How to meet asynchronously (almost) everywhere *

Jurek Czyzowicz [†‡] Arnaud Labourel [§] Andrzej Pelc [†¶]

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 is continuous, does not leave its route 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. In the geometric scenario agents may have different compasses and different units of length. 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 to within an arbitrarily small positive distance from each other.

keywords: rendezvous, graph, asynchronous, deterministic

*A preliminary version of this paper appeared in Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010).

[†]Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mails: jurek@uqo.ca, pelc@uqo.ca

[‡]Partially supported by NSERC discovery grant.

[§]Université de Provence (LIF), 13453 Marseille Cedex 13, France. E-mail: arnaud.labourel@lif.univ-mrs.fr
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) 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 have unique labels 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 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 consecutive positive integers. Degrees of nodes can be either finite or infinite. Ports at a node of finite degree d are enumerated $1, \ldots, d$ and ports at a node of infinite degree are enumerated $1, 2, \ldots$. Local labelings of ports at each node are fixed: every agent sees the same local labeling at each node of the graph. However, no coherence between those local labelings over all the graph 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.

*Notice that this definition of the adversary is very strong. In fact, all our positive results (algorithms) are valid even with this powerful adversary, and our negative results hold even for a weaker adversary that can only speed up or slow down the agent, without moving it back.

In the geometric scenario, we assume that the terrain in which agents operate is a closed subset of the Euclidean plane, i.e., contains all limits of 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. Each agent has its own North direction, orientation and its own unit of length. These directions, orientations and units may be different for different agents. 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 the agent and its *walk*. Roughly speaking, the 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 ℓ 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 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 α and distance x according to its own North direction, orientation and using its own unit of length. If the segment of length x in direction α 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 $\overline{\alpha}$ and the distance y (that will cause it to return to v). If the segment of length x in direction α 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 α. 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 $f_i : [t_i, t_{i+1}] \to [a_i, a_{i+1}]$ 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 a finite initial segment 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 [14]: Is deterministic asynchronous rendezvous feasible in any finite connected graph without knowing any upper bound on its size? (In [14] 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 arbitrary 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 $h : [0, 1] \longrightarrow P$, such that $P \subseteq T$ and $h(0) = u$, $h(1) = v$.) Moreover,

we do not assume that the North directions, orientations and units of length of the agents are the same: our rendezvous algorithm in the geometric scenario works, if for each agent, the angle between its North direction and the y-axis is any rational multiple of π, the orientation of each agent (clockwise or anticlockwise) is arbitrary, and the unit of length of each agent is rational. 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 to within 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 be removed either. Indeed, suppose that the terrain is a closed disc with an added semi-circle, one of whose ends is attached to the boundary of the disc. 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 could 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 [27]. 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., [10, 11, 19], 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, 10, 20] 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 [15] and estimating volumes of convex bodies [17].

A generalization of the rendezvous problem is that of gathering [18, 20, 21, 22, 25, 29], 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 [24]. In [30] the authors studied gathering many agents with unique labels. In [16, 23, 31] 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 [12, 18, 26] 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 [14]. 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 [28]. In recent work [9, 13] (published after the conference version of the present paper) the authors consider the asynchronous rendezvous problem in grids and in the plane under the additional assumption that each agent knows its initial position with respect to some common system of coordinates. In this stronger model they show that agents starting in arbitrary positions in the plane can get at distance 1 from each other at cost $O(d^2 polylog(d))$, where d is the initial distance between the agents. 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 [21, 22].

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 $[e_1, e_2, \ldots, e_n]$ of route R_1 and a prefix $[e_n, e_{n-1}, \ldots, e_1]$ 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 $[e_1, e_2, \ldots, e_n]$ and $[e_n, e_{n-1}, \ldots, e_1]$ 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 v 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 δ 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. □

Observe that a tunnel is not always necessary to guarantee rendezvous. Indeed, although a correspondence between the edges of the routes of the agents must exist to guarantee the rendezvous, there is no need for this correspondence to be a bijection. For instance, consider a path of length three in a graph. The first agent is on the left end of the path and its route is formed of the concatenation of three copies of the following sequence: a step to the right, a step to the left and then a step to the right. If the second agent is initially located at the right end of the path and its route consists of three steps to the left, then the rendezvous always occurs, although no tunnel is present.

We denote by $\mathbb{N}$ the set of positive integers.

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 $\mathcal{P} = \{(i, j, s', s'') \mid i, j \in \mathbb{N}, i < j \text{ and } \exists n \text{ s.t. } s', s'' \in \mathcal{S}_n\}$. We define the following bijection from the set of positive integers onto $\mathcal{P}$. Consider the function $f : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ given by the formula $f(x, y) = x+(x+y-1)(x+y-2)/2$. The function f is a bijection from $\mathbb{N}\times\mathbb{N}$ to $\mathbb{N}$. Let $g : \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N} \to \mathbb{N}$ be given by the formula $g(x, y, z, t) = f(f(x, y), f(z, t))$. The function g is a bijection from $\mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$. Now the enumeration $\varphi_1, \varphi_2, \ldots$ of $\mathcal{P}$ is defined by the formula $\varphi_i = g^{-1}(i)$.

For a finite path r in G, we denote by $\overline{r}$ the path with the same edges as in r, but in the reverse order. Remark that r and $\overline{r}$ 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.

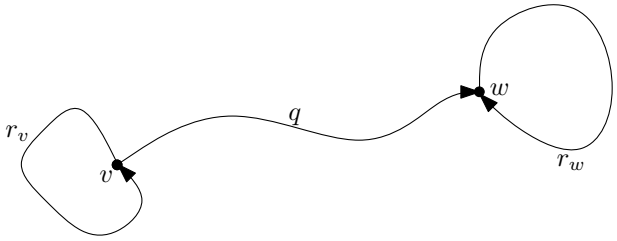
Each agent constructs its route in phases. At the beginning and at the end of each phase the agent is in its starting node. At phase k the previously constructed initial part of the route r_{hist} is extended while the agent processes quadruple φ_k (some of the extensions are null). This extension

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 $main$. In the "main" mode ($main = true$) the function is executed indefinitely, until rendezvous. In the "simulation" mode ($main = false$), the function is executed for all values up to a given p, or until rendezvous, whichever comes first. The symbol $\frown$ denotes the concatenation of sequences of segments. λ denotes the empty sequence of segments starting and ending at the current node. The algorithm outputs a rendezvous route r.



route of agent i:
$r_v \frown q \frown r_w \frown \overline{q} \frown \overline{r_v} \frown q \frown \overline{r_w} \frown \overline{q} \cdots$
route of agent j:
$r_w \frown \overline{q} \frown r_v \frown q \frown \overline{r_w} \frown \overline{q} \frown \overline{r_v} \frown q \ldots$
tunnel:
$r_v \frown q \frown r_w \frown \overline{q} \frown \overline{r_v} \frown q \frown \overline{r_w}$

Figure 1: Tunnel between the routes of two agents

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 $main$)
1 $k := 1; r := \lambda$;
2 **while** not rendezvous and ($k \leq p$ or $main$) **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 $s_1 = (p_1, \ldots, p_n)$; $m := 1$; $r_{s_1} := \lambda$;
8 **while** $m \leq n$ and p_m is a port at the current node **do**
9 $r_{s_1} := r_{s_1} \frown (e_m)$ where e_m is the edge corresponding to port p_m;
10 let b_m be the port corresponding to e_m at its other endpoint;
11 $m := m + 1$;
12 $r := r \frown r_{s_1}$;
13 **if** $s_2 = (b_n, \ldots, b_1)$ **then**
14 let w be the current node;
15 $r_{sim} :=$ `GraphRVREC`$(w, l', k-1, false)$;
16 $r := r \frown r_{sim} \frown \overline{r_{s_1}} \frown \overline{r_{hist}} \frown r_{s_1} \frown \overline{r_{sim}}$;
17 $r := r \frown \overline{r_{s_1}}$;
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 countable connected graph (finite or infinite).*

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' (resp. s'') be the finite sequence of ports corresponding to the path q (resp. the path $\overline{q}$). 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 $\rho = r_v \frown q \frown r_w \frown \overline{q} \frown \overline{r_v} \frown q \frown \overline{r_w} \frown \overline{q}$. Similarly, at the end of phase k, the portion of the route constructed by agent j is $\rho' = r_w \frown \overline{q} \frown r_v \frown q \frown \overline{r_w} \frown \overline{q} \frown \overline{r_v} \frown q$. By construction, the part $r_v \frown q \frown r_w \frown \overline{q} \frown \overline{r_v} \frown q \frown \overline{r_w}$ of ρ and the part $r_w \frown \overline{q} \frown r_v \frown q \frown \overline{r_w} \frown \overline{q} \frown \overline{r_v}$ of ρ' form a tunnel (see Fig 1).

In order to construct the reverse paths $\overline{q}$, $\overline{r_v}$, and $\overline{r_w}$, when the agent is traversing one of the paths q, r_v, or r_w, each time it reaches a new node, it stores the entry port number of the edge from which it arrives. The respective reverse path is obtained by taking the sequences of entry port numbers in the reverse order. □

4 Rendezvous in the geometric scenario

In this section we consider the problem of rendezvous in a terrain included in the 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.

For any system of coordinates σ, define North of σ to be the positive direction of its y-axis. Fix an arbitrary system of coordinates Σ. Points with rational coordinates in this system will be called *rational*. A system of coordinates is said to be *aligned* if it has the same North, orientation and unit of length as Σ and its origin is at a rational point. For any point u, let Σ_u be the shift of the system Σ 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 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 $(x_1, \ldots, x_k)$ in G_p linking squares x_1 containing u and x_k containing v. Let p^* be the polygonal path $(\overline{uw_1}, \overline{w_1w_2}, \ldots, \overline{w_{k-1}w_k}, \overline{w_kv})$, where w_i is the center of square x_i, for all $i = 1, \ldots, k$. 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. □

We first give a rendezvous algorithm for agents with an aligned system of coordinates with origin at their starting points. For this purpose 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 $s = \overline{p_1 p_2}$. 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 a fixed recursive 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 Σ_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 $p \in V_1$, the other end of the corresponding edge is some rational point u_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 u_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.

We now consider the generalized version of the rendezvous problem in the geometric scenario. Agents may have different systems of coordinates. More precisely the angle between their North directions can be $q\pi$, for any rational number q, the unit of length of each agent can be any positive rational and the orientation can be either clockwise or counterclockwise.

For $p, q \in \mathbb{Q}$, we consider the transformation $\tau_{q,p}$ that is the composition of the rotation by angle $q\pi$ and scaling by the factor of p and the transformation $\tau'_{q,p}$ that is the composition of $\tau_{q,p}$ and of the reflection with respect to the x-axis. Let $\mathcal{T} = \{\tau_{q,p}, \tau'_{q,p} \mid p, q \in \mathbb{Q}\}$. A system of coordinates σ is *acceptable*, if there exists $t \in \mathcal{T}$ such that $\sigma = t(\Sigma)$.

The following algorithm is a generalization of Algorithm `GeometricRV` in which agents can have

acceptable systems of coordinates instead of aligned systems of coordinates.

Algorithm `ExtendedGeometricRV`

The algorithm is a modification of Algorithm `GeometricRV`. First, the graph $G'_T = (V', E')$ used to represent the terrain T is different from G_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 internal points of T having rational coordinates in any acceptable system of coordinates. The set V'_2 (resp. E') is defined similarly as V_2 (resp. E) in G_T but with V'_1 replacing V_1. More precisely, in the definition of V'_2, instead of choosing as p_2 any rational point of the plane, as it was done in the definition of V_2, we choose as p_2 any point with rational coordinates in any acceptable system of coordinates. Let $(z'_1, z'_2, \dots)$ be a fixed recursive enumeration of triples composed of two rational numbers and of an element of $\mathcal{T}$. Let $z'_i = (q_1, q_2, t)$. Let u be the point in the plane with coordinates (q_1, q_2) in the system of coordinates $t(\sigma)$ with origin at p. Unlike in Algorithm `GeometricRV` the port numbering generated by each agent at p is not unique, but depends on the system of coordinates of the agent. The port at p corresponding to edge $\{p, u\}$ has number i in the system of coordinates σ.

The other difference is that during each phase the agent will process sextuples instead of quadruples. The set of sextuples considered is $\mathcal{P}' = \{(i, j, s', s'', t', t'') \mid i, j \in \mathbb{N}, i < j$ and $\exists n$ s.t. $s', s'' \in \mathcal{S}_n$ and $t', t'' \in \mathcal{T}\}$. We enumerate $\mathcal{P}'$ as follows. Consider the function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ given by the formula $f(x, y) = x + (x + y - 1)(x + y - 2)/2$. Let $g' : \mathbb{N}^6 \to \mathbb{N}$ be given by the formula $g'(u, v, x, y, z, t) = f(f(u, v), f(f(x, y), f(z, t)))$. Now the enumeration $\psi_1, \psi_2, \dots$ of $\mathcal{P}'$ is defined by the formula $\psi_i = g'^{-1}(i)$.

During phase k corresponding to $\psi_k = (i, j, s', s'', t', t'')$, agent i with the system of coordinates σ computes the path $r_{s'}$ corresponding to the sequence of ports s' in the system of coordinates $t'(\sigma)$ and simulates the $k - 1$ first phases of agent j in the system of coordinates $(t'')^{-1}(t'(\sigma))$. Agent j acts in an analogous way, replacing s' by s'' and switching the roles of t' and t''. $\Diamond$

Theorem 4.1 *Algorithm* `ExtendedGeometricRV` *guarantees asynchronous rendezvous for arbitrary two agents with acceptable systems of coordinates and starting from arbitrary rational interior points of any closed terrain T with path-connected interior.*

Proof: Let v, i and σ' (resp. w, j and σ'') be the starting point, the label and the system of coordinates of the first agent (resp. the second agent). There exists a path r_s with rational vertices in the system of coordinates Σ linking v to w by Lemma 4.1. Let s' (resp. s'') be the finite sequence of ports corresponding to the route r_s (resp. the route $\overline{r_s}$) in the system of coordinates Σ. Let t' (resp. t'') be the transformation such that $\Sigma = t'(\sigma')$ (resp. $\Sigma = t''(\sigma'')$). We show that after the phase corresponding to the sextuple $\psi_k = (i, j, s', s'', t', t'')$ during the executions of Algorithm `ExtendedGeometricRV` for agents i and j, the routes of the two agents form a tunnel. Let r_v (resp. r_w) be the output of the execution of the first $k-1$ phases of Algorithm `ExtendedGeometricRV` for agent i (resp. j) starting at node v (resp. w) with system of coordinates σ' (resp. σ''). At the end of phase k, the portion of the route constructed by agent i is $\rho = r_v \frown r_s \frown r_w \frown \overline{r_s} \frown \overline{r_v} \frown r_s \frown \overline{r_w} \frown \overline{r_s}$, since $\Sigma = t'(\sigma') = t''(\sigma'')$. Similarly, at the end of phase k, the portion of the route constructed

by agent j is $\rho' = r_w \frown \overline{r_s} \frown r_v \frown r_s \frown \overline{r_w} \frown \overline{r_s} \frown \overline{r_v} \frown r_s$. Hence the routes form a tunnel, which guarantees asynchronous rendezvous. □

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 and even when agents have the same North, the same orientation and the same unit of length.

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

Proof: Consider an agent with a fixed label operating in the empty plane. Since the terrain is fixed, the route of this agent depends only on the starting point. Let $R = (e_1, e_2, ...)$ 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''). Indeed, since the algorithm of the agent is deterministic and the terrain seen by the agent starting from w' or w'' is the same, the routes traversed will be the same up to the parallel shift between the two initial positions of the agent. 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 $(f_1, f_2, ...)$ 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. □

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 *ϵ-approximate rendezvous*, if at some point t of time, the agents get at distance at most ϵ from each other, regardless of the walks chosen by the adversary.

Theorem 4.2 *Algorithm* `ExtendedGeometricRV` *guarantees ϵ-approximate rendezvous for any $\epsilon > 0$, for arbitrary two agents, with acceptable systems of coordinates, starting from arbitrary points of any closed terrain T with path-connected interior.*

Proof: Consider any $\epsilon > 0$. Take 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 Σ_v, in the interior of the disc D of radius $\rho/2$ centered at w. By Lemma 4.1 (applied to the system Σ_v instead of Σ), there exists a rational polygonal line P in the system Σ_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 Σ_v, at distance less than r from w. Let P_v be the polygonal line P extended by the segment $\overline{w'w''}$. 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 Φ be the translation by the vector $(w''w)$ and let P_w be the image of P_v with respect to Φ. 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 Σ_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 `ExtendedGeometricRV` 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 Φ is a translation by a vector of length less than r (hence less than ϵ), it follows that the distance between $f(t)$ and $g(t)$ is less than ϵ. Since walks f and g were arbitrarily chosen by the adversary, this guarantees ϵ-approximate rendezvous. □

A consequence of Theorem 4.2 is that if agents have arbitrarily small positive visibility ranges (rather than 0 visibility range as we assumed), have acceptable systems of coordinates and start from arbitrary points of the (empty) plane, then Algorithm `ExtendedGeometricRV` 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. In this paper 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 [14].

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, and more precisely on the position of the quadruple φ_l corresponding to the initial configuration of the agents. During each phase k of the algorithm, the length of the routes of the two agents corresponding to the quadruple φ_k, is at least doubled. Hence, the complexity of the algorithm is at least exponential in terms of the number of quadruples with the same labels as those of the two agents, that are before φ_l in the enumeration. Thus a natural interesting question left for further investigations is the following:

> Does there exist a deterministic asynchronous rendezvous algorithm, working for all unknown connected finite graphs, with complexity polynomial in the labels of the agents and in the size of the graph?

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 Reaserch 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] E. Bampas, J. Czyzowicz, L. Gasieniec, D. Ilcinkas, A. Labourel, Almost optimal asynchronous rendezvous in infinite multidimensional grids, Proc. 24th International Symposium on Distributed Computing (DISC 2010), LNCS 6343, 297-311.

[10] 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.

[11] V. Baston and S. Gal, Rendezvous search when marks are left at the starting points, Naval Res. Log. 48 (2001), 722-731.

[12] 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.

[13] A. Collins, J. Czyzowicz, L. Gasieniec, A. Labourel, Tell me where I am so I can meet you sooner, Proc. 37th International Colloquium on Automata, Languages, and Programming (ICALP 2010), LNCS 6199, 502-514.

[14] 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.

[15] 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.

[16] A. Dessmark, P. Fraigniaud, D. Kowalski, and A. Pelc, Deterministic rendezvous in graphs, Algorithmica 46 (2006), 69-96.

[17] 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.

[18] 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.

[19] S. Gal, Rendezvous search on the line, Operations Research 47 (1999), 974-976.

[20] A. Israeli and M. Jalfon, Token management schemes and random walks yield self stabilizing mutual exclusion, Proc. 9th Annual ACM Symposium on Principles of Distributed Computing (PODC 1990), 119-131.

[21] 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.

[22] R. Klasing, E. Markou, A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, Theoretical Computer Science 390 (2008), 27-39.

[23] D. Kowalski, A. Malinowski, How to meet in anonymous network. Theoretical Computer Science 399 (2008), 141-156.

[24] 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.

[25] W. Lim and S. Alpern, Minimax rendezvous on the line, SIAM J. on Control and Optimization 34 (1996), 1650-1665.

[26] G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoretical Computer Science 384 (2007), 222-231.

[27] T. Schelling, The strategy of conflict, Oxford University Press, Oxford, 1960.

[28] G. Stachowiak, Asynchronous deterministic rendezvous on the line, Proc. 35th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2009), LNCS 5404, 497-508.

[29] L. Thomas, Finding your kids when they are lost, Journal on Operational Res. Soc. 43 (1992), 637-639.

[30] 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.

[31] 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.