v of the target is less than or equal to $\frac{1}{2}$ then the competitive ratio of Algorithm 5 is at most 5.*

Proof. According to Algorithm 5 and since $v < 1/2$, the robot will find the target either on its first trip away from the origin, after time at most $2d$, or after the first time it changes direction of movement. In the worst case it will spend time $2d$ in one direction and then additional time $\frac{2d+d+2dv}{1-v}$. It follows that the competitive ratio is at most

$$\frac{2d + \frac{2d+d+2dv}{1-v}}{\frac{d}{1-v}} = 5$$

which proves Lemma 10. ◀

Next we analyze the competitive ratio of the algorithm for $v > \frac{1}{2}$.

► **Lemma 11.** *For the NoSpeed/Away model, if the unknown speed v of the target is greater than $\frac{1}{2}$ then the competitive ratio of Algorithm 5 is at most $1 + 2^{1+\sum_{j=0}^k f_j} \cdot 4^{k+1}$ where k is the first k such that $v_k \geq v$.*

Proof. Let d_i be the distance from the origin the target would be if its speed was equal to v_i , where $v_i = 1 - 2^{-f_i}$ at time $\sum_{j=0}^{i-1} (1 - 2^{-f_j})$. In other words, if $v_i \geq v$, then d_i is the maximum distance of the target from the origin (and thus, the robot) at the beginning of round i of the algorithm. Thus, if the speed of the target is less than or equal to v_i and the robot moves toward it in round i , then it would take at most $x_i = \frac{d_i}{1-v_i} = 2^{f_i} d_i$ additional time for the robot to catch up to the target, for $i \geq 0$. Recall the algorithm involves the robot moving a distance x_i (in time x_i , since the robot's speed is 1) away from the origin and back in round i . Observe then that $d_0 = d$, $v_0 = 1/2$, and

$$d_i = d + 2v_i \sum_{j=0}^{i-1} x_j. \quad (13)$$

Therefore, it follows from the definition of x_i that

$$x_i = 2^{f_i} \left(d + 2v_i \sum_{j=0}^{i-1} x_j \right). \quad (14)$$

As a consequence

$$\sum_{j=0}^{i-1} x_j = \frac{x_i - 2^{f_i} d}{2^{f_i} \cdot 2 \cdot v_i} \quad (15)$$

Similarly, if we replace i with $i + 1$ we have that

$$\sum_{j=0}^i x_j = \frac{x_{i+1} - 2^{f_{i+1}} d}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} \quad (16)$$

Subtracting Equation (15) from Equation (16), we derive the recurrence

$$x_i = \frac{x_{i+1} - 2^{f_{i+1}} d}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} - \frac{x_i - 2^{f_i} d}{2^{f_i} \cdot 2 \cdot v_i} \quad (17)$$

Collecting similar terms and simplifying Equation (17) yields

$$\begin{aligned} \frac{x_{i+1}}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} &= \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i + \left(\frac{2^{f_{i+1}}}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} - \frac{2^{f_i}}{2^{f_i} \cdot 2 \cdot v_i} \right) d \\ &= x_i \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) + \frac{d}{2v_{i+1}} - \frac{d}{2v_i} \end{aligned} \quad (18)$$

$$\leq \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i \quad (19)$$

following from the fact the sum of the last two terms in Inequality 18 is less than or equal to 0.

If we simplify the right-hand side of Equation (19), we derive the following recursive inequalities

$$\begin{aligned} x_{i+1} &\leq 2^{f_{i+1}} \cdot 2 \cdot v_{i+1} \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i \\ &\leq 2^{f_{i+1}} \cdot 2 \cdot \left(1 + \frac{1}{2^{f_i}} \right) x_i \\ &\leq (2 \cdot 2^{f_{i+1}} + 2 \cdot 2^{f_{i+1}-f_i}) x_i \\ &\leq 2^{f_{i+1}} \cdot 4 \cdot x_i, \end{aligned} \quad (20)$$

which follows since $\frac{1}{2} \leq v_i < 1$ and $f_i < f_{i+1}$ for all i .

By repeated application of the last Recurrence (20) above and using the fact that by definition $x_0 = 2^{f_0}d$, it follows easily by induction that

$$\begin{aligned}
 x_{i+1} &\leq 2^{f_{i+1}} \cdot 4 \cdot x_i \\
 &\leq 2^{f_{i+1}+f_i} \cdot 4^2 \cdot x_{i-1} \\
 &\vdots \\
 &\leq 2^{f_{i+1}+f_i+f_{i-1}+\dots+f_1} \cdot 4^{i+1} \cdot x_0 \\
 &\leq 2^{\sum_{j=0}^{i+1} f_j} \cdot 4^{i+1} \cdot d
 \end{aligned} \tag{21}$$

Consider the first i such that $v_i \geq v$. It follows that $v_{i-1} < v$ which yields $1 - v < 1 - v_{i-1} = 2^{-f_{i-1}}$ and implies that $2^{f_{i-1}} < \frac{1}{1-v}$. Note that although $v_i \geq v$, the robot may not find the target in round i because it is located in the opposite direction. It is guaranteed, however, to find the target by round $i + 1$. Moreover the total time that has elapsed from the start until round i is $2 \sum_{j=0}^i x_j$ at which time the target is at distance $d + v2 \sum_{j=0}^i x_j$ from the origin.

As a consequence the competitive ratio of Algorithm 5 is at most

$$\begin{aligned}
 \frac{2 \sum_{j=0}^i x_j + \frac{d+2v \sum_{j=0}^i x_j}{1-v}}{\frac{d}{1-v}} &= 1 + \frac{2(v+1-v)}{d} \sum_{j=0}^i x_j \\
 &= 1 + \frac{x_{i+1} - 2^{f_{i+1}}d}{d} \cdot \frac{2}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} \quad (\text{By (16)}) \\
 &\leq 1 + \frac{x_{i+1}}{d} \cdot \frac{2}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} \\
 &\leq 1 + \frac{1}{v_{i+1}} 2^{\sum_{j=0}^i f_j} \cdot 4^{i+1} \quad (\text{By (21)})
 \end{aligned}$$

Since $v_{i+1} \geq 1/2$ we conclude with an upper bound on the competitive ratio of Algorithm 5 of

$$1 + 2^{1+\sum_{j=0}^i f_j} \cdot 4^{i+1} \tag{22}$$

which proves Lemma 11. ◀

We are now ready to prove the main theorem about the competitive ratio of Algorithm 5.

► **Theorem 12.** *For the NoSpeed/Away model, the competitive ratio of Algorithm 5 when applied to the sequence $f_j = 2^j$, for all $j \geq 0$, is at most*

$$\begin{cases} 5 & \text{if } v \leq \frac{1}{2} \\ 1 + \frac{16(\log \frac{1}{1-v})^2}{(1-v)^4} & \text{otherwise} \end{cases}$$

where \log is the base-2 logarithm.

Proof. Consider the first index i such that $v_i \geq v$. It follows that $v_{i-1} < v$, and so

$$1 - 2^{-2^{i-1}} < v \Rightarrow 2^{2^{i-1}} < \frac{1}{1-v}.$$

Then, by Lemma 11, the competitive ratio of is at most

$$\begin{aligned} 1 + 2^{1+\sum_{j=0}^i f_j} \cdot 4^{i+1} &= 1 + 2^{1+\sum_{j=0}^i 2^j} \cdot 4^{i+1} \\ &= 1 + 2^{2^{i+1}} \cdot 4^{i+1} \leq 1 + \left(\frac{1}{1-v}\right)^4 \cdot \left(4 \cdot \log\left(\frac{1}{1-v}\right)\right)^2 \\ &\leq 1 + \frac{16 \left(\log \frac{1}{1-v}\right)^2}{(1-v)^4} \end{aligned}$$

This proves Theorem 12. ◀

► **Remark 13.** By Theorem 1, $1 + \frac{2}{1-v}$ is a lower bound on any algorithm when both v, d are known. As a consequence it must also be a lower bound when d is known but v is not.

4.2 The NoSpeed/Toward Model

Now we consider the case where the target is moving toward the origin.

■ **Algorithm 6** Online Algorithm for NoSpeed/Toward Model

-
- 1: **input:** target initial distance d
 - 2: choose any direction and go for time d
 - 3: **if** target not found **then**
 - 4: change direction and go until target is found
-

► **Theorem 14.** *For the NoSpeed/Toward model, Algorithm 6 achieves an optimal competitive ratio at most 3.*

Proof. The robot chooses a direction (without loss of generality, say to the right) and goes for a time d (this is where the robot makes use of its knowledge of the distance d). If it does not find the target it changes direction. In the meantime the target has moved for a distance dv and now must be at location $-d + dv$. Therefore at the time the robot changes direction the distance between robot and target is equal to $d - (-d + dv) = 2d - dv$, and hence the meeting will take place in additional time $\frac{2d-dv}{1+v}$. It follows that the total time required for the robot to meet the target must be equal to $d + \frac{2d-dv}{1+v}$. The resulting competitive ratio satisfies

$$CR \leq \frac{d + \frac{2d-dv}{1+v}}{\frac{d}{v+1}} = 3.$$

This proves the upper bound.

To prove the lower bound we argue as follows. If the searcher never visits either of the points $\pm d$ then the competitive ratio is arbitrarily large for very small values of v . Let $\epsilon > 0$ be sufficiently small and let the speed of the target be $v = \epsilon/3$. Consider the first time, say t , that the robot reaches one of the points $\pm(d - \epsilon)$. Without loss of generality let this point be $d - \epsilon$ and suppose the target is adversarially placed at $-d$. Then at time t it will be located at $-d + tv$. Therefore the distance between the robot and the target at time t will be $d - \epsilon - (-d + tv) = 2d - tv - \epsilon$. The time it takes for robot to find the target, then, is at least $d - \epsilon + \frac{2d-tv-\epsilon}{1+v}$ and the competitive ratio is at least

$$\frac{d - \epsilon + \frac{2d-tv-\epsilon}{1+v}}{\frac{d}{1+v}} \geq 3 - \frac{2\epsilon + (t + \epsilon)v}{d}$$

It follows easily that if $t \geq 3d - \epsilon$ then $CR \geq \frac{t}{d/(1+v)} \geq 3 - 3\epsilon$. However, if $t \leq 3d - \epsilon$ then $\frac{2\epsilon + (t+\epsilon)v}{d} \leq 2\epsilon + 3v \leq 3\epsilon$, since by assumption $v = \epsilon/3$. Therefore again $CR \geq 3 - 3\epsilon$. This completes the proof of Theorem 14. ◀

5 The NoKnowledge Model

In this section we assume that neither the initial distance d nor the speed v of the target is known to the robot.

5.1 The NoKnowledge/Away Model

We now describe an approximation strategy resembling that described in Section 4.1. For this strategy though, the robot will need to guess both the target's speed *and* its initial distance.

Consider the situation where neither the distance d to the target nor its speed $v < 1$ is known to the robot. Also consider two monotone increasing non-negative integer sequences $\{f_i, g_i : i \geq 0\}$ such that $f_0 = 1$ and $g_0 = 0$ and $f_i < f_{i+1}$ and $g_i < g_{i+1}$, for all $i \geq 0$. The idea of the algorithm is to search for the target by making a guess for its speed and starting distance in rounds as follows. The robot, starting from the origin, alternates searching to the right and left. On the i -th round, it guesses that the target's speed does not exceed $v_i = 1 - 2^{-f_i}$ and that its initial distance from the origin does not exceed 2^{g_i} . Using these guesses, the robot searches exactly the distance required (which we will later denote d_i) to catch the target, given its guesses are correct *and* that the target is in the direction the robot searches in round i . If robot does not find the target after searching this distance, it returns to the origin and begins the next round. Later in the analysis we will show how to select the integer sequences $\{f_i, g_i : i \geq 0\}$ so as to obtain bounds on the competitive ratio. We formalize the algorithm described above as Algorithm 7.

Algorithm 7 Online Algorithm for NoKnowledge/Away Model

-
- 1: **Inputs;** Integer sequences $\{f_i, g_i : i \geq 0\}$ such that $f_i < f_{i+1}$ and $g_i < g_{i+1}$, for $i \geq 0$ and $f_0 = 1$ and $g_0 = 0$;
 - 2: $t \leftarrow 0$
 - 3: **for** $i \leftarrow 0, 1, 2, \dots$ until target found **do**
 - 4: $d_i \leftarrow 2^{g_i}$
 - 5: $v_i \leftarrow 1 - 2^{-f_i}$
 - 6: $x_i \leftarrow (-1)^i \cdot \frac{d_i + tv_i}{1 - v_i}$
 - 7: move to x_i and back to the origin
 - 8: $t \leftarrow t + |x_i|$
-

Since there is always an integer $i \geq 1$ such that both $v_i = 1 - 2^{-f_i} \geq v$ and $2^{g_i} \geq d$, it is clear that the robot will eventually succeed in catching the target. Next we analyze the competitive ratio of the algorithm.

► **Lemma 15.** *For the NoKnowledge/Away model, if Algorithm 7 terminates successfully in round $i + 1$ then its competitive ratio must satisfy*

$$CR \leq 1 + \frac{2(i+2)}{d} \cdot 2^{g_{i+1}} \cdot 2^{\sum_{j=0}^i f_j} \cdot 4^{i+1}. \quad (23)$$

Proof. We call each iteration of the loop in Algorithm 7 a *round*. For any round i , let d_i be the distance from the origin to where the target would be if its speed was equal to $v_i = 1 - 2^{-f_i}$ and its starting position 2^{g_i} . Recall that during the first $i - 1$ unsuccessful rounds, the target is moving further and further away from the origin. If the robot is at the origin and the speed of the target is v_i then it takes time at most $x_i = \frac{d_i}{1-v_i} = 2^{f_i} d_i$ for the robot to catch up to the target, for $i \geq 0$. Observe from the algorithm that $d_0 = 1$ and $v_0 = 1/2$ and

$$d_i = 2^{g_i} + 2v_i \sum_{j=0}^{i-1} x_j. \quad (24)$$

Therefore, it follows from the definition of x_i that

$$x_i = 2^{f_i} \left(2^{g_i} + 2v_i \sum_{j=0}^{i-1} x_j \right). \quad (25)$$

As a consequence

$$\sum_{j=0}^{i-1} x_j = \frac{x_i - 2^{f_i+g_i}}{2^{f_i} \cdot 2 \cdot v_i} \quad (26)$$

Similarly, if we replace i with $i + 1$ we have that

$$\sum_{j=0}^i x_j = \frac{x_{i+1} - 2^{f_{i+1}+g_{i+1}}}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}}. \quad (27)$$

Subtracting Equation (26) from Equation (27) we derive the recurrence

$$x_i = \frac{x_{i+1} - 2^{f_{i+1}+g_{i+1}}}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} - \frac{x_i - 2^{f_i+g_i}}{2^{f_i} \cdot 2 \cdot v_i} \quad (28)$$

Collecting similar terms and simplifying Equation (28) yields

$$\begin{aligned} \frac{x_{i+1}}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} &= \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i + \left(\frac{2^{f_{i+1}+g_{i+1}}}{2^{f_{i+1}} \cdot 2 \cdot v_{i+1}} - \frac{2^{f_i+g_i}}{2^{f_i} \cdot 2 \cdot v_i} \right) \\ &= \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i + 2^{g_{i+1}-1} \left(\frac{1}{v_{i+1}} - \frac{2^{g_i}}{2^{g_{i+1}} \cdot v_i} \right) \\ &\leq \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i + 2^{g_{i+1}-1} \end{aligned} \quad (29)$$

where Inequality (29) follows since $\frac{1}{v_{i+1}} - \frac{2^{g_i}}{2^{g_{i+1}} \cdot v_i} \leq 1$.

If we multiply out with the denominator in the lefthand side of Inequality (29) and simplify the righthand side we derive the following recursive inequalities

$$\begin{aligned} x_{i+1} &\leq 2^{f_{i+1}} \cdot 2 \cdot v_{i+1} \left(1 + \frac{1}{2^{f_i} \cdot 2 \cdot v_i} \right) x_i + 2^{f_{i+1}+g_{i+1}} v_{i+1} \\ &\leq \left(2(2^{f_{i+1}} - 1) + 2^{f_{i+1}-f_i} \cdot \frac{v_{i+1}}{v_i} \right) x_i + 2^{f_{i+1}+g_{i+1}} \end{aligned} \quad (30)$$

$$\leq 2^{f_{i+1}} \cdot 4 \cdot x_i + 2^{f_{i+1}+g_{i+1}}, \quad (31)$$

where in the derivation of Inequality (31) from the previous Inequality (30) we used the fact that $\frac{v_{i+1}}{v_i} \leq 2$.

By repeated application of the last Recurrence (31) above and using the fact that by definition $x_0 = 2^{f_0}d$, it follows easily by induction that

$$\begin{aligned}
 x_{i+1} &\leq 2^{f_{i+1}} \cdot 4 \cdot x_i + 2^{f_{i+1}+g_{i+1}} \\
 &\leq 2^{f_{i+1}+f_i} \cdot 4^2 \cdot x_{i-1} + 2^{f_{i+1}+f_i+g_i} \cdot 4^1 + 2^{f_{i+1}+g_{i+1}} \\
 &\quad \vdots \\
 &\leq 2^{g_0+\sum_{j=0}^{i+1} f_j} \cdot 4^{i+1} \cdot x_0 + 2^{g_1+\sum_{j=1}^{i+1} f_j} \cdot 4^i + 2^{g_2+\sum_{j=2}^{i+1} f_j} \cdot 4^{i-1} \\
 &\quad + \dots + 2^{f_{i+1}+g_{i+1}} \\
 &= \sum_{k=0}^{i+1} 2^{g_k+\sum_{j=k}^{i+1} f_j} \cdot 4^{i-k+1}
 \end{aligned} \tag{32}$$

since $x_0 = 1$.

The total time that has elapsed from the start until the beginning of last round i (when the robot visits the origin for the last time before catching the target) will be $\sum_{j=0}^i 2x_j$ at which time the target is at distance $d + v \sum_{j=0}^i 2x_j$ from the origin. As a consequence the competitive ratio of Algorithm 7 must satisfy the inequality

$$CR \leq \frac{2 \sum_{j=0}^i x_j + \frac{d+2v \sum_{j=0}^i x_j}{1-v}}{\frac{d}{1-v}}. \tag{33}$$

Simplifying the righthand side of Inequality (33) and using Identity (26) yields

$$\begin{aligned}
 CR &\leq 1 + \frac{2}{d} \sum_{j=0}^i x_j \\
 &\leq 1 + \frac{x_{i+1}}{d} \cdot \frac{1}{2^{f_{i+1}} \cdot v_{i+1}} \text{ (Use Equation (26))} \\
 &\leq 1 + \frac{1}{v_{i+1}d2^{f_{i+1}}} \sum_{k=0}^{i+1} 2^{g_k+\sum_{j=k}^{i+1} f_j} \cdot 4^{i-k+1} \text{ (Use Equation (32))} \\
 &\leq 1 + \frac{1}{v_{i+1}d} \sum_{k=0}^{i+1} 2^{g_k+\sum_{j=k}^i f_j} \cdot 4^{i-k+1}.
 \end{aligned}$$

Since $v_{i+1} \geq 1/2$ we conclude with

$$\begin{aligned}
 CR &\leq 1 + \frac{2}{d} \sum_{k=0}^{i+1} 2^{g_k+\sum_{j=k}^i f_j} \cdot 4^{i-k+1} \\
 &\leq 1 + \frac{2}{d} \sum_{k=0}^{i+1} 2^{g_k+\sum_{j=k}^i f_j} \cdot 4^{i-k+1} \\
 &\leq 1 + \frac{2(i+2)}{d} \cdot 2^{g_{i+1}} \cdot 2^{\sum_{j=0}^i f_j} \cdot 4^{i+1}
 \end{aligned} \tag{34}$$

This completes the proof of Lemma 15. ◀

We now prove the following theorem.

► **Theorem 16.** *For the NoKnowledge/Away model, Algorithm 7 with the sequences $g_i = f_i = 2^i$ has a competitive ratio of at most*

$$1 + \frac{16}{d} \left[\log \log \max \left(d, \frac{1}{1-v} \right) + 3 \right] \cdot \max \left(d, \frac{1}{1-v} \right)^8 \cdot \log^2 \max \left(d, \frac{1}{1-v} \right)$$

where \log is the base-2 logarithm.

Proof. Observe that if the robot finds the target in round $i + 1$, then by design, one or both of the robot's round $i - 1$ guesses for the target's speed ($1 - 2^{-2^{i-1}}$) or initial distance ($2^{2^{i-1}}$) must have been too low, otherwise the robot would have found the target in an earlier round. In other words, either $1 - 2^{-2^{i-1}} < v$ or $2^{2^{i-1}} < d$. It follows, then that $i - 1 < \log \log \max \left(d, \frac{1}{1-v} \right)$. Then by Lemma 15, an upper bound on the competitive ratio is given by

$$\begin{aligned} CR &\leq 1 + \frac{2(i+2)}{d} \cdot 2^{g_{i+1}} \cdot 2^{\sum_{j=0}^i f_j} \cdot 4^{i+1} \\ &= 1 + \frac{2(i+2)}{d} \cdot 2^{2^{i+1}} \cdot 2^{2^{i+1}-1} \cdot 4^{i+1} \\ &= 1 + \frac{(i-1)+3}{d} \cdot \left(2^{2^{i-1}} \right)^8 \cdot 16 \left(2^{i-1} \right)^2 \\ &= 1 + \frac{16}{d} \left[\log \log \max \left(d, \frac{1}{1-v} \right) + 3 \right] \cdot \max \left(d, \frac{1}{1-v} \right)^8 \cdot \log^2 \max \left(d, \frac{1}{1-v} \right) \end{aligned}$$

which proves Theorem 16. ◀

► **Remark 17.** Observe that a lower bound of $1 + 8 \frac{1+v}{(1-v)^2}$ follows directly from the NoDistance/Away model.

5.2 The NoKnowledge/Toward Model

We can prove the following theorem.

► **Theorem 18.** *The optimal competitive ratio is $1 + \frac{1}{v}$ and is given by the waiting Algorithm.*

Proof. The upper bound follows directly from Theorem 4. For the lower bound, consider an algorithm where the robot does not wait forever and instead moves a distance $d' > 0$ to the right (without loss of generality – a symmetric argument for the case where the robot moves to the left follows trivially) after waiting at the origin for time $t \geq 0$. Then consider the scenario where the target with speed $v = \frac{d}{t+d'}$ is initially at $-d$ for any distance $d \geq 1$. Thus, the target reaches the origin at exactly the time the robot reaches d' and so their earliest possible meeting time is

$$t + d' + \frac{d'}{1+v} = \frac{d}{v} + \frac{d'}{1+v} \geq \frac{d}{v}$$

Thus, the competitive ratio is at least

$$\frac{d/v}{d/(1+v)} = 1 + \frac{1}{v}$$

This proves Theorem 18. ◀

6 Conclusion

We considered linear search for an autonomous robot searching for an oblivious moving target on an infinite line. Two scenarios were analyzed depending on whether the target is moving towards or away from the origin (and this is known to the robot). In either of these two scenarios we considered the knowledge the robot has about the speed and starting distance of the target. For each scenario we gave search algorithms and analyzed their competitive ratio for the four possible cases arising. Our bounds are tight in all cases when the target is moving towards the origin. They are also shown to be tight when the target is moving away from the origin and its speed $0 < v < 1$ is known to the robot; for this scenario we also obtain upper bounds when v is not known to the robot. It remains an open problem to prove tight bounds for the case when v is unknown to the robot and the target is moving away from the origin. It also remains open to find tight bounds for the case where the direction of movement of the target is unknown.

References

- 1 R. Ahlswede and I. Wegener. *Search problems*. Wiley-Interscience, 1987.
- 2 Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*, volume 55 of *International series in operations research and management science*. Kluwer, 2003.
- 3 Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Rawlins. Searching in the plane. *Inf. Comput.*, 106(2):234–252, 1993. doi:10.1006/inco.1993.1054.
- 4 A. Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.
- 5 R. Bellman. An optimal search. *Siam Review*, 5(3):274–274, 1963.
- 6 A. Bonato and R. Nowakowski. *The game of cops and robbers on graphs*. American Mathematical Soc., 2011.
- 7 Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil M. Shende. Search on a line by byzantine robots. *Int. J. Found. Comput. Sci.*, 32(4):369–387, 2021. doi:10.1142/S0129054121500209.
- 8 Jurek Czyzowicz, Ryan Killick, and Evangelos Kranakis. Linear rendezvous with asymmetric clocks. In Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira, editors, *22nd International Conference on Principles of Distributed Systems, OPODIS 2018, December 17-19, 2018, Hong Kong, China*, volume 125 of *LIPICs*, pages 25:1–25:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.OPODIS.2018.25.
- 9 Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, and Jaroslav Opatrny. Search on a line with faulty robots. *Distributed Comput.*, 32(6):493–504, 2019. doi:10.1007/s00446-017-0296-0.
- 10 Ryan Killick. Lower bound for linear search. Unpublished Manuscript, 2022.
- 11 Gianluca De Marco, Luisa Gargano, Evangelos Kranakis, Danny Krizanc, Andrzej Pelc, and Ugo Vaccaro. Asynchronous deterministic rendezvous in graphs. *Theor. Comput. Sci.*, 355(3):315–326, 2006. doi:10.1016/j.tcs.2005.12.016.
- 12 B. J. McCabe. Searching for a one-dimensional random walker. *J. Applied Probability*, pages 86–93, 1974.
- 13 Sven Schuierer. Lower bounds in on-line geometric searching. *Comput. Geom.*, 18(1):37–53, 2001. doi:10.1016/S0925-7721(00)00030-4.
- 14 L. Stone. *Theory of optimal search*. Academic Press New York, 1975.