

# How to meet asynchronously (almost) everywhere

Jurek Czyzowicz <sup>\*†</sup>

Arnaud Labourel <sup>\*‡</sup>

Andrzej Pelc <sup>\*§</sup>

## Abstract

Two mobile agents (robots) with distinct labels have to meet in an arbitrary, possibly infinite, unknown connected graph or in an unknown connected terrain in the plane. Agents are modeled as points, and the route of each of them only depends on its label and on the unknown environment. The actual walk of each agent also depends on an asynchronous adversary that may arbitrarily vary the speed of the agent, stop it, or even move it back and forth, as long as the walk of the agent in each segment of its route is continuous, does not leave it and covers all of it. Meeting in a graph means that both agents must be at the same time in some node or in some point inside an edge of the graph, while meeting in a terrain means that both agents must be at the same time in some point of the terrain. Does there exist a deterministic algorithm that allows any two agents to meet in any unknown environment in spite of this very powerful adversary? We give deterministic rendezvous algorithms for agents starting at arbitrary nodes of any anonymous connected graph (finite or infinite) and for agents starting at any interior points with rational coordinates in any closed region of the plane with path-connected interior. While our algorithms work in a very general setting – agents can, indeed, meet almost everywhere – we show that none of the above few limitations imposed on the environment can be removed. On the other hand, our algorithm also guarantees the following *approximate rendezvous* for agents starting at *arbitrary* interior points of a terrain as above: agents will eventually get at an arbitrarily small positive distance from each other.

**keywords:** rendezvous, graph, asynchronous, deterministic

---

\*Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mails: jurek@uqo.ca, labourel.arnaud@gmail.com, pelc@uqo.ca

†Partially supported by NSERC discovery grant.

‡This work was done during this author's stay at the Université du Québec en Outaouais as a postdoctoral fellow.

§Partially supported by NSERC discovery grant and by the Research Chair in Distributed Computing at the Université du Québec en Outaouais.

# 1 Introduction

**The problem and the model.** Two mobile agents (robots) modeled as points starting at different locations of an unknown environment have to meet. This task is known in the literature as the rendezvous problem, and has been studied under two alternative scenarios. Either the agents move in a network, modeled by an undirected connected graph (the *graph scenario*), or they move in (some subset of) the plane (the *geometric scenario*).

In this paper we study the *asynchronous* version of the rendezvous problem, under both above scenarios: each agent designs its route and an adversary controls the speed of each agent, can vary this speed, stop the agent, or even move it back and forth, as long as the walk of the agent in each segment of its route is continuous, does not leave it and covers all of it. In the asynchronous version of the graph scenario, meeting at a node may be impossible even in the two-node graph, as the adversary can desynchronize the agents and make them visit nodes at different times. Thus it is necessary to relax the requirement and allow agents to meet either in a node or inside an edge. Such a definition of meeting is natural, e.g., when agents are robots traveling in a labyrinth. We consider an embedding of the underlying graph in the three-dimensional Euclidean space, with nodes of the graph being points of the space and edges being pairwise disjoint line segments joining them (hence there are no edge crossings). Agents are modeled as points moving inside this embedding.

If nodes of the graph are labeled and the labeling is known, then agents can decide to meet at a predetermined node and the rendezvous problem reduces to graph exploration. However, in many applications, when rendezvous is needed in a network of unknown topology, such unique labeling of nodes may be unavailable, or agents may be unable to perceive such labels, e.g., due to security reasons. Hence it is important to design rendezvous algorithms for agents operating in *anonymous* graphs, i.e., graphs without unique labeling of nodes. It is important to note that the agents have to be able to *locally* distinguish ports at a node: otherwise, an agent may even be unable to visit all neighbors of a node of degree 3 (after visiting the second neighbor, the agent cannot distinguish the port leading to the first visited neighbor from that leading to the unvisited one). Consequently, agents initially located at two nodes of degree 3, might never be able to meet. This justifies a common assumption made in the literature: all ports at a node are locally labeled by distinct positive integers. Degrees of nodes can be either finite or infinite. No coherence between those local labelings is assumed. When an agent leaves a node, it is aware of the port number by which it leaves and when it enters a node, it is aware of the entry port number. It can also verify, at each node, whether a given positive integer is a port number at this node. Agents know neither the graph, nor the initial distance between them. They cannot mark the nodes or the edges in any way. Rendezvous has to be accomplished regardless of local labelings of ports. Each agent terminates its walk at the time of meeting the other agent.

In the geometric scenario, we assume that the terrain in which agents operate is a closed subset of the Euclidean plane, i.e., it contains limits of all converging sequences of points in it. The *boundary* of the terrain is defined as the set of points having arbitrarily close points both in the terrain and outside of it. Since the terrain is closed, the boundary is included in it. All other points of the terrain are its *interior* points. Each agent can only distinguish if it is currently in an interior point of the terrain or in its boundary. Agents do not know the terrain in which they operate, they cannot “see” any vicinity of the currently visited point and they cannot leave any marks. Agents are equipped with a compass and with the same unit of length. Thus systems of coordinates of

agents are aligned and the origin for each agent is at its starting point. Again, an agent terminates its walk at the time of meeting the other agent.

If agents are identical, i.e., they do not have distinct identifiers, and execute the same algorithm, then deterministic rendezvous is impossible, e.g., in the ring (graph scenario) or in the plane (geometric scenario): the adversary will make the agents move always in the same direction at the same speed, keeping them at the same distance at all times, thus they will never meet. Hence we assume that agents have distinct identifiers, called labels, which are two different positive integers. This is their only way to break symmetry. We assume that each agent knows its own label but not the label of the other agent. This excludes, e.g., rendezvous strategies of the type “waiting for mommy”, in which the agent with smaller label remains idle and the other agent explores the graph or the terrain. We do not impose any restriction on the memory of the agents: from the computational perspective they are viewed as Turing machines.

Two important notions used to describe movements of agents are the *route* of each agent and its *walk*. Roughly speaking, each agent chooses the route *where* it moves and the adversary describes the walk on this route, deciding *how* the agent moves. More precisely, these notions are defined as follows. The adversary initially places an agent with label  $\ell$  at some node of the graph or at some point in the terrain. Given this label and this starting point, the route is chosen by the agent and is defined as follows. In the case of the graph, the agent chooses one of the available ports at the current node. After getting to the other end of the corresponding edge, the agent chooses one of the available ports at this node, and so on, indefinitely (until rendezvous). The resulting route of the agent is the corresponding sequence of edges, which is a (not necessarily simple) path in the graph. The route in a terrain is a sequence  $(S_1, S_2, \dots)$  of segments, where  $S_i = [a_i, a_{i+1}]$ , defined in stages as follows, given the agent’s label and the starting point. In stage  $i$  the agent starts at point  $a_i$ , and  $a_1$  is the starting point chosen by the adversary. The agent chooses a direction  $\alpha$  and distance  $x$ . If the segment of length  $x$  in direction  $\alpha$  starting in  $v$  intersects the boundary of the terrain at some distance  $y \leq x$  from  $v$ , the agent becomes aware of it in the intersection point  $w$  closest to  $v$ . In this case, the stage ends at  $w$  and in the next stage the agent chooses the reverse direction  $\bar{\alpha}$  and the distance  $y$  (that will cause it to return to  $v$ ). If the segment of length  $x$  in direction  $\alpha$  starting in  $v$  does not intersect the boundary of the terrain, the stage ends when the agent reaches point  $u$  at distance  $x$  from  $v$  in direction  $\alpha$ . Stages are repeated indefinitely (until rendezvous).

We now describe the walk  $f$  of an agent on its route. Let  $R = (S_1, S_2, \dots)$  be the route of an agent. In the graph scenario this is a (not necessarily simple) infinite path in the (spatial embedding of) the graph, and in the geometric scenario it is an infinite polygonal line in the plane. Let  $(t_1, t_2, \dots)$ , where  $t_1 = 0$ , be an increasing sequence of reals, chosen by the adversary, that represent points in time. Let  $f_i : [t_i, t_{i+1}] \rightarrow [a_i, a_{i+1}]$  be any continuous function, chosen by the adversary, such that  $f_i(t_i) = a_i$  and  $f_i(t_{i+1}) = a_{i+1}$ . For any  $t \in [t_i, t_{i+1}]$ , we define  $f(t) = f_i(t)$ . The interpretation of the walk  $f$  is as follows: at time  $t$  the agent is at the point  $f(t)$  of its route. This general definition of the walk and the fact that it is constructed by the adversary capture the asynchronous characteristics of the process. The movement of the agent can be at arbitrary speed, the agent may sometimes stop or go back and forth, as long as the walk in each segment of the route is continuous and covers all of it.

Notice that the power of the asynchronous adversary to produce any continuous walk on the routes determined by the agents implies the following significant difference with respect to the synchronous scenario. While in the latter scenario the relative movement of the agents depends only on their routes, in our setting, this movement is also controlled by the adversary.

Agents with routes  $R_1$  and  $R_2$  and with walks  $f_1$  and  $f_2$  meet at time  $t$ , if points  $f_1(t)$  and  $f_2(t)$  are identical. A rendezvous is guaranteed for routes  $R_1$  and  $R_2$ , if the agents using these routes meet at some time  $t$ , regardless of the walks chosen by the adversary. A rendezvous algorithm executed by agents in a graph or in a terrain produces routes of agents, given the label of each agent and its starting point.

It should be stressed that, while routes of agents are formally defined as infinite sequences of segments, our results imply that in any instance of the rendezvous problem, meeting will occur at some finite time, and thus each agent will compute only finitely many segments of its route. As mentioned above, agents compute their routes in stages, and given any walk chosen by the adversary, each stage is completed in finite time. There is no stopping issue in our solution: rendezvous always occurs at some stage for each of the agents and then both agents stop. Another feature of our rendezvous algorithms is that in the choice of consecutive segments of its route an agent does not use the knowledge of the walk to date. Thus the route depends only on the label of the agent, on the environment (graph or terrain), and on the starting point chosen by the adversary, but not on its other actions.

**Our results.** We give two deterministic algorithms, the first for rendezvous in the graph scenario and the second in the geometric scenario. For the graph scenario, our algorithm accomplishes rendezvous in any connected countable\* (finite or infinite) graph, for arbitrary starting nodes. A consequence of this very general result is the positive answer to the following question from [12]: Is deterministic asynchronous rendezvous feasible in any finite connected graph without knowing any upper bound on its size? (In [12] the authors presented a deterministic asynchronous rendezvous algorithm in arbitrary finite connected graphs with *known* upper bound on the size.)

For the geometric scenario, our algorithm accomplishes rendezvous for agents starting at any interior points with rational coordinates in any closed region of the plane with path-connected interior. (Recall that a subset  $T$  of the plane is path-connected, if for any points  $u, v \in T$ , there is a continuous function  $h : [0, 1] \rightarrow P$ , such that  $P \subseteq T$  and  $h(0) = u$ ,  $h(1) = v$ .) On the other hand, our algorithm guarantees the following *approximate rendezvous* for agents starting at *arbitrary* interior points of a terrain as above: agents will eventually get at an arbitrarily small positive distance from each other. This implies the perhaps surprising result that if agents have arbitrarily small positive visibility ranges (rather than 0 visibility range as we assume) and they start in arbitrary points of the (empty) plane, then they will see each other in finite time, regardless of the actions of the adversary.

**Discussion of limitations.** While our algorithms work in a very general setting – agents can, indeed, meet almost everywhere – it turns out that none of the few limitations imposed on the environment can be removed. For the graph scenario, the only limitation is connectivity of the graph. It is clear that rendezvous in disconnected graphs is impossible, if the agents start in different connected components. For the geometric scenario, let us review the limitations one by one. First, we assume that the terrain is closed. This assumption cannot be entirely removed for the following technical reason. Consider the construction of a route in an open disc. An agent starting at any point, that chooses in the first stage an arbitrary direction and a sufficiently large distance, at some point would have to leave the disc. Since it does not see anything in its vicinity, it cannot know where the boundary is before hitting it, and it cannot hit it, as it is not allowed to leave the terrain. It follows that the agent could not construct further segments of its route. The second assumption is that agents start at interior points of the terrain. This assumption cannot

---

\*A graph is countable if the set of its nodes is countable, i.e., if there exists a one-to-one function from this set into the set of natural numbers.

be removed either. Indeed, suppose that the terrain is a closed disc with a semi-circle attached to it. This is a closed subset of the plane with nonempty path-connected interior. Suppose that one agent starts in the disc and the other at the end of the semi-circle. Since agents need to move along polygonal lines, the second agent could not move at all and the first one cannot reach it. Our next assumption is that the interior of the terrain is path-connected. To show that this assumption cannot be removed, consider two disjoint closed discs joined by an arc of a circle. This terrain is closed and path-connected, but if each agent starts inside a different disc, again they cannot meet, because agents need to move along polygonal lines, and hence cannot traverse the joining arc. The final assumption is that the starting points of the agents have rational coordinates. In Section 4 we prove that if the agents start in *arbitrary* points, then rendezvous cannot be guaranteed even in the plane. We show, however, that for arbitrary starting points approximate rendezvous is guaranteed.

**Related work.** The rendezvous problem was first described in [25]. A detailed discussion of the large literature on rendezvous can be found in the excellent book [4]. Most of the results in this domain can be divided into two classes: those considering the geometric scenario (rendezvous in the line, see, e.g., [9, 10, 17], or in the plane, see, e.g., [7, 8]), and those discussing rendezvous in graphs, e.g., [2, 5]. Some of the authors, e.g., [2, 3, 6, 9, 18] consider the probabilistic scenario where inputs and/or rendezvous strategies are random. Randomized rendezvous strategies use random walks in graphs, which were thoroughly investigated and applied also to other problems, such as graph traversing [1], on-line algorithms [13] and estimating volumes of convex bodies [15]. A generalization of the rendezvous problem is that of gathering [16, 18, 19, 20, 23, 27], when more than 2 agents have to meet in one location.

If graphs are unlabeled, deterministic rendezvous requires breaking symmetry, which can be accomplished either by allowing marking nodes or by labeling the agents. Deterministic rendezvous with anonymous agents working in unlabeled graphs but equipped with tokens used to mark nodes was considered e.g., in [22]. In [28] the authors studied gathering many agents with unique labels. In [14, 21, 29] deterministic rendezvous in graphs with labeled agents was considered. However, in all the above papers, the synchronous setting was assumed. Asynchronous gathering under geometric scenarios has been studied, e.g., in [11, 16, 24] in different models than ours: agents could not remember past events, but they were assumed to have at least partial visibility of the scene. The first paper to consider deterministic asynchronous rendezvous in graphs was [12]. The authors concentrated on complexity of rendezvous in simple graphs, such as the ring and the infinite line. They also showed feasibility of deterministic asynchronous rendezvous in arbitrary finite connected graphs with *known* upper bound on the size. Further improvements of the above results for the infinite line were proposed in [26]. Gathering many robots in a graph, under a different asynchronous model and assuming that the whole graph is seen by each robot, has been studied in [19, 20].

## 2 Preliminary notions and results

A fundamental notion on which our algorithms are based is that of a *tunnel*. Consider any graph  $G$  and two routes  $R_1$  and  $R_2$  starting at nodes  $v$  and  $w$ , respectively. We say that these routes form a tunnel, if there exists a prefix  $[e_1, e_2, \dots, e_n]$  of route  $R_1$  and a prefix  $[e_n, e_{n-1}, \dots, e_1]$  of route  $R_2$ , for some edges  $e_i$  in the graph, such that  $e_i = \{v_i, v_{i+1}\}$ , where  $v_1 = v$  and  $v_{n+1} = w$ . Intuitively, the route  $R_1$  has a prefix  $P$  ending at  $w$  and the route  $R_2$  has a prefix which is the reverse of  $P$ , ending at  $v$ . By a slight abuse of terminology we will also say that prefixes  $[e_1, e_2, \dots, e_n]$  and  $[e_n, e_{n-1}, \dots, e_1]$  form a tunnel.

**Proposition 2.1** *If routes  $R_1$  and  $R_2$  form a tunnel, then they guarantee rendezvous.*

**Proof:** Consider an embedding of the graph  $G$  in the three-dimensional Euclidean space, with nodes of the graph being points of the space and edges being pairwise disjoint line segments joining them. Consider routes  $R_1$  and  $R_2$  starting at nodes  $v$  and  $w$ , respectively. Let agent  $a_i$  execute route  $R_i$ . Let  $P$  be the polygonal line joining  $v$  with  $w$ , corresponding to the prefixes of the routes, given by the tunnel. Let  $D$  be its length defined as the sum of lengths of edges in the corresponding prefixes of the routes. (For non-simple paths in the graph, the same edge is counted many times.) Consider any walks  $f_1$  on  $R_1$  and  $f_2$  on  $R_2$ . Let  $t'$  be the first moment when an agent leaves its starting point and let  $t''$  be the moment when an agent gets to the end of  $P$  other than its starting point. For any  $t \in [t', t'']$ , let  $d_1(t)$  be the distance of agent  $a_1$  from its starting point  $v$  at time  $t$ , counted on the route  $R_1$ , and let  $d_2(t)$  be the distance of agent  $a_2$  from its target point  $w$  at time  $t$ , counted on the route  $R_2$ . Let  $\delta(t) = d_2(t) - d_1(t)$ . We have  $\delta(t') = D$  and  $\delta(t'') = d \leq 0$ . The function  $\delta$  is thus a continuous function from the interval  $[t', t'']$  onto some interval  $[d', D]$ , where  $d' \leq d$ , in view of the continuity of walks  $f_1$  and  $f_2$ . Since 0 belongs to the interval  $[d', D]$ , there must exist a moment  $t$  in the interval  $[t', t'']$ , for which  $\delta(t) = 0$ . For this point we have  $f_1(t) = f_2(t)$ , and the rendezvous occurs.  $\square$

We now recall some basic facts from set theory, that will be used in further considerations.

**Proposition 2.2** *The set of rational numbers and the set of positive rational numbers are countable. The cartesian product of two countable sets is countable. The set of all finite sequences with terms in a countable set is countable.*

### 3 Rendezvous in the graph scenario

Let  $G = (V, E)$  be the connected graph in which the rendezvous must be performed. Let  $\mathcal{S}_n$  be the set of sequences of  $n$  positive integers. Let  $\mathcal{P} = \{(i, j, s', s'') \mid i, j \in \mathbb{N}, i < j \text{ and } \exists n \text{ s.t. } s', s'' \in \mathcal{S}_n\}$ . There exists a bijection from the set of positive integers onto  $\mathcal{P}$ , in view of Proposition 2.2. Let  $(\varphi_1, \varphi_2, \dots)$  be a fixed enumeration of  $\mathcal{P}$ . All agents have to agree on the same enumeration. It is easy to produce a formula computing  $\phi_k$  for any  $k$ . This formula is included in the rendezvous algorithm. For a finite path  $r$  in  $G$ , we denote by  $\bar{r}$  the path with the same edges as in  $r$ , but in the reverse order. Remark that  $r$  and  $\bar{r}$  form a tunnel.

We first give a high-level idea of the algorithm referring to lines of the pseudo code given below. We “force” the routes of any two agents to form a tunnel for every possible combination of starting nodes and labels of the two agents. By Proposition 2.1, this suffices to guarantee rendezvous. Any starting configuration of robot  $i$  placed at node  $v$  and robot  $j$  placed at node  $w$  by the adversary corresponds to a quadruple  $(i, j, s', s'')$  where  $s'$  is a sequence of ports inducing a path from  $v$  to  $w$  and  $s''$  is a sequence of ports inducing the reverse path from  $w$  to  $v$ .

Each agent constructs its route in phases. At the beginning and at the end of each phase the agent is in its starting node. At phase  $k$  the previously constructed initial part of the route  $r_{hist}$  is extended while the agent processes quadruple  $\varphi_k$  (some of the extensions are null). This extension guarantees that the routes of agents of the corresponding starting configuration will form a tunnel. When agent with label  $l$  processes quadruple  $\varphi_k = (i, j, s', s'')$  nothing happens if  $l \neq i$  and  $l \neq j$  (line 4). If  $l = i$ , agent  $i$  tries to extend its route to guarantee rendezvous with agent  $j$  under the hypothesis that a path from  $v$  to  $w$  corresponds to the sequence  $s'$  of ports and the reverse path corresponds to the sequence  $s''$ . For this to happen, the agent first tries to follow the path  $r(s')$  induced by the sequence  $s'$  of ports (lines 8-11). This attempt is considered successful if the following conditions are satisfied:

- at consecutive nodes of the traversed path, ports with numbers from the sequence  $s'$  are available,
- the reverse path corresponds to the sequence  $s''$  of ports.

When the attempt is successful (the condition of line 13 is satisfied) the agent is at node  $w$  and it simulates the first  $k - 1$  phases of the execution of the algorithm by agent with label  $j$  starting from  $w$ . The effect of this simulation is the path  $r_{sim}$ . Upon completion of this part, agent with label  $i$  returns to  $w$ . Now the agent is able to further extend its path to form a tunnel with the route of agent  $j$  (line 16). Finally, whether the attempt to follow the path  $r(s')$  is successful or not, the agent with label  $i$  backtracks to  $v$  (line 17). If  $l = j$ , the above actions are performed with the roles of  $i$  and  $j$  reversed and the role of  $s'$  and  $s''$  reversed.

Algorithm **GraphRV** calls the recursive function **GraphRVREC**. This function is called in two different modes controlled by the boolean  $mode$ . In the “main” mode ( $mode = true$ ) the function is executed indefinitely, until rendezvous. In the “simulation” mode ( $mode = false$ ), the function is executed for all values up to a given  $p$ , or until rendezvous, whichever comes first. The symbol  $\hat{\ } \circlearrowleft$  denotes the concatenation of sequences.

**Algorithm GraphRV**

INPUT: A starting node  $v \in V$  and a label  $l$  of the agent.

OUTPUT: A rendezvous route  $r$ .

**GraphRVREC**( $v, l, 0, true$ );

**function GraphRVREC**(node  $v$ , label  $l$ , integer  $p$ , boolean  $mode$ )

1  $k := 1; r := \lambda;$

2 **while** not rendezvous and ( $k \leq p$  or  $mode$ ) **do**

3   let  $\varphi_k = (i, j, s', s''); r_{hist} := r;$

4   **if**  $l = i$  or  $l = j$  **then**

5     **if**  $l = i$  **then**  $s_1 := s'; s_2 := s''; l' := j;$

6     **else**  $s_1 := s''; s_2 := s'; l' := i;$

7     let  $s_1 = (p_1, \dots, p_n); m := 1; r(s_1) := \lambda;$

8     **while**  $m \leq n$  and  $p_m$  is a port at the current node **do**

9        $r(s_1) := r(s_1) \hat{\ } \circlearrowleft (e_m)$  where  $e_m$  is the edge corresponding to port  $p_m$ ;

10      let  $a_m$  be the port corresponding to  $e_m$  at its other endpoint;

11       $m := m + 1;$

12      $r := r \hat{\ } \circlearrowleft r(s_1);$

13     **if**  $s_2 = (a_n, \dots, a_1)$  **then**

14       let  $w$  be the current node;

15        $r_{sim} := \text{GraphRVREC}(w, l', k - 1, false);$

16        $r := r \hat{\ } \circlearrowleft r_{sim} \hat{\ } \circlearrowleft \overline{r(s_1)} \hat{\ } \circlearrowleft \overline{r_{hist}} \hat{\ } \circlearrowleft r(s_1) \hat{\ } \circlearrowleft \overline{r_{sim}};$

17      $r := r \hat{\ } \circlearrowleft \overline{r(s_1)};$

18      $k = k + 1;$

19 **return**  $r$

**Theorem 3.1** *Algorithm GraphRV guarantees asynchronous rendezvous for arbitrary two agents starting from any nodes of an arbitrary connected graph.*

**Proof:** Let  $v$  and  $i$  (resp.  $w$  and  $j$ ) be the starting node and the label of the first agent (resp. the second agent). There exists a path  $q$  linking  $v$  to  $w$ , since the graph  $G$  is connected. Let  $s'$

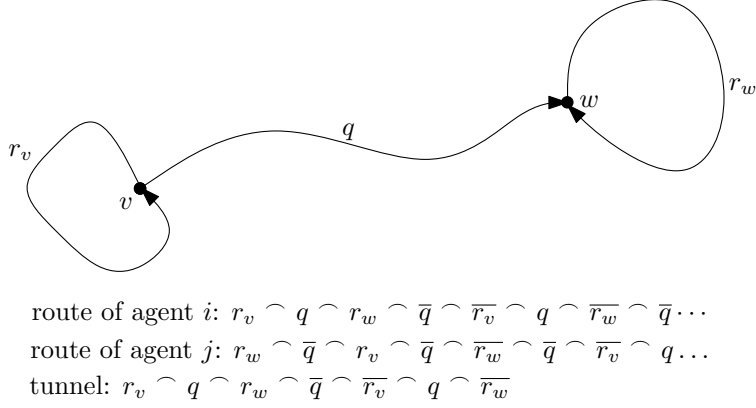


Figure 1: Tunnel between the routes of two agents

(resp.  $s''$ ) be the finite sequence of ports corresponding to the path  $q$  (resp. the path  $\bar{q}$ ). In view of Proposition 2.1, it suffices to prove that the routes of the two agents form a tunnel. We show that, after the phase corresponding to the quadruple  $\varphi_k = (i, j, s', s'')$  during the execution of Algorithm **GraphRV** for agents  $i$  and  $j$ , the routes of the two agents form a tunnel. Observe that this phase eventually occurs during any execution of **GraphRVREC**, since all recursive calls of any phase  $k' < k$  are done with a parameter  $p$  strictly smaller than  $k'$ . Thus all these phases are completed in finite time.

First, we show by induction that, at the beginning of phase  $k$  of any execution of Algorithm **GraphRV**, each agent is at its starting node. This is clearly true for  $k = 1$ . Assume that the property holds for  $k - 1$ . It follows that during the execution of the phase  $k - 1$ , the paths  $r_{hist}$  and  $r_{sim}$  are cycles. Hence, after the execution of line 16, the agent ends in node  $w$ . After the execution of line 17, the agent returns to the starting node  $v$  of the phase  $k - 1$ . So, the agent starts phase  $k$  in the same node, and the property is true for all  $k$ .

Let  $r_v$  (resp.  $r_w$ ) be the output of the execution of the first  $k - 1$  phases of Algorithm **GraphRV** for agent  $i$  (resp.  $j$ ) starting in node  $v$  (resp.  $w$ ). At the beginning of phase  $k$ , the portion of the route constructed by agent  $i$  is  $r_v$ . After the execution of line 12, the portion of the route constructed by agent  $i$  is  $r_v \frown q$ , since the agent has started the phase in node  $v$ . The path  $r_{sim}$  computed by the recursive call of **GraphRVREC** is equal to  $r_w$ . It follows that at the end of phase  $k$ , the portion of the route constructed by agent  $i$  is  $\rho = r_v \frown q \frown r_w \frown \bar{q} \frown \bar{r}_v \frown q \frown \bar{r}_w \frown \bar{q}$ . Similarly, at the end of phase  $k$ , the portion of the route constructed by agent  $j$  is  $\rho' = r_w \frown \bar{q} \frown r_v \frown q \frown \bar{r}_w \frown \bar{q} \frown \bar{r}_v \frown q$ . By construction, the part  $r_v \frown q \frown r_w \frown \bar{q} \frown \bar{r}_v \frown q \frown \bar{r}_w$  of  $\rho$  and the part  $r_w \frown \bar{q} \frown r_v \frown q \frown \bar{r}_w \frown \bar{q} \frown \bar{r}_v$  of  $\rho'$  form a tunnel (see Fig 1).  $\square$

## 4 Rendezvous in the geometric scenario

In this section we consider the problem of rendezvous in a terrain included in the Euclidean plane. As announced in the introduction, we restrict attention to closed subsets of the plane whose interior is path-connected. We observed that these restrictions cannot be removed.

Fix a system of coordinates  $\Sigma$  with the  $y$ -axis pointing to North (shown by compasses of the agents) and with the unit of length equal to that of the agents. Points with rational coordinates in  $\Sigma$  will



be called *rational*. A polygonal line all of whose vertices, including extremities, are rational will be called a *rational line*. For any point  $u$ , let  $\Sigma_u$  be the shift of the system  $\Sigma$  with origin at point  $u$ .

**Lemma 4.1** *In any path-connected, open subset  $S$  of the plane and for any rational points  $u, v \in S$ , there exists a rational polygonal line included in  $S$ , with extremities  $u$  and  $v$ , all of whose vertices are rational.*

**Proof:** By path-connectivity of  $S$ , there exists a path  $p$  included in  $S$  with extremities  $u$  and  $v$ , which is a continuous image of the interval  $[0, 1]$ . Let  $d$  be the distance from  $p$  to  $cS$  - the complement of  $S$ . Since  $p \cap cS = \emptyset$  and both  $p$  and  $cS$  are closed sets, we have  $d > 0$ . Partition the plane into squares of side length at most  $d/2$  with rational vertices. Let  $Q$  be the set of squares intersecting  $p$ . Since  $p$  is a bounded set,  $Q$  is finite. Consider the graph  $G_p$  with node set  $Q$ , such that  $x, y \in Q$  are adjacent if  $p$  contains a point belonging to a common boundary of  $x$  and  $y$ . Since  $G_p$  is connected, there exists a path  $(x_1, \dots, x_k)$  in  $G_p$  linking squares  $x_1$  containing  $u$  and  $x_k$  containing  $v$ . Let  $p^*$  be the polygonal path  $(\overline{uw_1}, \overline{w_1w_2}, \dots, \overline{w_{k-1}w_k}, \overline{w_kv})$ , where  $w_i$  is the center of square  $x_i$ , for all  $i = 1, \dots, k$ . The path  $p^*$  is rational and contained in the union of squares from  $Q$ . Since each point of  $p^*$  is at distance at most  $d\sqrt{2}/2$  from some point of  $p$ , we have  $p^* \cap cS = \emptyset$ , hence  $p^*$  is included in  $S$ .  $\square$

We define the following graph  $G_T = (V, E)$ , for a given terrain  $T$ . The set of nodes  $V$  is the union of two disjoint subsets  $V_1, V_2$ . The set  $V_1$  is the set of all interior, rational points of  $T$  and the set  $V_2$  is defined below.

For each pair of points  $p_1, p_2$ , such that  $p_1 \in V_1$  and  $p_2$  is any rational point of the plane, we consider the segment  $s = \overline{p_1p_2}$ . If  $s$  does not intersect the boundary of  $T$ , then  $p_2$  must belong to  $V_1$  and we add the edge  $\{p_1, p_2\}$  to  $E$ . If  $s$  intersects the boundary of  $T$ , we add a new node  $v$  to  $V_2$  and we add the edge  $\{p_1, v\}$  to  $E$ . Note that such a node  $v$  is always added to  $V_2$  when point  $p_2$  is on the boundary of  $T$  or outside of  $T$  (and it may or may not be added to  $V_2$  when  $p_2$  is in the interior of  $T$ ). Since each node in  $V_2$  corresponds to a pair of rational points  $p_1, p_2$ , there is a countable number of nodes in  $V_2$ , each of them having degree 1. The unique port at any node in  $V_2$  has number 1 and ports at any node in  $V_1$  are defined as follows. Let  $(z_1, z_2, \dots)$  be any fixed enumeration of all pairs of rational numbers. Let  $z_i = (q_1, q_2)$ . Let  $u$  be the point in the plane with coordinates  $(q_1, q_2)$  in the system  $\Sigma_p$ . The port at  $p$  corresponding to edge  $\{p, u\}$  has number  $i$ .

### Algorithm GeometricRV

The algorithm is a direct application of Algorithm **GraphRV** to the graph  $G_T$ . The agent operating in an unknown terrain  $T$  designs a route in the corresponding unknown graph  $G_T$  as follows. When the agent chooses a port at a node  $p \in V_1$ , this edge corresponds to some rational point  $q_i$  in the plane that the agent tries to reach from  $p$ . Two cases may occur. The agent either walks in the interior of  $T$  until reaching  $q_i$ , which corresponds to the traversal of an edge between two nodes of  $V_1$ , or it hits the boundary of  $T$ , which corresponds to a visit of a node  $p' \in V_2$ . At a node  $p' \in V_2$  there is no choice of port, since its degree is 1. The agent takes the unique port which leads to the (already visited) node  $p \in V_1$ . The resulting route is a sequence of segments joining rational interior points of the terrain  $T$  and of pairs of consecutive segments  $(\overline{vb}, \overline{bv})$ , where  $v$  is a rational interior point of  $T$  and  $b$  is a point on the boundary of  $T$ .

Since by Lemma 4.1 graph  $G_T$  is connected, rendezvous is guaranteed in the graph  $G_T$ , which implies rendezvous in  $T$ .

**Theorem 4.1** *Algorithm GeometricRV guarantees asynchronous rendezvous for arbitrary two agents starting from arbitrary rational interior points of any closed terrain  $T$  with path-connected interior.*

Theorem 4.1 should be contrasted with the following negative result showing that the restriction on the starting points of the agents cannot be removed, even for rendezvous in the (empty) plane.

**Proposition 4.1** *There is no algorithm that guarantees asynchronous rendezvous of arbitrary agents starting from arbitrary points in the plane.*

**Proof:** Consider the agent with label  $\ell$  operating in the empty plane. Since the terrain is fixed, the route of this agent depends only on the starting point. Let  $R = (e_1, e_2, \dots)$  be the route of the agent with label 1 starting at a fixed point  $v$ . Consider the route  $R_2(w)$  of the agent with label 2 starting at point  $w$ . Since there are no boundary points in the terrain, for any starting points  $w'$  and  $w''$ , route  $R_2(w'')$  is a parallel shift of route  $R_2(w')$  by the vector  $(w', w'')$ . Both of them are polygonal lines.

We will say that two routes are *almost disjoint* if all vertices of each of them are outside the other route. Observe that if, for some starting point  $w$ , route  $R_2(w)$  of agent 2 is almost disjoint from route  $R$  of agent 1, then the adversary can avoid rendezvous of agents 1 and 2 following these routes, by first moving agent 1 to the end of the first segment of its route, then moving agent 2 to the end of the first segment of its route and so on, alternating traversals of agents on consecutive segments of their routes. Hence, in order to prove our result, it is enough to show the existence of a point  $w^*$ , such that route  $R_2(w^*)$  is almost disjoint from route  $R$ .

Let  $(f_1, f_2, \dots)$  be the sequence of vectors corresponding to consecutive segments of the route  $R_2(w)$ , for any starting point  $w$ . For any fixed point  $w$ , denote by  $f_j(w)$  the segment corresponding to vector  $f_j$  on route  $R_2(w)$ . Let  $(p_1, p_2, \dots)$  be any sequence that orders all couples  $(e_i, f_j)$ , for all positive integers  $i, j$ . For any  $k$ , let  $p_k = (e_{i_k}, f_{j_k})$ . We will construct by induction a descending sequence of closed discs  $(D_1, D_2, \dots)$  of positive radii, satisfying the following invariant. For all points  $w \in D_k$ , both endpoints of the segment  $e_{i_k}$  are outside of the segment  $f_{j_k}(w)$  and both endpoints of the segment  $f_{j_k}(w)$  are outside of the segment  $e_{i_k}$ . Suppose that the invariant is satisfied for  $k - 1$  (for  $k = 1$  we may take as  $D_0$  any disc with radius 1). The set of points  $w$  inside disc  $D_{k-1}$  that may possibly violate the invariant for  $k$  is contained in the union of four segments: two segments parallel to  $f_{j_k}$  and two segments parallel to  $e_{i_k}$ . There exists a closed disc of positive radius contained in  $D_{k-1}$  which is disjoint from those four segments. Let  $D_k$  be such a disc. Thus the invariant is satisfied for  $k$ . This completes the construction by induction.

The intersection of all discs  $D_k$  is non-empty. Let  $w^*$  be a point in this intersection. Since  $(p_1, p_2, \dots)$  enumerated all couples  $(e_i, f_j)$ , it follows that route  $R_2(w^*)$  is almost disjoint from route  $R$ , and hence agents 1 and 2 starting at points  $v$  and  $w^*$ , respectively, do not meet for some walks chosen by the adversary.  $\square$

While Proposition 4.1 shows that rendezvous of agents starting from arbitrary points is impossible, it turns out that a slightly easier task can be accomplished in this setting. For any  $\epsilon > 0$ , we say that routes  $R_1$  and  $R_2$  of agents guarantee  $\epsilon$ -*approximate rendezvous*, if at some point  $t$  of time, the agents get at distance at most  $\epsilon$  from each other, regardless of the walks chosen by the adversary.

**Theorem 4.2** *Algorithm GeometricRV guarantees  $\epsilon$ -approximate rendezvous for any  $\epsilon > 0$ , for arbitrary agents starting from arbitrary interior points of any closed terrain  $T$  with path-connected interior.*

**Proof:** Fix an  $\epsilon > 0$ . Consider any two agents starting at interior points  $v$  and  $w$  of the terrain  $T$ . Let  $\rho > 0$  be the distance from  $w$  to the boundary of  $T$ . Choose a point  $w'$  with rational coordinates in  $\Sigma_v$ , in the interior of the disc  $D$  of radius  $\rho/2$  centered at  $w$ . By Lemma 4.1 (applied to the system  $\Sigma_v$  instead of  $\Sigma$ ), there exists a rational polygonal line  $P$  in the system  $\Sigma_v$ , included in the interior of  $T$  and joining  $v$  and  $w'$ . Let  $d > 0$  be the distance between  $P$  and the boundary of  $T$ . Let  $r = \min(\rho/2, d/2, \epsilon)$ . Choose a point  $w''$  with rational coordinates in the system  $\Sigma_v$ , at distance less than  $r$  from  $w$ . Let  $P_v$  be the polygonal line  $P$  extended by the segment  $w'w''$ . Note that  $P_v$  is at distance at least  $r$  from the boundary of  $T$ . Indeed, if  $x \in P$ , then the distance from  $x$  to the boundary is at least  $d > r$ , and if  $x \in \overline{w'w''}$ , then the distance from  $x$  to the boundary is at least  $\rho/2 \geq r$ , as the entire segment  $\overline{w'w''}$  is included in disc  $D$ .

Let  $\Phi$  be the translation by the vector  $(w''w)$  and let  $P_w$  be the image of  $P_v$  with respect to  $\Phi$ . The point  $w$  is one of the extremities of  $P_w$ . Since the distance from  $w''$  to  $w$  is less than  $r$ , the entire polygonal line  $P_w$  is in the interior of  $T$ . The polygonal line  $P_w$  is rational in the system  $\Sigma_w$ .

Let  $f$  be any walk of the first agent on any route with the initial part  $P_v$  starting at  $v$ , and let  $g$  be any walk of the second agent on any route with the initial part  $P_w$  starting at  $w$ . Consider the composition  $g^* = \Phi^{-1} \circ g$ . Hence  $g^*$  is a walk on a route with initial part  $P_v$ , starting at  $w''$ . By Theorem 4.1, Algorithm `GeometricRV` guarantees rendezvous of agents at some point in time  $t$ , for the walk  $f$  starting at  $v$  and the walk  $g^*$  starting at  $w''$ .

Consider the positions at time  $t$  of both agents starting at  $v$  and  $w$ , in walks  $f$  and  $g$ . Since  $f(t) = g^*(t) = \Phi^{-1}(g(t))$ , and  $\Phi$  is a translation by a vector of length less than  $r$  (hence less than  $\epsilon$ ), it follows that the distance between  $f(t)$  and  $g(t)$  is less than  $\epsilon$ . Since walks  $f$  and  $g$  were arbitrarily chosen by the adversary, this guarantees  $\epsilon$ -approximate rendezvous.  $\square$

A consequence of Theorem 4.2 is that if agents have arbitrarily small positive visibility ranges (rather than 0 visibility range as we assumed) and they start in arbitrary points of the (empty) plane, then Algorithm `GeometricRV` guarantees that they will see each other in finite time, regardless of the actions of the adversary.

## 5 Conclusion

We provided deterministic asynchronous rendezvous algorithms for graphs and for terrains in the plane. We studied only the feasibility of rendezvous and our results are very general: for the graph scenario, we showed that rendezvous is possible in any connected countable (finite or infinite) graph, starting from any nodes, without any information on the graph. The only thing an agent needs to know is its own label. In particular, this result implies a positive solution of a problem from [12].

Our algorithms rely on an arbitrary fixed enumeration of quadruples  $(i, j, s', s'')$ , where  $i$  and  $j$  are positive integers and  $s'$  and  $s''$  are finite sequences of positive integers. The complexity of the algorithm (measured by the worst-case length of paths that the agents have to traverse until rendezvous) depends on this enumeration, but we do not think any enumeration can make the algorithm efficient. Thus a natural interesting question left for further investigations is the following:

Does there exist a deterministic asynchronous rendezvous algorithm, working for all connected countable unknown graphs, with complexity polynomial in the labels of the agents and in the initial distance between them?

## References

- [1] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, and C. Rackoff, Random walks, universal traversal sequences, and the complexity of maze problems, Proc. Annual Symposium on Foundations of Computer Science FOCS'1979, 218-223.
- [2] S. Alpern, The rendezvous search problem, SIAM J. on Control and Optimization 33 (1995), 673-683.
- [3] S. Alpern, Rendezvous search on labelled networks, Naval Research Logistics 49 (2002), 256-274.
- [4] S. Alpern and S. Gal, The theory of search games and rendezvous. Int. Series in Operations research and Management Science, number 55, Kluwer Academic Publishers, 2002. Kluwer Academic Publisher, 2002.
- [5] J. Alpern, V. Baston, and S. Essegaier, Rendezvous search on a graph, Journal of Applied Probability 36 (1999), 223-231.
- [6] E. Anderson and R. Weber, The rendezvous problem on discrete locations, Journal of Applied Probability 28 (1990), 839-851.
- [7] E. Anderson and S. Fekete, Asymmetric rendezvous on the plane, Proc. 14th Annual ACM Symp. on Computational Geometry, 365-373, 1998.
- [8] E. Anderson and S. Fekete, Two-dimensional rendezvous search, Operations Research 49 (2001), 107-118.
- [9] V. Baston and S. Gal, Rendezvous on the line when the players' initial distance is given by an unknown probability distribution, SIAM J. on Control and Optimization 36 (1998), 1880-1889.
- [10] V. Baston and S. Gal, Rendezvous search when marks are left at the starting points, Naval Res. Log. 48 (2001), 722-731.
- [11] M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, Solving the Robots Gathering Problem, Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP 2003), 1181-1196.
- [12] G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoretical Computer Science 355 (2006), 315-326.
- [13] D. Coppersmith, P. Doyle, P. Raghavan, and M. Snir, Random walks on weighted graphs, and applications to on-line algorithms, Proc. 22nd Annual ACM Symposium on Theory of Computing (STOC'1990), 369-378.
- [14] A. Dessmark, P. Fraigniaud, D. Kowalski, and A. Pelc, Deterministic rendezvous in graphs, Algorithmica 46 (2006), 69-96.
- [15] M. Dyer, A. Frieze, and R. Kannan, A random polynomial time algorithm for estimating volumes of convex bodies, Proc. 21st Annual ACM Symposium on Theory of Computing (STOC'1989), 375-381.

- [16] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous oblivious robots with limited visibility, Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science STACS'2001, LNCS 2010, 247-258.
- [17] S. Gal, Rendezvous search on the line, Operations Research 47 (1999), 974-976.
- [18] A. Israeli and M. Jalfon, Token management schemes and random walks yield self stabilizing mutual exclusion, Proc. PODC'1990, 119-131.
- [19] R. Klasing, A. Kosowski, A. Navarra, Taking advantage of symmetries: gathering of asynchronous oblivious robots on a ring. Proc. 12th International Conference on Principles of Distributed Systems, (OPODIS 2008), 446-462.
- [20] R. Klasing, E. Markou, A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, Theoretical Computer Science 390 (2008), 27-39.
- [21] D. Kowalski, A. Malinowski, How to meet in anonymous network. Theoretical Computer Science 399 (2008), 141-156.
- [22] E. Kranakis, D. Krizanc, N. Santoro and C. Sawchuk, Mobile agent rendezvous in a ring, Proc. 23rd International Conference on Distributed Computing Systems (ICDCS'2003), 592-599.
- [23] W. Lim and S. Alpern, Minimax rendezvous on the line, SIAM J. on Control and Optimization 34 (1996), 1650-1665.
- [24] G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoretical Computer Science 384 (2007), 222-231.
- [25] T. Schelling, The strategy of conflict, Oxford University Press, Oxford, 1960.
- [26] G. Stachowiak, Asynchronous Deterministic Rendezvous on the Line, SOFSEM 2009: 497-508.
- [27] L. Thomas, Finding your kids when they are lost, Journal on Operational Res. Soc. 43 (1992), 637-639.
- [28] X. Yu and M. Yung, Agent rendezvous: a dynamic symmetry-breaking problem, Proc. International Colloquium on Automata, Languages, and Programming (ICALP'1996), LNCS 1099, 610-621.
- [29] A. Ta-Shma, U. Zwick, Deterministic rendezvous, treasure hunts and strongly universal exploration sequences., Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), 599-608.