Linear Rendezvous with Asymmetric Clocks

Jurek Czyzowicz[1]
Département d'informatique, Université du Québec en Outaouais, Gatineau, Canada

Ryan Killick[2]
School of Computer Science, Carleton University, Ottawa, Canada

Evangelos Kranakis[3]
School of Computer Science, Carleton University, Ottawa, Canada

Abstract

Two anonymous robots placed at different positions on an infinite line need to rendezvous. Each robot possesses a clock which it uses to time its movement. However, the robot's individual parameters in the form of their walking speed and time unit may or may not be the same for both robots. We study the feasibility of rendezvous in different scenarios, in which some subsets of these parameters are not the same. As the robots are anonymous, they execute the same algorithm and when both parameters are identical the rendezvous is infeasible. We propose a universal algorithm, such that the robots are assured of meeting in finite time, in any case when at least one of the parameters is not equal for both robots.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms, Theory of computation → Distributed algorithms

Keywords and phrases anonymous, asymmetric clock, infinite line, rendezvous, mobile robot, speed, competitive ratio

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2018.25

1 Introduction

Rendezvous is concerned with two robots arbitrarily placed in a known search region and moving about until they meet each other. In this paper we will study symmetric rendezvous in which the two robots are instructed to employ the same algorithm. In our setting the robots' environment is an infinite line on which each robot may move at a constant speed. Each robot is equipped with its own clock which is used to time its movements and this clock is not necessarily consistent between the robots.

The rendezvous problem was studied for numerous models and various types of environments in randomized as well as deterministic settings. The fundamental question related to deterministic rendezvous concerns feasibility, or, more exactly, to identify the parameters of the model for which the rendezvous is possible to achieve (in finite time). The main concern related to the feasibility of rendezvous is that of *symmetry breaking*. Typical example of symmetry breaking is the use of the robot label in which the robot is aware of its label and may use its value as a parameter.

In the present paper the symmetry is broken yet another way. If the robots differ according to their speeds or private time units, we have a universal algorithm which guarantees rendezvous. Furthermore, and contrary to the case of labeled robots, knowledge as to which

[1] Research supported in part by NSERC Discovery grant
[2] Research supported by the Ontario Graduate Scholarship. Eligible for best student paper award.
[3] Research supported in part by NSERC Discovery grant

22nd International Conference on Principles of Distributed Systems (OPODIS 2018).
Editors: Jiannong Cao, Faith Ellen, Luis Rodrigues, and Bernardo Ferreira; Article No. 25; pp. 25:1–25:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of the parameters is different is not necessary. Our robot is completely unaware of the value(s) of its individual parameters and it does not use them in the computations needed to run the algorithm.

When the rendezvous is feasible, research is concerned with the efficiency of the algorithm, which is usually measured by the time required until the meeting of the two robots takes place. The objective is to design algorithms that achieve good *competitive ratios* for the time spent by the robots to rendezvous divided by the time spent by the robots if they were running an optimal algorithm.

1.1 Model

We consider the symmetric rendezvous problem of two mobile robots $\mathcal{R}$ and $\mathcal{R}'$ modeled as points on the infinite line. The robots are initially located an unknown distance d from each other and the rendezvous problem is solved if it ever happens that the robots occupy the same position on the line at the same time (i.e. their trajectories intersect). The robots cannot see each other and must employ the same algorithm in order to rendezvous. We assume that robots can store and compute real numbers with arbitrary precision.

We consider a model in which each robot has its own constant speed and in which each is equipped with a clock allowing them to measure their travel time. Each robot will consider itself as the origin of its own coordinate system and it will use its clock to fix the distance unit for this coordinate system as the product of its maximum speed and local time unit. We explicitly consider the possibility that the robots have different speeds and / or clocks. We study algorithms which progress in a synchronous and continuous time model (i.e. robots are always active).

Without loss of generality, we will present our analysis from the viewpoint of the robot $\mathcal{R}$ and thus assume that this robot has maximum unit speed, and that its clock is "correct" in the sense that it agrees with some predefined global coordinate system. On the other hand, we set the speed of $\mathcal{R}'$ as $v > 0$, and set its time unit as $\tau > 0$ with the result that one time unit as measured by the clock of $\mathcal{R}'$ will actually be τ time units as measured by the clock of $\mathcal{R}$. The robots will determine their progress/distance traveled in an algorithm as the product of their travel time and maximum speed.

We specifically focus on three sub-models obtained from the preceding general model by fixing one of v, τ, or the product $v\tau$ to one. In the equal time-unit model (or T-Model) $\tau = 1$, in the equal distance-unit model (or D-Model) $v\tau = 1$, and in the equal speeds model (or V-Model) $v = 1$. Since only one of v or τ is independent in these models we will assume without loss of generality that $0 < v < 1$ in the T- and D-Models and take $0 < \tau < 1$ in the V-Model.

In analyzing the time complexity of our rendezvous algorithms we will employ an adversarial argument in which we assume that an adversary is able to choose the values of d, v, and/or τ in order to maximize the *competitive ratio* of a given algorithm (i.e. we employ a worst-case analysis). Loosely defined, the competitive ratio of a rendezvous algorithm $\mathcal{A}$ is the maximum ratio of the time it takes the robots to rendezvous using $\mathcal{A}$ divided by the time it would take them given that they are running an optimal algorithm. We will give a more precise definition of the competitive ratio at a later time.

To end this section we observe that to be most general one should consider the possibility that the robots also differ in their orientations (i.e. sense of positive direction). However, in all cases for rendezvous on the line studied here, having different orientations will only help the robots in achieving their goal. As a result, all derived upper bounds will not be affected by a difference in orientation and, since including this only serves to complicate notation, we will not explicitly consider it in this work.

1.2 Related Work

Search may be viewed as a game between two players having divergent goals, one trying to hide for as long as possible and the other one attempting to minimize its search time. As a contrast, in rendezvous the two involved players have converging goals in that they are aiming to find one another as quickly as possible. The duality of the two problems has been presented and investigated in the beautiful book [1].

The search problem for one robot on an infinite line was initiated independently by Bellman [7] and Beck [5]. There have been numerous studies on search even for an environment as simple as the infinite line, emphasizing various aspects arising from the capabilities of one or more robots and the status of the search domain. These include randomized [23], group search [10], linear terrains [14], faulty searchers [15], and turn costs [17]. The efficiency of linear search is most often measured by the *competitive ratio*, which is the time spent by the robot to complete its search divided by the time needed by an omniscient robot that knows the location of the target. The competitive ratio of 9 is obtained by the *cow-path* algorithm (cf. [7] and Beck [5]) and was first proved to be optimal for stochastic linear search in [6] and deterministically in [2, 3].

The approach to solving the rendezvous problem is most often fundamentally different from the techniques applied for search (although, somewhat surprisingly, this will not always be the case in the present paper). The solution of the symmetric rendezvous problem requires that the two robots are somehow equipped to break symmetry [25]. There has been extensive research literature concerned with taking advantage of "innate" asymmetries of the studied model. For example, [16, 26, 18, 19] focus on robots having distinct labels, [24, 27] on robots equipped with identical tokens that can be placed on selected nodes, [12] on a robot's awareness of its GPS position in the environment.

The rendezvous (and its more general version of gathering) problem has been also studied for robots of different speeds [8, 20], inconsistent compasses [11, 22] and chirality or sense of direction [4, 9]. However, in the studies previously mentioned, these differences were obstacles that needed to be circumvented by the suggested algorithms, rather than used for the benefit of the proposed approach, which is the case of the present paper. To the best of our knowledge, linear rendezvous for robots with asymmetric clocks has never been studied before.

1.3 Results and outline

In this paper we study the rendezvous problem when the robots are equipped with asymmetric clocks. In addition to exploring novel ways of defining rendezvous algorithms, we demonstrate the feasibility of rendezvous and give two algorithms solving rendezvous in finite time provided that at least one of a robot's maximum speed or time unit differs from the other (see end of Section 5). In addition, we study rendezvous in the three restricted models, T-model, D-model, and V-model.

In Section 3 we analyze rendezvous in the T-Model and show that possessing equal time-units reduces rendezvous into the problem of search, allowing one to optimally solve the problem with a competitive ratio of 9. In Section 4 we analyze rendezvous under the assumption that the robots have equal distance-units, i.e. $v\tau = 1$. Here we get our first taste of the difficulties involved with asymmetric clocks. We show that rendezvous is solved with a competitive ratio of $\frac{105}{11} \approx 9.55$. Furthermore, in the limit of large d, the competitive ratio is 9.

In Section 5 we analyze rendezvous when the robots' speeds are equal and $0 < \tau < 1$. This is the most difficult model to analyze and we will observe large differences in the algorithms employed and resulting upper bounds, as compared to the T- and D-Models. We give two algorithms that solve rendezvous, the first with a competitive ratio of $\mathcal{O}\left(\frac{\tau \log^2(d)}{\log(\tau)}\right)$ and the second with a slightly tunable competitive ratio of $\mathcal{O}\left(\frac{\tau \log^{1+c}(d)}{c \log(\tau)}\right)$, where $c > 0$ is a parameter of the algorithm. We also offer arguments as to why it may not be possible to achieve a constant competitive ratio in this model.

2 Preliminaries and Notation

In this section we introduce some preliminary ideas and notations used throughout our analysis. We begin with some notation.

As usual, for any real number we use $|\cdot|$ to indicate its absolute value, $\log(\cdot)$ to indicate the base-2 logarithm, and $\ln(\cdot)$ to indicate the natural logarithm with base e.

2.1 Time and position space

It will be useful to consider rendezvous algorithms as specifying a piecewise-continuous trajectory in a two dimensional space where the horizontal axis represents a robot's position on the line and the vertical axis represents the flow of time. In this representation one can view the effect of v, τ, and d as a scaling and translation of the coordinate axes of the local xt-space of $\mathcal{R}'$ as compared to $\mathcal{R}$. As a result, if we represent the trajectory $t(x)$ of $\mathcal{R}$ as a (possibly multivalued) function of position on the line, then the actual trajectory of $\mathcal{R}'$ will be $\tau \cdot t\left(\frac{x \pm d}{v\tau}\right)$. Equivalently, if the trajectory of $\mathcal{R}$ is represented as a function of time $x(t)$, then the trajectory of $\mathcal{R}'$ is $v\tau x\left(\frac{t}{\tau}\right) \pm d$. It would help to remember these transformations as they will be used repeatedly throughout our analysis.

2.2 Rendezvous algorithms

In what follows assume that we are speaking about the robot $\mathcal{R}$.

We consider algorithms whereby robots move between an infinite sequence of *turning-points* $P_k = (X_k,\ T_k)$ which specify the time and location on the line at which a robot reverses its direction. On the round k of such an algorithm a robot will move from its initial position to the turning-point P_k and then return to its initial position. The time necessary to finish a round will be $2|X_k|$ and the time at which a robot begins the round k is $2\sum_{i=0}^{k-1}|X_i|$. The time T_k at which a robot is at the turning-point P_k can thus be written as

$$T_k = 2\sum_{i=0}^{k-1}|X_i| + |X_k|. \tag{1}$$

Since T_k is dependent on X_k we will often refer to X_k as the k^{th} turning-point. We will also adopt the convention that P_{-1} indicates a robot's starting location and assume without loss of generality that $X_k > 0$ when k is even (i.e. a robot initially moves to the right). Finally, we will always explicitly state whether or not $\mathcal{R}'$ begins on the left or right of $\mathcal{R}$ and assume that $d > 0$.

Using the same terminology as [1] we call an infinite sequence of turning-points periodic and monotonic if, for all $k \geq 0$, it satisfies

$$X_{2k-1} < X_{2k+1} < \cdots < 0 < \cdots < X_{2k} < X_{2k+2}. \tag{2}$$

Algorithm 1 Rendezvous[X_k].

1: $k = 0$
2: **repeat**
3: Move to the position X_k and return to initial position. $k = k + 1$.
4: **until** Meeting occurs

As discussed[4] in [1], an adversary may achieve an arbitrarily large competitive ratio if an algorithm ever has a first turning-point. This results from the ability of an adversary to choose d arbitrarily small with respect to this first turning-point and force $\mathcal{R}$ to initially move away from the initial position of $\mathcal{R}'$. As a result, one either needs to consider doubly infinite sequences of turning-points or, more practically, make the additional assumption that the robots possess knowledge of a lower bound on d (which might be indirectly due to the robot having a finite size or visibility range). In this work we are primarily interested in large values of d and / or v and τ close to one, we will take the latter approach and assume that the adversary is restricted to values of d that are comparable in size to the first turning-point of a rendezvous algorithm.

We observe that any periodic and monotonic algorithm will have turning-points that span a cone-shaped curve $\mathcal{T}(x)$ in xt-space satisfying the identity $T_k = \mathcal{T}(X_k)$ (i.e. the turning points lie on the boundary of a cone-shaped curve). This observation allows us to give an alternate, and particularly useful representation of a rendezvous algorithm – we specify the curve $\mathcal{T}(x)$ and have the robots infer where their turning-points are located. Using this method a robot will be instructed to move in one direction until its trajectory in xt-space intersects the curve $\mathcal{T}(x)$. It will then reverse its direction and move at full speed until its trajectory again intersects $\mathcal{T}(x)$, and so on. The robots can compute their turning-points for this type of algorithm using the relation $T_k = \mathcal{T}(X_k)$.

If an algorithm is given by its turning-points X_k then we say that X_k induces the curve $\mathcal{T}(x)$. If an algorithm is specified by $\mathcal{T}(x)$ then we say that $\mathcal{T}(x)$ induces the turning-points X_k. We call an algorithm symmetric if the curve $\mathcal{T}(x)$ is an even function of x. We will be interested in algorithms that are symmetric, periodic, and monotonic and we call SPM the class of all such algorithms. We will use the following formal definition, based on $\mathcal{T}(x)$, for an algorithm to be in the class SPM:

▶ **Definition 1.** An algorithm with turning-points X_k is in the class SPM if the curve $\mathcal{T}(x)$ which induces these turning-points satisfies:

1. $\mathcal{T}(x)$ is an even function, i.e. $\mathcal{T}(x) = \mathcal{T}(-x)$.
2. There exists an $X_0 > 0$ such that $X_0 = \mathcal{T}(X_0)$.
3. For all $|x| > X_0$, $\mathcal{T}(x) > |x|$.
4. $\mathcal{T}(x)$ is continuously differentiable for all $x > X_0$.

One can easily confirm that if $\mathcal{T}(x)$ satisfies the above definition, the induced turning-points will be periodic and monotonic (as per the condition (2)). In the sequel we will construct algorithms which belong to the class SPM.

We formally define Algorithm 1 and Algorithm 2 which respectively take X_k and $\mathcal{T}(x)$ as parameters. So far we have only discussed rendezvous algorithms from the perspective of $\mathcal{R}$. When referring to the turning-points etc. of $\mathcal{R}'$ we will indicate this using a prime. So,

[4] Technically this was discussed for the case of search but the argument also applies here.

Algorithm 2 Rendezvous[$\mathcal{T}(x)$].

1: Run Algorithm 1 with X_k defined by $T_k = \mathcal{T}(X_k)$.

for example, the turning-points of $\mathcal{R}'$ will be $P'_k = (X'_k,\ T'_k)$ and the curve induced by these turning-points will be $\mathcal{T}'(x)$. We note that $X'_k = v\tau X_k \pm d$, $T'_k = \tau T_k$, and $\mathcal{T}'(x) = \tau\mathcal{T}\left(\frac{x \mp d}{v\tau}\right)$.

2.3 Robot trajectories and rendezvous points

Viewed as a whole, a rendezvous algorithm will specify a trajectory in xt-space composed of a series of line segments connected at their endpoints. For an algorithm defined by the sequence X_k we can explicitly write the equations of the lines traversed by $\mathcal{R}$ as

$$t_k(x) = (-1)^k x + T_k - |X_k|. \tag{3}$$

where $t_k(x)$ is the segment beginning at P_{k-1} and ending at P_k. Likewise, the lines traversed by $\mathcal{R}'$ can be written as

$$t'_k(x) = (-1)^k \frac{x \mp d}{v} + \tau T_k - \tau|X_k|. \tag{4}$$

We claim the following:

▶ **Lemma 2.** *Assume that we have chosen $|X_k|$ such that Algorithm 1 solves rendezvous. Then, if the robots meet when $\mathcal{R}$ is approaching its k^{th} turning-point and $\mathcal{R}'$ is approaching its j^{th} turning-point then the time of rendezvous can be written as*

$$t_{k,j} = \frac{\pm(-1)^k d - (-1)^{k+j} v\tau(T_j - |X_j|) + T_k - |X_k|}{1 - (-1)^{k+j} v}$$

Proof. It is obvious that the possible points of rendezvous will occur at intersections of pairs of lines $t_k(x)$ and $t'_j(x)$, $k, j \geq 0$. The lemma follows from solving the equation $t_k(x) = t'_j(x)$. ◀

2.4 Competitive ratios

We are interested in algorithms that achieve small competitive ratios where we have defined the competitive ratio of an algorithm $\mathcal{A}$ as the supremum ratio of the time it takes to rendezvous using $\mathcal{A}$ to the time taken to rendezvous if the robots employ an optimal algorithm. We now give more precise definitions.

▶ **Definition 3.** Let $\mathcal{A}$ be an algorithm solving rendezvous in the T- or D-Models and let the time of rendezvous be $T_*(v, d)$. Then the competitive ratio of $\mathcal{A}$ is $CR = \sup_{v,d} \frac{(1-v)T_*(v,d)}{d}$.

▶ **Definition 4.** Let $\mathcal{A}$ be an algorithm solving rendezvous in the V-Model and let the time of rendezvous be $T_*(\tau, d)$. Then the competitive ratio of $\mathcal{A}$ is $CR = \sup_{\tau,d} \frac{(1-\tau)T_*(\tau,d)}{d}$.

We will justify these definitions as we consider each model in turn.

2.5 Feasibility of rendezvous

Here we establish the feasibility of rendezvous. To this end consider an algorithm in the class SPM with the curve $\mathcal{T}(x)$. We say that $\mathcal{T}(x)$ *contains* $\mathcal{T}'(x)$ (resp. $\mathcal{T}'(x)$ contains $\mathcal{T}(x)$) if there exists an x_0 such that for all x satisfying $|x| > |x_0|$ we have $\mathcal{T}(x) \leq \mathcal{T}'(x)$ (resp.

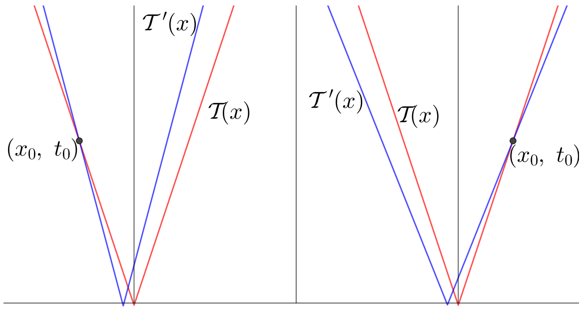

Figure 1 Illustration of the idea of containment. On the left $\mathcal{T}(x)$ contains $\mathcal{T}'(x)$ and on the right $\mathcal{T}'(x)$ contains $\mathcal{T}(x)$. In both cases the containment point is indicated.

$\mathcal{T}(x) \geq \mathcal{T}'(x)$). If such an x_0 exists we call x_0 and $t_0 = \mathcal{T}(x_0)$ the *containment point* and *containment time* of $\mathcal{T}(x)$ and $\mathcal{T}'(x)$. Intuitively, x_0 will be an intersection point of $\mathcal{T}(x)$ and $\mathcal{T}'(x)$. Figure 1 illustrates these definitions. We can now claim the following:

▶ **Lemma 5.** *If either of $\mathcal{T}(x)$ or $\mathcal{T}'(x)$ contains the other, then rendezvous is guaranteed.*

Proof. Assume that $\mathcal{T}(x)$ contains $\mathcal{T}'(x)$ and let t_0 be the containment time. Let X_k be the first turning-point that $\mathcal{R}$ reaches after the time t_0 and assume that $X_k > 0$. In this situation $\mathcal{R}$ will be on the right of $\mathcal{R}'$ once it reaches X_k and will be on the left of $\mathcal{R}'$ once it reaches X_{k+1}. Likewise, if $X_k < 0$, then $\mathcal{R}$ will be on the left of $\mathcal{R}'$ at X_k and on the right of $\mathcal{R}'$ at X_{k+1}. In either case the robots must rendezvous between the turning-points X_k and X_{k+1} of $\mathcal{R}$. In a similar manner one can confirm that the robots will rendezvous between the turning-points X'_j and X'_{j+1} of $\mathcal{R}'$ where X_j is the first turning-point $\mathcal{R}'$ reaches after the time t_0. ◀

In both the T- and D-models there are a wide range of algorithms in the class SPM with curves $\mathcal{T}(x)$ such that $\mathcal{T}(x)$ contains $\mathcal{T}'(x)$. In particular, if $v \neq 1$ and $\mathcal{T}(x)$ is a linear function, then $\mathcal{T}'(x) = \tau\mathcal{T}\left(\frac{x \mp d}{v\tau}\right) = \frac{1}{v}\mathcal{T}(x \mp d)$ and $\mathcal{T}(x)$ will clearly contain this. If $v = 1$ then there are still a variety of curves one can choose (for example, $\mathcal{T}(x) = x^2$) and thus rendezvous is guaranteed in general.

3 T-Model

In this section we analyze rendezvous under the assumption that the robots have the same time units, i.e. $\tau = 1$. This assumption turns out to be rather powerful as it allows us to reduce the problem into the problem of search for a stationary target.

▶ **Theorem 6.** *Rendezvous in T-Model is equivalent to search for a stationary target at distance $d_* = \frac{d}{1-v}$.*

Proof. Assume that we have a rendezvous algorithm that specifies the trajectory $R(t)$ for $\mathcal{R}$. The robot $\mathcal{R}'$ will then follow the actual trajectory $R'(t) = vR(t) \pm d$. Now consider a coordinate system moving with the robot $\mathcal{R}'$ and scaled by a factor of $\frac{1}{1-v}$. In this coordinate system $\mathcal{R}'$ will appear to be stationary at the position $\pm\frac{d}{1-v}$ and $\mathcal{R}$ will appear to move along the trajectory $R_*(t) = R(t)$. We can thus view this problem as a search problem for a target at distance $\frac{d}{1-v}$ and any algorithm solving search will also solve rendezvous in the same amount of time. ◀

This clearly justifies our definition of the competitive ratio for this model. Furthermore, this result allows us to draw on many of the results known about search and, in particular, allows one to optimally solve the rendezvous problem.

▶ **Theorem 7.** *Rendezvous in T-Model is optimally solved with a competitive ratio of 9 using Algorithm 1 with $|X_k| = 2^k$.*

Proof. Observe that Algorithm 1 with $|X_k| = 2^k$ is the familiar cow-path algorithm which optimally solves search for a target at distance d in time $9d$. Since search is equivalent to rendezvous in T-Model, this algorithm will optimally solve rendezvous as well. ◀

Note that the optimal solution in this case lies within the class SPM.

4 D-Model

In this section we analyze rendezvous under the assumption that the robots have the same distant-units, i.e. $v\tau = 1$. Unlike the T-model, we cannot transform the problem into the search problem and thus we will approach our analysis slightly different. We begin with a lemma that justifies our definition of the competitive ratio for this model.

▶ **Theorem 8.** *Rendezvous in D-model takes time at least $\frac{d}{1-v}$.*

Proof. Consider any rendezvous algorithm with turning points X_k. Assume that the robots rendezvous between the $(k-1)^{st}$ and k^{th} turning points of $\mathcal{R}$ and the $(j-1)^{st}$ and j^{th} turning points of $\mathcal{R}'$. Furthermore, assume that $\mathcal{R}'$ begins to the right of $\mathcal{R}$. In this case $\mathcal{R}$ must be moving to the right when the robots rendezvous and thus k will be even. Assume that d and τ are chosen such that the robots rendezvous when j is also even. Then, by Lemma 2, the robots will rendezvous at the time

$$t_{k,j} = \frac{d - (T_j - |X_j|) + (T_k - |X_k|)}{1-v} = \frac{d - 2\sum_{i=0}^{j-1}|X_i| + 2\sum_{i=0}^{k-1}|X_i|}{1-v}.$$

Since $v < 1$ and $v\tau = 1$ we have $\tau > 1$ and thus $\mathcal{R}'$ will always take longer to finish a round. We must therefore have $j \le k$ and the rendezvous time will be at least $\frac{d}{1-v}$. ◀

We note that the above lower bound is general and applies to any algorithm (not just those in the class SPM). We now claim the following:

▶ **Theorem 9.** *Rendezvous in D-Model is solved with a competitive ratio of $\frac{105}{11}$ using Algorithm 1 with $|X_k| = 2^k$. Furthermore, in the limit of large d, the competitive ratio is 9.*

Proof. Assume that $\mathcal{R}$ is approaching its k^{th} turning-point and when $\mathcal{R}'$ is approaching its j^{th} turning-point the robots rendezvous. Then, by Lemma 2, they will rendezvous at the time

$$t_{k,j} = \frac{\pm(-1)^k d - (-1)^{k+j}(T_j - |X_j|) + T_k - X_k}{1-(-1)^{k+j}v}.$$

Observe that the robots will never rendezvous when $\mathcal{R}$ is traveling away from $\mathcal{R}'$. Thus, if $\mathcal{R}'$ begins to the right of $\mathcal{R}$, k must be even. Likewise, if $\mathcal{R}'$ begins to the left of $\mathcal{R}$, k must be odd. On the other hand, $\mathcal{R}'$ may be moving towards or away from $\mathcal{R}$ when they rendezvous and thus j may be even or odd. This gives two possible rendezvous times depending on the parity of $k+j$. If $k+j$ is even then $t_{k,j} = \frac{d+2^{k+1}-2^{j+1}}{1-v}$ and if $k+j$ is odd then

$t_{k,j} = \frac{d+2^{k+1}+2^{j+1}-4}{1+v}$. The competitive ratios for these two cases are $CR_+ = 1 + \frac{2}{d}(2^k - 2^j)$ and $CR_- = \frac{1-v}{1+v}\left[1 + \frac{2}{d}(2^k + 2^j - 2)\right]$ where CR_+ (resp. CR_-) corresponds to $k+j$ even (resp. $k+j$ odd).

We note that, in order for $\mathcal{R}$ and $\mathcal{R}'$ to meet while traveling towards their k^{th} and j^{th} turning-points, the time $t_{k,j}$ must satisfy $T_{k-1} \leq t_{k,j} \leq T_k$ and $T'_{j-1} \leq t_{k,j} \leq T'_j$. Furthermore, since $\mathcal{R}'$ will take longer to finish a round, there can be at most one turning-point of $\mathcal{R}'$ between any two turning-points of $\mathcal{R}$.

Now assume that $k+j$ is even. In this case an adversary will get the best payoff if they choose d as small as they can without causing the robots to rendezvous on an earlier round. The smallest value of d is achieved if d and v can be chosen such that the $(k-2)^{nd}$ turning-point of $\mathcal{R}$ is arbitrarily close to the $(j-2)^{nd}$ turning-point of $\mathcal{R}'$ (see the left side of Figure 2). We claim that the adversary can choose d and v to achieve this. To see why assume that the adversary has chosen d and v such that $T_{k-2} = T'_{j-2}$. In order to rendezvous at the time $t_{k,j}$ then we need to satisfy $T'_{j-1} \leq t_{k,j} \leq T'_j$. Since the robots will not rendezvous when $\mathcal{R}$ is moving away from $\mathcal{R}'$, we can easily conclude that $T'_{j-1} \leq t_{k,j}$. Also, since $t_{k,j}$ must be smaller than T_k, we can also conclude that $t_{k,j} \leq T'_j$ due to the fact that $T'_{j-2} = T_{k-2}$ and because $\mathcal{R}'$ takes longer to finish a round. Thus, in the worst case, we may take $X_{k-2} = X'_{j-2}$ which, after simplification, tells us that we should take $d = 2^{k-2} - 2^{j-2}$. Using this result in the expression for CR_+ yields a competitive ratio of 9.

Now assume that $k+j$ is odd. In this case the competitive ratio will depend on both d and v. However, we can similarly conclude that the adversary would like to minimize d and thus they will try to choose d and v in order to make the $(j-1)^{st}$ turning-point of $\mathcal{R}'$ equal (or arbitrarily close) to the $(k-2)^{nd}$ turning-point of $\mathcal{R}$. However, in this case, we claim that the adversary cannot always do this. Indeed, if $T_{k-2} = T'_{j-1}$ and $X_{k-2} = X'_{j-1}$ then, after some manipulation, we find that,

$$2^{j-1} = \frac{dv}{1-v} + \frac{2}{3} \tag{5}$$

and

$$2^{k-2} = \frac{d}{1-v} + \frac{2}{3}. \tag{6}$$

Since we are assuming that $\mathcal{R}'$ meets $\mathcal{R}$ as it is traveling towards its j^{th} turning-point we need $t_{k,j} \leq T'_j$. Using (6), (5), and the expression for $t_{k,j}$ we find that d must satisfy $d \leq \frac{2}{3}\left(\frac{1}{v} - 1\right)$ and thus j must satisfy $2^{j-1} = \frac{dv}{1-v} + \frac{2}{3} \leq \frac{4}{3}$. Clearly, the only j that satisfies this is $j = 1$ and we can conclude that, in the case that $k+j$ is odd, the worst-case situations occur when $j = 1$ (and k is even). When $j = 1$, we have from (5) that $d = \frac{1-v}{3v}$ and from (6) that $2^k = \frac{4}{3}\left(\frac{1}{v} + 2\right)$. Using these results in the expression for CR_- gives us $CR_- = \frac{15v+9}{1+v}$ and this can be seen to increase with v. However, we cannot simply maximize this over v. If we take $k = 2$ then we need $v = 1$ and $d \leq 0$ and this is clearly not possible. We can thus conclude that $k \geq 4$. If $k = 4$ then one can confirm that $v = \frac{1}{10}$, and if $k > 4$ then $v < \frac{1}{10}$. Thus, we find the maximum competitive ratio when $v = \frac{1}{10}$. Substituting this result in for CR_- we find that $CR_- \leq \frac{105}{11}$.

Finally, since we found that $2^k = \frac{4}{3}\left(\frac{1}{v} + 2\right)$ in the worst case, we can see that for very large k we will have v very small. Since k will be large for large d and since $CR_- = \frac{15v+9}{1+v} = 9$ when $v = 0$, we can conclude that the competitive ratio approaches 9 as d gets large. ◀

We note that the upper-bound of Theorem 9 is tight for the algorithm considered since it is easily confirmed that the competitive ratio is exactly 9.55 when $v = \frac{1}{10}$ and $d = 3$.

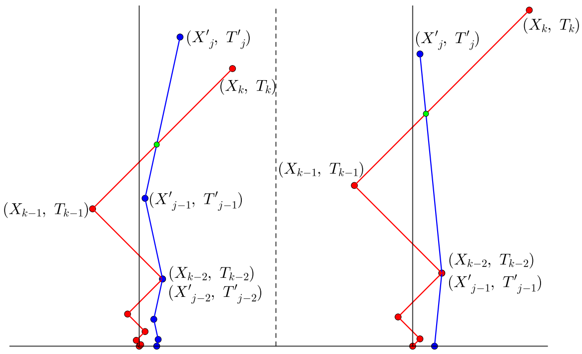

Figure 2 The two worst case possibilities in D-Model using Algorithm 1 with $|X_k| = 2^k$. The point of rendezvous is indicated in green and in both cases the robots rendezvous as $\mathcal{R}$ is approaching its jk^{th} turning-point and when $\mathcal{R}'$ is approaching its j^{th} turning-point. Left: The case that $k+j$ is even. Right: The case that $k+j$ is odd. The absolute worst-case is depicted and occurs when $d = 3$ and $v = \frac{1}{10}$ such that the competitive ratio is 9.55. Note that the scales of the depicted trajectories are not the same for the two cases.

What is somewhat surprising about this result is that the competitive ratio is nearly identical to that of the T-Model when d is large, and this is achieved using the same algorithm. We suspect that this algorithm is optimal here since, in the limit that v goes to zero, this model reduces to search for which a competitive ratio of 9 is optimal. We do not have a formal proof of this, however, and thus it is still an open question whether or not this algorithm is optimal.

5 V-Model

In this section we analyze upper bounds on rendezvous when the robots' speeds are equal and $0 < \tau < 1$. In both the T-Model and D-Model the robots employed Algorithm 1 with a geometric sequence of turning-points ($|X_k| = 2^k$) in order to rendezvous with a small constant competitive ratio. We will see that the V-Model is rather more complicated, and, in particular, a geometric sequence of turning-points will not work (see Lemma 18 at the end of this section). We will therefore have to employ a different type of algorithm. We begin, however, with a lower-bound to justify our definition of the competitive ratio:

▶ **Theorem 10.** *Rendezvous in V-Model takes time at least $\frac{d}{1-\tau}$.*

Proof. For concreteness assume that $\mathcal{R}'$ begins to the right of $\mathcal{R}$. Then, the robots will rendezvous as $\mathcal{R}$ is traveling to the right and $\mathcal{R}'$ is traveling to the left. Consider the j^{th} turning-point of $\mathcal{R}'$ and assume that j is even such that $\mathcal{R}'$ moves left after this turning-point. Let X_k be the first even turning-point of $\mathcal{R}'$ after the j^{th} turning-point of $\mathcal{R}'$. Observe that the robots will rendezvous before the $(j+1)^{st}$ turning-point of $\mathcal{R}'$ provided that $X'_j - X_k \leq T_k - T'_j$. Now assume that τ is close enough to one such that we may assume that $j = k$. In this case we can rewrite the condition $X'_j - X_k \leq T_k - T'_j$ as $T_k + X_k \geq \frac{d}{1-\tau}$.

Since we are free to choose d and τ let us choose these parameters such that $T_k + X_k = \frac{d}{1-\tau} - \epsilon$ for an arbitrarily small $\epsilon > 0$. Then the robots will not rendezvous until some time after the $(k+2)^{nd}$ turning-point of $\mathcal{R}'$. Since $T_{k+2} = T_{k+1} + |X_{k+1}| + |X_{k+2}| = T_k + |X_k| + 2|X_{k+1}| + |X_{k+2}| = \frac{d}{1-\tau} - \epsilon + 2|X_{k+1}| + |X_{k+2}|$ we can conclude that the robots will take at least $\frac{d}{1-\tau}$ time to rendezvous. ◀

Note that this lower bound applies to any algorithm. Now for an upper bound. We claim the following:

▶ **Theorem 11.** *Rendezvous in V-Model is solved with a competitive ratio of* $\frac{18\log^2(d)\tau}{|\log(\tau)|} + \mathcal{O}\left(\frac{\log(d)\tau}{|\log(\tau)|}\right)$ *using Algorithm 1 with* $|X_k| = (k+2)2^k$.

An overview of the proof of this theorem is as follows. We first show that Algorithm 1 with $|X_k| = (k+2)2^k$ does indeed solve rendezvous and we will do this by demonstrating that $\mathcal{T}(x)$ contains $\mathcal{T}'(x)$. Since $\mathcal{T}(x)$ contains $\mathcal{T}'(x)$, the robots will be guaranteed to rendezvous by the second turning-point reached by $\mathcal{R}$ after the containment time and we will use this fact to bound the rendezvous time. Throughout the proof we will need to make use of the Lambert-W function (or simply Lambert function) $W(x)$ which is defined as the inverse function of $f(x) = xe^x$ (we consider the real valued branches only and thus x is restricted to the range $x \geq \frac{-1}{e}$). Since $W(x)$ is multivalued on the open interval $(-e^{-1}, 0)$ it is usual to define $W_{-1}(x)$ as the branch which attains values ≤ -1 and reserve the use of $W(x)$ to refer to the principal branch which attains values ≥ -1. We will need the following properties of $W(x)$ which are found in, or trivially derived from, the results in [21] and [13]:

▶ **Lemma 12.** *The two real valued branches* $W(x)$ *and* $W_{-1}(x)$ *satisfy:*

$$W(x) \leq \ln(x) - \ln(\ln(x)) + \frac{e}{e-1} \cdot \frac{\ln(\ln(x))}{\ln(x)}, \quad x \geq e \tag{7}$$

$$W(x) \geq \ln(x) - \ln(\ln(x)) + \frac{\ln(\ln(x))}{2\ln(x)}, \quad x \geq e \tag{8}$$

$$W_{-1}(x) < \ln(-x), \tag{9}$$

$$\frac{d}{dx}\frac{x}{W(x)} = \frac{1}{1+W(x)}, \tag{10}$$

$$\frac{d^2}{dx^2}\frac{x}{W(x)} < 0. \tag{11}$$

Before we can demonstrate that the algorithm solves rendezvous we need to first determine the curve $\mathcal{T}(x)$ induced by the turning-points $|X_k| = (k+2)2^k$.

▶ **Lemma 13.** *Let* $\mathcal{T}(x)$ *be the curve induced by Algorithm 1 with* $|X_k| = (k+2)2^k$*. Then*

$$\mathcal{T}(x) = 3|x| - \frac{4\ln(2)|x|}{W(4\ln(2)|x|)}.$$

Proof. One can observe that the turning-points $|X_k| = (k+2)2^k$ form an arithmetico-geometric sequence and its sum has the closed form expression $\sum_{i=0}^{k-1}|X_i| = k2^k$. We therefore have $T_k = 2\sum_{i=0}^{k-1}|X_i| + |X_k| = (3k+2)2^k$. We may rewrite T_k as $T_k = 3(k+2)2^k - 4\cdot 2^k = 3|X_k| - \frac{4|X_k|}{k+2}$.

To express T_k fully in terms of $|X_k|$ we need to invert $|X_k| = (k+2)2^k$. We can do this using the Lambert function. First rewrite the equation $|X_k| = (k+2)2^k$ as $4\ln(2)|X_k| = \ln(2)(k+2)e^{\ln(2)(k+2)}$. In this form we can directly apply the definition of the Lambert function to get the solution $\ln(2)(k+2) = W(4\ln(2)|X_k|)$. We can therefore express T_k as $T_k = 3|X_k| - \frac{4\ln(2)|X_k|}{W(4\ln(2)|X_k|)}$. Since $\mathcal{T}(X_k) = T_k$ the lemma follows. ◀

Now that we have determined the curve $\mathcal{T}(x)$ induced by the turning points $|X_k| = (k+2)2^k$, we can show that $\mathcal{T}(x)$ will contain $\mathcal{T}'(x)$ and thus the algorithm will solve rendezvous.

▶ **Lemma 14.** *Consider Algorithm 1 with* $|X_k| = (k+2)2^k$*. Then* $\mathcal{T}(x)$ *contains* $\mathcal{T}'(x)$*.*

Proof. We need to show that, for all d and $\tau < 1$, there exists an x_0 such that for all $|x| > x_0$ we have $\mathcal{T}'(x) > \mathcal{T}(x)$. To do this we will assume that $d = 0$ and show that the difference $D(x) = \mathcal{T}'(x) - \mathcal{T}(x)$ grows without bound for all τ satisfying $0 < \tau < 1$. If this is the case, then, no matter the value of d, there will eventually be an x_0 such that for all $x > x_0$ we have $\mathcal{T}'(x) > \mathcal{T}(x)$.

To this end consider the rate of change of $\mathcal{T}(x)$ for $x > 0$. Using (10) we find that $\frac{d\mathcal{T}(x)}{dx} = 3 - \frac{4\ln(2)}{1+W(4\ln(2)x)}$. Since $W(0) = 0$ and $W(x)$ is an increasing function, $\mathcal{T}(x)$ is also increasing for all $x > 0$. Furthermore, if $d = 0$ we have $\frac{d\mathcal{T}'(x)}{dx} = 3 - \frac{4\ln(2)}{1+W(4\ln(2)\frac{x}{\tau})}$ and this is clearly larger than $\frac{d\mathcal{T}(x)}{dx}$ for $0 < \tau < 1$. The rate of change of $D(x)$ must therefore always be positive and thus $D(x)$ does indeed grow without bound. ◀

We need one more simple lemma before tackling the proof of Theorem 11.

▶ **Lemma 15.** *Consider Algorithm 1 with $|X_k| = (k+2)2^k$ and let $\mathcal{T}(x)$, x_0, and t_0 be the induced curve, and containment time and position. Then $t_0 < 3x_0$ and $\tau\mathcal{T}(x-d) > \mathcal{T}(x) - 3d$.*

Proof. The first part of the lemma follows easily from the fact that $\frac{d\mathcal{T}(x)}{dx} < 3$. The second part also follows easily from the facts that $\frac{d\mathcal{T}(x)}{dx} < 3$ and $\mathcal{T}(x)$ is a convex function (see (11)). As a result, $\mathcal{T}(x)$ always lies below any secant line, and every secant line will have a slope less than three. ◀

Proof. (Theorem 11) We would like to bound the rendezvous time and to do this we will first bound the containment time. Since the robots must rendezvous by the second turning-point of $\mathcal{R}$ after the containment time, and since $\frac{|X_{k+2}|}{|X_k|} \leq 8$, the rendezvous time will be bounded by eight times the containment time. By Lemma 15 the containment time is itself bounded by three times the containment position and thus we will actually determine a bound on the containment position. Thus, if T_*, t_0, and x_0 are respectively the rendezvous time, containment time, and containment position, then $T_* < 8t_0 < 24|x_0|$. We will assume that $x_0 > 0$ and note that, since $\mathcal{T}(x)$ contains $\mathcal{T}'(x)$, we must have $d > 0$ in order for $x_0 > 0$.

To begin, we note that x_0 is the solution to the equation $\mathcal{T}'(x) - \mathcal{T}(x) = 0$, and, since this difference is increasing, any x satisfying $\mathcal{T}'(x) - \mathcal{T}(x) > 0$ will suffice for a bound. In particular, since $\mathcal{T}'(x) = \tau\mathcal{T}(\frac{x-d}{\tau}) > \tau\mathcal{T}(\frac{x}{\tau}) - 3d$, we have $\mathcal{T}'(x) - \mathcal{T}(x) > \tau\mathcal{T}(\frac{x}{\tau}) - \mathcal{T}(x) - 3d$ and we will thus bound x_0 by an x satisfying $\tau\mathcal{T}(\frac{x}{\tau}) - \mathcal{T}(x) > 3d$.

To simplify notation we introduce the variable $y = 4\ln(2)x$ and set $f(y) = \frac{1}{W(y)} - \frac{1}{W(\frac{y}{\tau})}$ and $D(y) = y \cdot f(y)$. We therefore wish to find a y satisfying $D(y) = y \cdot f(y) > 3d$. By (8) and (7) we can write $f(y) \geq \frac{1}{\ln(y)-\ln(\ln(y))+\frac{e}{e-1}\frac{\ln(\ln(y))}{\ln(y)}} - \frac{1}{\ln(\frac{y}{\tau})-\ln(\ln(\frac{y}{\tau}))+\frac{\ln(\ln(\frac{y}{\tau}))}{2\ln(\frac{y}{\tau})}}$. Now set $z = \ln(y)$. The right hand side of the above inequality then becomes

$$h(z) = \frac{1}{z - \ln(z) + \frac{e}{e-1}\frac{\ln(z)}{z}} - \frac{1}{z - \ln(\tau) - \ln(z - \ln(\tau)) + \frac{\ln(z-\ln(\tau))}{2(z-\ln(\tau))}}.$$

One can confirm that $h(z)$ admits a generalized Puiseaux series in the limit $z \to \infty$. Keeping only the leading term of this series we find that $h(z) = \frac{\ln(\frac{1}{\tau})}{z^2} + \mathcal{O}\left(\frac{\ln^2(z)}{z^3}\right)$. We can therefore conclude that $f(y) > \frac{\ln(\frac{1}{\tau})}{\ln^2(y)}$, $D(y) > \frac{\ln(\tau)y}{\ln^2(y)}$, and we now wish to find a y satisfying $\frac{\ln(\tau)y}{\ln^2(y)} \geq 3d$. Let y_+ be the solution to $\frac{\ln(\tau)y}{\ln^2(y)} = 3d$. We can use the Lambert W function to solve this equation. We find that $y_+ = \frac{12d}{\ln(\frac{1}{\tau})}W_{-1}^2\left(-\sqrt{\frac{\ln(\frac{1}{\tau})}{12d}}\right)$. Since the rendezvous time satisfies

$T_* < 8t_0 < 24x_0 < \frac{6y_+}{\ln(2)}$, we find that $T_* < \frac{72d}{\ln(2)\ln(\frac{1}{\tau})} W_{-1}^2 \left(-\sqrt{\frac{\ln(\frac{1}{\tau})}{12d}}\right)$. To express this in terms of more familiar functions we can use (9) to write

$$T_* < \frac{72d}{\ln(2)\ln(\frac{1}{\tau})} \ln^2 \left(\sqrt{\frac{\ln(\frac{1}{\tau})}{12d}}\right) = \frac{18d\ln^2(d)}{\ln(2)\ln(\frac{1}{\tau})} + \mathcal{O}\left(\frac{d\ln(d)}{\ln(\frac{1}{\tau})}\right)$$

Expressing T_* with base-2 logarithms and dividing by $\frac{d}{1-\tau}$ gives the desired bound on the competitive ratio. ◀

If we abandon the use of algorithms with turning-points that are easily defined, then we can get an algorithm with a competitive ratio which is slightly “tunable”. We claim the following:

▶ **Theorem 16.** *Rendezvous in V-Model is solved with a competitive ratio of* $\frac{72\ln^c(2) d \log(d)^{1+c}}{c|\log(\frac{1}{\tau})|}$ $+\mathcal{O}\left(\frac{d\log^c(d)}{c\log(\frac{1}{\tau})}\right)$ *using Algorithm 2 with* $\mathcal{T}(x) = 3|x| - \frac{|x|}{\ln^c(|x|)}$ *where* $c > 0$ *is a parameter of the algorithm.*

As the proof of Theorem 16 is essentially identical to that for Theorem 11 we do not provide it here.

There are a couple of things to note about this upper bound. First, although we can reduce the exponent of the $\log(d)$ term by making c small, we cannot make it arbitrarily small without suffering a large multiplicative constant due to the $\frac{1}{c}$ term in the competitive ratio. Thus, the algorithm of Theorem 16 will be most useful if one knows a lower bound on d as this will allow one to compare the bounds of Theorem 11 and 16 and choose an appropriate c. Without this knowledge it may be better to just stick with the algorithm of Theorem 11 as it has the benefit of having simple turning-points.

The upper-bounds of Theorem 11 and 16 are clearly much worse than the competitive ratios found for both the T- and D-Models. In those cases we had a constant competitive ratio and in these cases the competitive ratio is unbounded. One might then expect that we can do better. This, however, does not seem to be the case. We provide two arguments for this. First off, if one tries to use an algorithm in which the leading term of $\mathcal{T}(x)$ is $\omega(x)$ then we arrive to a similar result – the competitive ratio is unbounded.

▶ **Lemma 17.** *If the leading term of $\mathcal{T}(x)$ is $\omega(x)$ then the competitive ratio is $\omega(1)$.*

Proof. We observe that before the robots can rendezvous, there must be, at the very least, an intersection point (x_*, t_*) of the curves $\mathcal{T}(x)$ and $\mathcal{T}'(x)$ satisfying $0 < |x_*| < d$. We will show that we can always choose d and τ to make the time $t_* = \omega\left(\frac{d}{1-\tau}\right)$ if the leading term of $\mathcal{T}(x)$ is $\omega(x)$. For concreteness, assume that $\mathcal{R}'$ begins to the right of $\mathcal{R}$ such that x_* is the intersection point of the right arm of $\mathcal{T}(x)$ and the left arm of $\mathcal{T}'(x)$. Furthermore, since $\mathcal{T}(x) = \omega(x)$, we will express the leading order term of $\mathcal{T}(x)$ as $x \cdot f(x)$ for some positive function $f(x) = \omega(1)$.

We claim that x_* is bigger than $\frac{d}{2}$. Indeed, the right arm of $\mathcal{T}(x)$ is bounded from below by x and the left arm of $\mathcal{T}'(x)$ is bounded from below by $d - x$. Since $x = d - x$ when $x = \frac{d}{2}$, we must have $x_* > \frac{d}{2}$. However, if $x_* > \frac{d}{2}$ then $t_* > \mathcal{T}(\frac{d}{2}) > \frac{d}{2}f(\frac{d}{2})$. For any fixed $\tau \neq 1$ we can take d large enough that $\frac{d}{2}f(\frac{d}{2}) > \frac{d}{1-\tau}g(\frac{d}{1-\tau})$ for an appropriately chosen function $g(x) = \omega(1)$. Thus, $t_* = \omega\left(\frac{d}{1-\tau}\right)$ and, since the rendezvous time is larger than t_*, the lemma follows. ◀

Thus, the two algorithms analyzed in this section do seem to be from the "right" class of algorithms one should consider if one is to hope for a constant competitive ratio. The next lemma – which demonstrates why a geometric sequence of turning-points cannot be used – also supports this conclusion:

▶ **Lemma 18.** *If $\mathcal{T}(x)$ has the form $\mathcal{T}(x) = ax + G(x)$ with $a > 1$ and $|G(x)| = O(1)$ then there are choices of $\tau \neq 1$ such that the robots will never rendezvous.*

Proof. Let us first determine what the turning-points of $\mathcal{T}(x)$ look like. Since the turning-points satisfy the relation $\mathcal{T}(X_k) = T_k$ and $T_k = 2\sum_{i=0}^{k-1}|X_i| + |X_k|$ we have $a|X_k| + G(X_k) = 2\sum_{i=0}^{k-1}|X_i| + |X_k|$ and this simplifies to $|X_k| = \frac{2}{a-1}\sum_{i=0}^{k-1}|X_i| - G(X_k)$. Set $G_k = G(X_k)$ and observe that

$$|X_{k+1}| = \frac{2}{a-1}\sum_{i=0}^{k}|X_i| - G_{k+1} = \frac{2}{a-1}|X_k| + \left(\frac{2}{a-1}\sum_{i=0}^{k-1}|X_i| - G_k\right) + G_k - G_{k+1}$$
$$= \frac{a+1}{a-1}|X_k| + G_k - G_{k+1}$$

and we can therefore see that X_{k+1} is nearly a geometric sequence with common ratio $\frac{a+1}{a-1}$.

Let $b = \frac{a+1}{a-1}$, and assume that $\tau = b^{-2}$ and that $\mathcal{R}'$ begins to the right of $\mathcal{R}$. In this case we can write

$$|X_{k+2}| = b|X_{k+1}| + G_{k+1} - G_{k+2} = b(b|X_k| + G_k - G_{k+1}) + G_{k+1} - G_{k+2}$$
$$= b^2|X_k| + bG_k - (b-1)G_{k+1} - G_{k+2}.$$

Combining this with the fact that $|X'_{k+2}| = \tau|X_{k+2}| + d$ gives us

$$|X'_{k+2}| - |X_k| = d + \frac{1}{b^2}[bG_k - (b-1)G_{k+1} - G_{k+2}] = d + \Delta_k.$$

implying that the robot $\mathcal{R}'$ will be a distance $d + \Delta_k$ from its $(k+2)^{nd}$ turning-point when $\mathcal{R}$ reaches its k^{th} turning-point. Clearly, in order to rendezvous, $d + \Delta_k$ must eventually decrease to zero. However, since $G(x) = \mathcal{O}(1)$, we can always take d sufficiently large such that this difference is bounded from below by a positive constant. Thus, with an appropriate choice of τ and d, the robots will never rendezvous. ◀

We are thus left with a rather small class of functions that $\mathcal{T}(x)$ can belong to – if $\mathcal{T}(x) = \omega(x)$ then the algorithm will take too much time, and if $\mathcal{T}(x) = a \cdot x \pm \mathcal{O}(1)$ then rendezvous cannot be solved. Thus, the only possibility left is if $\mathcal{T}(x) = a \cdot x + f(x)$ with $|f(x)| = o(x)$ and $|f(x)| = \omega(1)$. The two algorithms analyzed in this section each used curves of this form.

It is interesting to note that simulations of the two algorithms in this section show that there are choices of d and τ to (nearly) match the upper-bounds derived here. Even more interesting is that, in order to achieve these worst-case situations, one chooses d and τ precisely so that the two robots have turning-points that are arbitrarily close to the containment point of the curves $\mathcal{T}(x)$ and $\mathcal{T}'(x)$. This reflects a similar argument we made when we derived an upper-bound on rendezvous in the D-Model. In that case it also turned out that there were choices of d and τ in order to achieve the upper bound. If one could prove that it is always possible to choose d and τ such that the robots do have turning-points arbitrarily close to the containment point for all algorithms with $\mathcal{T}(x) = a \cdot x + f(x)$, $|f(x)| = o(x)$, and $|f(x)| = \omega(1)$ then a lower-bound that grows with d would easily follow. This is easier said than done, however, and we leave it as an open problem whether or not one can achieve a constant competitive ratio in the V-Model.

Finally, we note that both algorithms provided in this section are universal in the sense that they will also solve the problem if both v and τ are different than one (it is trivial to see that one of $\mathcal{T}(x)$ or $\mathcal{T}'(x)$ will contain the other if $v \neq 1$). Since a robot does not need to know the values of its parameters in order to employ these algorithms we can conclude that it is sufficient to rendezvous if at least one of v or τ is different than one.

▶ **Theorem 19.** *Both of Algorithm 1 with $X_k = (k+2)2^k$ and Algorithm 2 with $\mathcal{T}(x) = 3|x| - \frac{|x|}{\ln^c(|x|)}$ solve rendezvous in general if at least one of v or τ is different than one.*

The time complexity of this more general model does not turn out to be all that interesting to study since, in the worst cases, an adversary chooses $v = 1$. Thus, the general model reduces to the V-Model and all of the results derived here still apply.

6 Discussion and conclusion

The focus of our paper was on symmetric rendezvous on an infinite line for two robots endowed with asymmetric clocks. After introducing the new concept of asymmetric clocks, we gave a universal algorithm which ensures feasibility of rendezvous if at least one of the robots' maximal speeds or time units differ. We analyzed the impact of equal time-unit, distance-unit, and equal speeds of the robots on the competitive ratio of the cost of rendezvous. The problem considered not only provides a surprising twist to the well-known rendezvous problem on an infinite line, it also creates interesting avenues for future research. These may include improving the algorithms, tightening bounds, employing robots that may have alternative capabilities (visibility and variable speed), as well as extensions to gathering for multiple robots.

References

1. S. Alpern and S. Gal. *The theory of search games and rendezvous*, volume 55. Kluwer Academic Publishers, 2002.
2. R. Baeza Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106(2):234–252, 1993.
3. R. Baeza-Yates and R. Schott. Parallel searching in the plane. *Computational Geometry*, 5(3):143–154, 1995.
4. L. Barrière, P. Flocchini, P. Fraigniaud, and N. Santoro. Rendezvous and Election of Mobile Agents: Impact of Sense of Direction. *Theory Comput. Syst.*, 40(2):143–162, 2007.
5. A. Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.
6. A. Beck and D. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429, 1970.
7. R. Bellman. An optimal search. *SIAM Review*, 5(3):274–274, 1963.
8. S. Bouchard, Y. Dieudonné, A. Pelc, and F. Petit. On deterministic rendezvous at a node of agents with arbitrary velocities. *Inf. Process. Lett.*, 133:39–43, 2018.
9. Q. Bramas and S. Tixeuil. Wait-Free Gathering Without Chirality. In *Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings*, pages 313–327, 2015.
10. M. Chrobak, L. Gasieniec, Gorry T., and R. Martin. Group Search on the Line. In *Proceedings of SOFSEM 2015, LNCS 8939*, pages 164–176. Springer, 2015.

11 R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 549–560. Springer, 2006.

12 A. Collins, J. Czyzowicz, L. Gasieniec, and A. Labourel. Tell me where I am so I can meet you sooner. In *International Colloquium on Automata, Languages, and Programming*, pages 502–514. Springer, 2010.

13 R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, December 1996.

14 J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, and J. Opatrny. Search on a Line with Faulty Robots. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC*, pages 405–414, 2016.

15 J. Czyzowicz, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, and S. Shende. Linear search with terrain-dependent speeds. In *International Conference on Algorithms and Complexity*, pages 430–441. Springer, 2017.

16 G. De Marco, L. Gargano, E. Kranakis, D Krizanc, A. Pelc, and U. Vaccaro. Asynchronous deterministic rendezvous in graphs. *Theoretical Computer Science*, 355(3):315–326, 2006.

17 E. D. Demaine, S. P. Fekete, and S. Gal. Online searching with turn cost. *Theoretical Computer Science*, 361(2):342–355, 2006.

18 A. Dessmark, P. Fraigniaud, D. R. Kowalski, and A. Pelc. Deterministic Rendezvous in Graphs. *Algorithmica*, 46(1):69–96, 2006.

19 Y. Dieudonné, A. Pelc, and V. Villain. How to Meet Asynchronously at Polynomial Cost. *SIAM J. Comput.*, 44(3):844–867, 2015.

20 O. Feinerman, A. Korman, S. Kutten, and Y. Rodeh. Fast rendezvous on a cycle by agents with different speeds. *Theoretical Computer Science*, 688:77–85, 2017.

21 A. Hoorfar and M. Hassani. Inequalities on the Lambert W function and hyperpower function. *JIPAM. Journal of Inequalities in Pure & Applied Mathematics [electronic only]*, 9, January 2008.

22 T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita. The Gathering Problem for Two Oblivious Robots with Unreliable Compasses. *SIAM J. Comput.*, 41(1):26–46, 2012.

23 M.-Y. Kao, J. H. Reif, and S. R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Information and Computation*, 131(1):63–79, 1996.

24 E. Kranakis, N. Santoro, C. Sawchuk, and D. Krizanc. Mobile agent rendezvous in a ring. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, pages 592–599. IEEE, 2003.

25 A. Pelc. Deterministic rendezvous in networks: A comprehensive survey. *Networks*, 59(3):331–347, 2012.

26 G. Stachowiak. Asynchronous Deterministic Rendezvous on the Line. In *SOFSEM 2009: Theory and Practice of Computer Science*, pages 497–508. Springer, 2009.

27 X. Yu and M. Yung. Agent rendezvous: A dynamic symmetry-breaking problem. In *International Colloquium on Automata, Languages, and Programming*, pages 610–621. Springer, 1996.