Deterministic Rendezvous with Different Maps

Ashley Farrugia Leszek Gąsieniec Łukasz Kuszner Eduardo Pacheco

August 19, 2019

Abstract

We consider a rendezvous problem in which two identical anonymous mobile entities A and B, called later *robots*, are asked to meet at some node in the network modelled by an arbitrary undirected graph $G = (V, E)$. Most of the work devoted to rendezvous in graphs assumes robots have access to the same sets of nodes and edges, and the topology of connections is either known or unknown to the robots. In this work we assume that each robot may access only specific nodes and edges in G of which full map is given to the robot in advance. We consider *three variants of rendezvous* differentiated by the level of restricted maneuverability of robots in both synchronous and asynchronous models of computation. In each adopted variant and model of computation we study feasibility of rendezvous, and if rendezvous is possible we propose the relevant algorithms and discuss their efficiency.

1 Introduction

In this paper we consider several variants of the rendezvous problem, a combinatorial or geometric challenge in which two mobile entities, called later *robots*, are asked to meet at the same point and time in space. Usually the search space is represented either by a network of discrete nodes between which robots can move along existing connections (edges) or by a geometric environment in which movement of robots is restricted by the topological properties of the space.

The rendezvous problem has been studied in many different settings and mainly with the emphasis on trade-offs between the knowledge of the robots and the efficiency of the proposed solutions [36, 37]. In some cases, e.g., in *symmetric* systems populated by *anonymous* (indistinguishable) robots where the tools and advice given to each robot are identical, deterministic rendezvous may not be feasible [6]. In this context, any evidence (e.g., computed in due course) helping to distinguish between participating robots very often prove to be vital in achieving rendezvous. For instance, Feinerman *et al.* [30] adopted different speeds of robots to enable rendezvous in anonymous rings. However, one should be aware of the fact that heterogeneity of robots increases uncertainty in the system and in turn deteriorates efficiency of the rendezvous process. Such phenomena have been recently studied by Dereniowski *et al.* [23], where robots are differentiated by the time they require to adopt particular routes in the network. They show that in networks with nodes equipped with unique labels breaking symmetry is no longer the main source of problems, and homogenity of robots supports more efficient rendezvous. In this paper we refer to the extreme case of synchronous rendezvous considered in [23] by assuming that the cost imposed on each edge is either unit or infinite, and in turn disallow robots to visit certain parts of the network. To the best of our knowledge, the conference version [29] of this paper is the only past work on the same models and variants of rendezvous. Having said this, we would like to note that apart from close ties with the classical rendezvous problem our work has also applications in communication in cognitive radio networks in which *blind rendezvous* (one of the rendezvous variants studied in this paper) has been considered recently in complete graphs [12, 35].

1.1 Related work

The rendezvous problem was studied in different models and under a number of diverse assumptions. A vast literature includes several exhaustive surveys on rendezvous and other searching problems, see, e.g., [5, 6, 38]. The past work on rendezvous includes both deterministic algorithms surveyed recently in [38] as well as randomized approaches including extensively cited work in [3, 4, 9, 10]. Another group of algorithms focuses on geometric settings including past work on the line [10, 11] and the plane [7, 8] as well as more recent on fat (with non-zero radius) robots [1, 15]. Rendezvous algorithms designed for infinite (Euclidean) spaces for both synchronized and asynchronous solutions are considered in [13, 14, 17].

A very close relative of rendezvous is the *gathering problem*, in which many (more than 2) robots are expected to meet. For example, gathering in graphs with different topologies has been extensively studied in [18, 19, 25, 26], gathering of possibly faulty robots has been studied by Agmon and Peleg [2], and gathering protocols tolerating single faulty robot can be found in [24]. More recently, Das *et al.* [21] considered the gathering problem with a powerful malicious robot and weaker honest robots. Finally, gathering of robots with limited visibility has been studied by Degener *et al.* [22] and Katreniak [32].

There has been also research effort in better understanding of models with robots having extra knowledge (at the start, or sensed/communicated in due course), often referred to as *advice*. Izumi et al. [31] consider robots with unreliable compasses, Das et al. [20] analyse the case with robots equipped with a device which measures its distance to the other robot on the conclusion of each step, while in [28] the authors allow robots to sniff others up to a certain distance. In [27] to accomplish rendezvous robots communicate by beeps.

As indicated earlier, a large volume of rendezvous algorithms have been considered for graph based environments, see, e.g., [16, 33, 37] and more recent work on dynamic, evolving graphs [40]. However, in contrast to this paper all previous work on rendezvous is devoted to the case with both robots having access to the same parts of the network.

1.2 Model of Computation and Rendezvous Variants

We consider rendezvous of *anonymous* (also indistinguishable with respect to the control mechanism) robots in networks modelled by undirected graphs. The network $G = (V, E)$ is a simple connected graph, where $|V| = n$ and $|E| = m$, in which nodes $s_A, s_B \in V$ are the starting points of robots A and B respectively. Moreover, for each robot $X \in \{A, B\}$ we define its *reachability graph*, also known as *the map* $G_X = (V_X, E_X)$, a subgraph of G in which V_X and E_X are respectively the sets of nodes and edges reachable from the starting point s_X. We also assume that each robot X has *a priori* access **only** to map G_X, i.e., part of the network accessible by X. This assumption is different to the past work on rendezvous where robots operate on the same network which topology may be unknown. We define $k_X = |V_X|$ as *the size* of map G_X and assume w.l.o.g. that $k_A \le k_B$. While the two robots are anonymous, we use some extra assumptions with respect to the network nodes and in some models the edges too. In particular, we assume that nodes of network $G = (V = \{v_1, v_2, \ldots, v_n\}, E)$ are ordered implicitly, i.e. we assume that $v_i < v_{i+1}$, for all $i = 1, 2, \ldots n-1$. The ordering is implicit in a sense that (with exception of Section 2.3) nodes have no labels. In particular, if $V_X = \{v_1^{(X)}, v_2^{(X)}, \ldots, v_{k_X}^{(X)}\}$, $v_a^{(X)} = v_i$, and $v_b^{(X)} = v_j$, where $v_i, v_j \in V$ and $i < j$, then robot X knows that $v_a^{(X)} < v_b^{(X)}$. Please note that while the orders in G_A and G_B are consistent, robot A is not aware of whether G_A contains more nodes than G_B. Also the mapping between the nodes in G_A and G_B is unknown. Finally, let $T(V_X)$ be a rooted tree in G_X that spans all nodes in V_X in which the starting point s_X is placed in the root of $T(V_X)$ and the order of children is consistent with the order of nodes in V_X.

We consider both synchronous and asynchronous models of computation. In the **synchronous model** robots A and B have access to the global clock ticking in discrete *time steps* $0, 1, 2, \ldots$. The protocol in each robot starts with the global clock set to time 0. During a single time step each robot assesses the node it resides in (this includes detection of the other robot), and decides whether to stay at the same node or to move to one of its neighbours via an available edge. During traversal along the edge the "eyes" (all detection mechanism) of the robot are switched off. Consequently, since the robots cannot meet on edges rendezvous has to take place at some node. The running time of all algorithms is bounded, i.e., the robots stop eventually.

In the **asynchronous model** each robot computes its trajectory, and in particular the sequence of visited nodes and edges, without access to the global clock. Instead, when the robot is ready to move along a chosen edge it awaits the relevant "go" signal from the adversary. In this model we assume that the use of edges is exclusive, i.e., two robots cannot be located on the same edge at any time. In consequence, rendezvous is possible only on nodes which is consistent with the synchronous model. Also instead of the running time, as the complexity measure we adopt here the longer of the two robots' trajectories.

Table 1: Summary of results

<table>
<tr><th>model
variant</th><th>synchronous
(time complexity)</th><th>asynchronous
(length of trajectory)</th></tr>
<tr><td>EM</td><td>$\Theta(k_A + k_B)$ (Thm 1)</td><td>$O((k_A + k_B)^2)$ (Thm 7)</td></tr>
<tr><td>EI
$E_A \subseteq E_B$</td><td rowspan="2">$O((k_A + k_B)\log(k_A + k_B))$ (Thm 2)</td><td>feasible (Thm 9)</td></tr>
<tr><td>NI</td><td rowspan="3">not feasible (Thm 8)</td></tr>
<tr><td>BR+
(labels)</td><td>$\min\{O((k_A + k_B)^3 \log\log n,$
$O((k_A + k_B)^2 \log n)\}$ (Thm 5)</td></tr>
<tr><td>BR</td><td>not feasible (Thm 4)</td></tr>
</table>

In what follows we study three **variants** of rendezvous characterised by different levels of restrictions imposed on maps provided to robots A and B.

1. Edge Monotonic (EM) Variant In this variant each robot $X \in \{A, B\}$ has the associated weight w_X and each edge e_i in E has a weight restriction $w(e_i)$. These weight restrictions are consistent with the indices of edges $e_1, e_2, \ldots e_m$ in E. I.e., edge e_j tolerates weights non-smaller than $w(e_i)$, for any $1 \le i < j \le m$. Let i_X be the smallest index, s.t., e_{i_X} tolerates weight w_X. We assume that robot X is only allowed to traverse edges with index $\ge i_X$.

2. Node Inclusive (NI) Variant Here we only assume $V_A \subseteq V_B$, allowing otherwise an arbitrary relationship between the sets of edges E_A and E_B.

3. Blind Rendezvous (BR) Variant Here we only assume $V_A \cap V_B \neq \emptyset$ and the relationship between E_A and E_B is arbitrary.

1.3 Our results

In this paper we study the three variants of rendezvous in both synchronous and asynchronous models. In Section 2 we study synchronized rendezvous. In particular, in Subsection 2.1 devoted to EM variant we present an optimal $O(k_A + k_B)-$time rendezvous algorithm **RV1**; in Subsection 2.2 we present rendezvous Algorithm **RV2** that meets two robots in almost linear time $O((k_A+k_B)\log(k_A+k_B))$ for NI variant; and finally in Section 2.3 we show that BR variant of rendezvous has no solution. In order to overcome this deficiency we introduce a new variant BR+ enriched with explicit labels and present two algorithms **RV3** and **RV4** which superposition allows robots to rendezvous in time $\min\{O((k_A + k_B)^3 \log\log n, O((k_A + k_B)^2 \log n)\}$.

In Section 3 we focus on the asynchronous model. In Subsection 3.1 we present Algorithm **RV5** for EM variant of length $O((k_A + k_B)^2)$. Section 3.2 focuses on rendezvous feasibility studies for a variant EI, which is a subclass of the variant NI defined by imposing a further restriction that one of the edge sets is included in the other. We show that without this assumption the rendezvous is not possible, what implies also infeasibility of asynchronous Blind Rendezvous. The summary of the results is given in Table 1. Please note that the relevant results hold when the two robots start in the same connected component of G. Otherwise, since the rendezvous algorithms are upper bounded (depending on the model) either in terms of the time complexity or the length of the adopted trajectory, the robots eventually conclude that the rendezvous is not feasible.

2 Synchronous Rendezvous

In this section we focus on the synchronous model. We propose and analyse several efficient algorithms for different variants of rendezvous.

Algorithm RV1($X \in \{A, B\}$) **Step 1** Walk from s_X to the target node v_X^* along the shortest path in $SL(X)$; **Step 2** Wait in v_X^* until conclusion of time step $2k_X$; **Step 3** Walk along the Euler tour of $T(V_X)$ and *Halt*.

Figure 1: Pseudo-code of algorithm for EM variant of rendezvous

2.1 Edge Monotonic (EM) Variant

Recall that in this variant edges in $E = \{e_1, e_2, ..., e_m\}$ are ordered according to their weight restrictions $w(e_1), w(e_2), \ldots, w(e_m)$, i.e., $w(e_i) \le w(e_j)$ for all $1 \le i < j \le m$. When the total order on edges is needed the ties between edges with the same weight restriction can be broken with the help of the implicit order imposed on the nodes. We define a sequence of subgraphs $G(l) = (V(l), E(l))$, where for any $l \in \{1, \ldots, m\}$, $E(l) = \{e_l, e_{l+1}, ..., e_m\}$ and $V(l)$ is the set of nodes in V induced by the edges of $E(l)$, and where $E(l + 1) \subset E(l)$. In this variant each robot X is associated with the threshold index $i_X \in \{1, \ldots, m\}$ determining set of edges $E(i_X)$ traversable by X. In other words, robot X can walk only along edges from $E(i_X)$. For each $l \in \{i_X, \ldots, m\}$, we also define $V_X(l)$ as the set of nodes reachable from s_X via edges in $E(l)$ as well as $G_X(l) = (V_X(l), E_X(l))$ as the graph induced in G_X by set $V_X(l)$. The following lemma holds.

Lemma 1. *In EM Variant either* $(V_A \subseteq V_B)$ *or* $(V_B \subseteq V_A)$*, or* $V_A \cap V_B = \emptyset$*.*

Proof. The statement of the lemma is false if all of terms $(V_A \subseteq V_B), (V_B \subseteq V_A)$, and $V_A \cap V_B = \emptyset$ are false too. Assume w.l.o.g. that $V_A \cap V_B \ne \emptyset$, where $V_A = V_A(i_A)$ and $V_B = V_B(i_B)$, and $i_A \ge i_B$. Since $i_B \le i_A$ (edges traversable by A are also traversable by B) and $V_A \cap V_B \ne \emptyset$ (the reachability graphs G_A and G_B coincide) all edges and points in $G_A(i_A)$ are also available to B, meaning $V_A \subseteq V_B$. □

For each robot X we define a *sleeve of graphs* which is denoted by $SL(X)$.

Definition 1. *The sleeve of graphs $SL(X)$ with respect to X is the maximal sequence of decreasing connected components $G_X(i_X)$, $G_X(i_X + 1), \ldots, G_X(l^*)$, which satisfy $|V_X(l + 1)| > |V_X(l)|/2$, for all $i_X \le l < l^* \le m$. A subsequence $G_X(i_X + j)$, $G_X(i_X + j + 1), \ldots, G_X(l^*)$, for any $j \in \{0, 1, \ldots, l^* - i_X\}$, is called a tail of $SL(X)$ and the smallest (in the adopted order) node $v_X^* \in V_X(l^*)$ is called the target in $SL(X)$.*

The rendezvous task is accomplished when robots A and B meet at some node eventually. Figure 1 contains a pseudo-code of Algorithm **RV1** which is designed for EM variant of rendezvous.

Theorem 1. *If rendezvous is feasible, i.e., $s_A \in G_B$, Algorithm* **RV1** *admits meeting in asymptotically optimal time $O(k_A + k_B)$.*

Proof. Recall $k_A \le k_B$. According to Lemma 1 if rendezvous is feasible, i.e., $V_A \cap V_B \ne \emptyset$ we conclude that $V_A \subseteq V_B$. We consider two complementary cases:

Case 1 $[2k_A > k_B]$ Since $2k_A > k_B$, according to Definition 1 sleeve $SL(A)$ is a tail of $SL(B)$ and the two sleeves share the same target v^*. The robots A and B are initially placed in their own sleeves at distance at most $k_B < 2k_A$ from the joint target v^*. This admits rendezvous in **Step 1** in time bounded by k_B.

Case 2 $[2k_A \le k_B]$ In this case robot A halts at the latest at time step $4k_A$ on the conclusion of **Step 3**, i.e., after $2k_A$ time steps devoted to **Step 1** and **Step 2**, followed by additional $2k_A - 2$ time steps devoted to the Euler tour traversal in $T(V_A)$) in **Step 3**. Note, however, that robot B enters **Step 3** in time step $2k_B + 1 > 4k_A$, when robot A is already immobilized. Since during **Step 3** robot B visits all nodes in V_B (that include also all nodes in V_A) rendezvous must occur.

□

1. **Algorithm RV2**($X \in \{A, B\}$)
2. **Step 1** Compute j_X and $b_X[0, \ldots, j_X]$.
3. **Step 2** **for** $j = 1, 2, \ldots, j_X$ **do**
4. **if** $(j = j_X)$ {*active stage*}
5. **use** 2^j **time steps** to **walk** to and **wait** in v_X^*.
6. (i) **for** $i = 0, 1, \ldots, j$ **do**
7. **if** $(b_X[i] = 1)$
8. (a) **use** $2 \cdot 2^j$ time steps for Euler tour in $T(V_X)$
9. and **return** to v_X^*
10. **else** (b) **wait** $2 \cdot 2^j$ time steps in v_X^*
11. **else** (ii) **wait** $2^j \cdot (2j + 3)$ time steps where you are.

Figure 2: Pseudo-code of algorithm for NI variant of rendezvous

2.2 Node Inclusion (NI) Variant

In this variant we assume that all nodes are ordered and $k_A \le k_B$, where $V_A \subseteq V_B$. However, as we have no order on edges in this variant the concept of sleeve of graphs cannot be utilised. Instead, one has to propose an alternative mechanism that will allow to distinguish between the two robots, and with this goal in mind we focus on the values k_A and k_B. Note that if $k_A = k_B$ due to the inclusion assumption we also have $V_A = V_B$. In this case, since orders of nodes in V_A and V_B are consistent the robots can meet at the smallest (in order) node v^* in V_A and V_B, which must coincide. Otherwise, the values of k_A and k_B differ and each robot X can adopt k_X as its unique identifier. Furthermore, apart from adopting unique identities also some synchronization mechanism is needed (sizes of k_A and k_B can be dramatically different) which will allow robots to coordinate their individual moves. The rendezvous mechanism used by each robot X is based on synchronized stages which increase in length. Robot X becomes active in the first stage which is long enough to accommodate actions reflecting the size k_X. This is stage j_X, where $2^{j_X - 1} \le k_X < 2^{j_X}$. Thus the rendezvous algorithm utilised by robot X operates in stages $j = 1, 2, 3, ..., j_X$, where during stages 1 through $j_X - 1$ the robot remains dormant and in its only active stage j_X the robot visits all nodes in V_X with the aim to accomplish the rendezvous process. Note that if $j_A < j_B$ (and $V_A \subset V_B$), in stage j_B, when A is already immobilized, by visiting all nodes in V_B (a superset of V_A) robot B must conclude the rendezvous process. In the complementary case, i.e., when $j_A = j_B$, the binary expansions $b_A[0, \ldots, j_A]$ and $b_B[0, \ldots, j_B]$ of k_A and k_B respectively are utilised to differentiate between the robots.

Lemma 2. *If $j_A = j_B$ and $k_A < k_B$ there exists $i \in \{0, 1, \ldots, j_A = j_B\}$, s.t., $b_A[i] = 0$ and $b_B[i] = 1$.*

Proof. If for $i = 0, 1, \ldots, j_A = j_B$, $(b_A[i] = 0) \implies (b_B[i] = 0)$, we result in contradiction $k_A \ge k_B$. □

The pseudo-code of Algorithm **RV2** for the NI variant of rendezvous is given in Figure 2. If at any time step the two robots A and B meet, rendezvous is accomplished and the two robots *halt*. We prove the following theorem.

Theorem 2. *If rendezvous is feasible, i.e., $s_A \in G_B$, Algorithm* **RV2** *admits meeting in time $O((k_A + k_B)\log(k_A + k_B))$.*

Proof. The rendezvous algorithm runs in j_X stages controlled by the loop **for** in line 3. We distinguish two cases. In the first case, in which we assume $j_A < j_B$, when robot B is in the active stage robot A is already immobilized, and B meets A during traversal of the Euler tour in $T(V_B)$, see line 8 of the code. Otherwise, when $j_A = j_B$ we have two subcases. In the first subcase when $k_A = k_B$ the robots meet in the shared smallest node v^*, see line 5. In the second subcase, where $k_A < k_B$, according to Lemma 2 there exists i, s.t., $b_A[i] = 0$ and $b_B[i] = 1$ when robot B traverses the Euler tour in $T(V_B)$ and robot A is immobilized. Thus this subcase admits rendezvous too.

With respect to the time complexity we first observe that the execution time of algorithm **RV2** is bounded and it depends strictly on parameter j_X. The time complexity of each stage $j = 1, ..., j_X$ is bounded by $(2^j \cdot (2j + 3))$, as indicated in line 11 in the pseudocode, resulting in the total complexity $\sum_{j=1}^{j_X}(2^j \cdot (2j+3)) \le \sum_{j=1}^{j_X}(2^j \cdot (2j_X+3)) =$

1. **Algorithm RV2b**($X \in \{A, B\}$)
2. **Step 1 for** $j = 1, 2, \ldots, k_X$ **do**
3. **walk** to j-th node
3. **wait** until time $d \cdot j$
4. **Step 2** *Halt*

Figure 3: Pseudo-code of algorithm for NI variant of rendezvous with bounded diameter

$(2j_X + 3)\sum_{j=1}^{j_X}(2^j) = (2j_X + 3)\cdot(2^{j_X+1} - 2) = O(k_X \cdot \log k_X)$, since $2^{j_X} - 1 \le k_X < 2^{j_X}$, and admits the time complexity $O((k_A + k_B)\log(k_A + k_B))$. ☐

2.2.1 Bounded diameter networks

Here we comment on the case when both robots know the common constant bound d on diameters of G_A and G_B, i.e., $\max\{\mathrm{diam}(G_A), \mathrm{diam}(G_B)\} \le d = O(1)$. In such case there exists a simple asymptotically optimal linear time rendezvous solution, see Algorithm **RV2b** described in Figure 3.

Theorem 3. *If rendezvous is feasible, i.e., $s_A \in G_B$, and both robots are aware of the common bound $d = O(1)$ on diameters in G_A and G_B, Algorithm* **RV2b** *admits rendezvous in asymptotically optimal time $O(k_A + k_B)$.*

Proof. Recall that $V_A \subseteq V_B$ and the orders of the nodes in both sets are consistent. Let u be the last node (in order) visited by robot A. Note that u belongs also to V_B. We observe that when robot B visits u the other agent A is already immobilized in u as agent B must visit at least the same number of nodes as A before arriving in u. Thus, rendezvous of A and B will take place at the latest in u in time bounded by $k_B \cdot d = O((k_A + k_B))$. ☐

2.3 Blind Rendezvous (BR) Variant

In this section we consider rendezvous where the relationship between the maps of robots is arbitrary. We first show that without any additional information, even if $V_A \cap V_B \neq \emptyset$, rendezvous cannot be reached.

Theorem 4. *In an arbitrary graphs BR variant of rendezvous is not feasible.*

Proof. Assume that for each robot $X \in \{A, B\}$ we have $V_X = \{v_1^{(X)}, v_2^{(X)}\}$ as well as $E_X = \{(v_1^{(X)}, v_2^{(X)})\}$, where node $v_2^{(A)}$ coincides with $v_1^{(B)}$ in G and for each X the starting node s_X coincides with v_1^X in G_X. It is enough to observe that without any additional information the symmetry tie cannot be broken. And indeed, since the robots are anonymous (indistinguishable) whenever robot A visits $v_2^{(A)}$ at the same time robot B visits $v_2^{(B)}$, and in turn the two robots never visit the shared node in G simultaneously. ☐

In order to overcome this deficiency, one can assume that the nodes in V_X (apart from being ordered) have also *explicit labels* in G and across G_X, for each $X \in \{A, B\}$. We refer to this enhanced variant as **BR+** variant of rendezvous. In consequence, if a node $v_a^{(A)} \in V_A$ coincides with $v_b^{(B)} \in V_B$ they can both operate on the same label. We assume that the labels are drawn from the set of integers $\{1, 2, \ldots, n\}$, and we use notation $b_i^{(X)}$ (or $b_i^{(X)}[\log n, ..., 1, 0]$) to denote the binary expansion of the explicit label of $v_i^{(X)} \in V_X$.

We also assume n is known to both robots. Otherwise no rendezvous algorithm would stop and report infeasibility of rendezvous when $V_A \cap V_B = \emptyset$, as robots are not aware of the sizes of maps of one another.

Before we present two rendezvous algorithms we show that the symmetry tie problem, see Theorem 4, can be overcome if the explicit labels are available. W.l.o.g. we assume that the order of labels is consistent with the order imposed on nodes on each map. If this is not the case a new (consistent) order for nodes in V_A and V_B can be computed on the basis of explicit labels (we only care about nodes in $V_A \cap V_B$). The following result has been shown in [12]. Our proof is much simpler and based on binary representation of explicit labels.

Lemma 3. **[12]** *Assume that the map of each robot $X \in \{A, B\}$ is an ordered pair of nodes $(v_1^{(X)}, v_2^{(X)})$ connected by a symmetric edge, where nodes $v_2^{(A)}$ and $v_1^{(B)}$ coincide in G and nodes $v_1^{(A)}$ and $v_2^{(B)}$ don't. In such network one can break the symmetry tie to reach rendezvous in time $O(\log\log n)$.*

Proof. (Simplified) We first observe that according to the imposed order $b_1^{(A)} < b_2^{(A)} = b_1^{(B)} < b_2^{(B)}$. The case in which $s_A = v_2^{(A)}$ and $s_B = v_1^{(B)}$ is trivial, and another case where $s_A = v_1^{(A)}$ and $s_B = v_2^{(B)}$ can be easily resolved by an algorithm that alternates between the two nodes (e.g., in every other time step). Let $1 \le r_A \le \log n$ be the largest integer position, s.t., $b_1^{(A)}[r_A] \ne b_2^{(A)}[r_A]$. Since $b_1^{(A)} < b_2^{(A)}$ one can conclude that $b_1^{(A)}[r_A] = 0$ and $b_2^{(A)}[r_A] = 1$. Similarly let $1 \le r_B \le \log n$ be the most significant bit, s.t., $b_1^{(B)}[r_B] \ne b_2^{(B)}[r_B]$. Since $b_1^{(B)} < b_2^{(B)}$ one can also conclude that $b_1^{(B)}[r_B] = 0$ and $b_2^{(B)}[r_B] = 1$. We observe that since $b_2^{(A)} = b_1^{(B)}$ one can conclude that $r_A \ne r_B$ as the respective positions cannot contain 0 and 1 at the same time. Moreover binary expansions br_A and br_B of r_A and r_B respectively are limited to $\log\log n + 1$ bits.

We consider a symmetry breaking algorithm in which in $2\log\log n + 2$ time steps $i = 1, 2, \ldots, 2\log\log n + 2$ each robot $X \in \{A, B\}$ moves to the other node only if $i = 2 \cdot l$ (i is even) or if $i = 2 \cdot l - 1$ (i is odd) and $br_X[l] = 1$, for $l = 1, \ldots, \log\log n + 1$. Note that since $r_A \ne r_B$ for some $1 \le l \le \log\log n + 1$ we have $br_A[l] \ne br_B[l]$ and if until now the rendezvous is not reached (all previous moves were symmetric and in the last odd time step, when the symmetry was broken robots occupy different nodes) in the next even step the rendezvous process is concluded. □

Corollary 1. *Note that Lemma 3 applies to two pairs of nodes located at distance 1. If for $X \in \{A, B\}$, G_X includes an ordered pair of nodes $(v_1^{(X)}, v_2^{(X)})$ with distance bounded by d between them, robot X is at one of these two nodes, $v_2^{(A)}$ and $v_1^{(B)}$ refer to the same node, and $v_1^{(A)}$ and $v_2^{(B)}$ are different nodes, then the symmetry breaking rendezvous takes time $O(d\log\log n)$.*

In the remaining part of this section we present two rendezvous algorithms followed by their superposition. The first algorithm **RV3** has the time complexity $O((k_A + k_B)^3 \log\log n)$ and its idea is based on the blind rendezvous algorithm from [12] where the problem was studied in complete graphs. The second algorithm **RV4** has the time complexity $O((k_A + k_B)^2 \log n)$ making it superior to **RV3** when $k_A + k_B > \omega(\tau)$, where $\tau = \frac{\log n}{\log\log n}$ is the threshold value. This rendezvous algorithm resembles algorithm **RV2** however here the symmetry tie is broken with the help of explicit labels.

2.3.1 Blind rendezvous in time $O((k_A + k_B)^3 \log\log n)$

Similarly to its predecessor **RV2** also the first blind rendezvous algorithms **RV3**, see Figure 4, operates in stages accommodating geometrically increasing estimates on sizes of the maps. This is needed as neither of the robots knows the size of the map of the other robot. Each robot starts using active stages only when the current estimate is large enough to accommodate its map. The rendezvous process terminates in time $O((k_A + k_B)^3 \log\log n)$ if the maps of both robots are smaller than the threshold value τ. Otherwise, algorithm **RV3** is followed by execution of algorithm **RV4**. If at any time step the two robots A and B meet, the rendezvous is accomplished and the two robots *halt*.

Theorem 5. *If $k_A + k_B < \tau = \frac{\log n}{\log\log n}$ and $V_A \cap V_B \ne \emptyset$ (rendezvous is feasible) Algorithm* **RV3** *admits meeting in BR+ variant in time $O((k_A + k_B)^3 \log\log n)$.*

Proof. The rendezvous algorithm runs in $\lceil \log \tau \rceil$ stages controlled by the loop **for** in line 3. Robot X starts executing active stages as soon as the stages can accommodate the size of X's map. If the size of the map is too big, robot X awaits execution of the second rendezvous algorithms **RV4**, see lines 11–12. During an active round all pairs (a, b) from the Cartesian product $\{1, \ldots, 2^j\} \times \{1, \ldots, 2^j\}$ are drawn in the lexicographic order. Only certain pairs are valid, i.e., when both $v_a^{(X)}$ and $v_b^{(X)}$ exists. In each valid pair if only one node exists robot X remains in this node for the duration of the symmetry breaking procedure. Otherwise, if both nodes exist the breaking symmetry procedure is executed with the distance between the two nodes bounded by 2^j.

If rendezvous is feasible we must have nodes $v_a^{(A)} \in V_A$ and $v_b^{(B)} \in V_B$ that coincide by sharing the same label. If the pair $(v_a^{(X)}, v_b^{(X)})$ exists in both maps thanks to the symmetry breaking procedure eventually robot A will visit $v_a^{(A)}$

Algorithm RV3($X \in \{A, B\}$)
Step 1 Compute j_X and the threshold $\tau = \frac{\log n}{\log \log n}$.
Step 2 **for** $j = 1, 2, \ldots, \lceil \log \tau \rceil$ **do**
    if $(j \geq j_X)$ {*active stage*}
        (i) **for** all pairs $(a, b) \in \{1, \ldots, 2^j\} \times \{1, \ldots, 2^j\}$
            ordered lexicographically **do**
            if (both of $v_a^{(X)}, v_b^{(X)}$ exist)
                (a) **run** symmetry breaking algorithm
                    from Lemma 3 in pair $(v_a^{(X)}, v_b^{(X)})$
                    in $2^j \lceil \log j \rceil + 2$ time steps
            else (b) **wait** the relevant $2^j \lceil \log j \rceil + 2$ time steps
                    in the current location;
    else (ii) **wait** the relevant $2^{2j} \cdot (2^j \lceil \log j \rceil + 2)$ time steps.

Figure 4: Pseudo-code of algorithm for BR+ variant of rendezvous

at the same time when entity B visits $v_b^{(B)}$ and the rendezvous is reached. If only one element of the pair $(v_a^{(X)}, v_b^{(X)})$ exists, i.e., either $v_a^{(A)}$ for A or $v_b^{(B)}$ for B the respective robot is asked to wait in the existing node of the pair resulting in rendezvous too. Otherwise the robots await the next pair from the Cartesian product without movement for the period corresponding to execution of the symmetry breaking procedure. Thus the actions performed by robots A and B remain fully synchronized.

The time complexity of stage $j = 1, ..., \lceil \log \tau \rceil$ is bounded by $O(2^{3j} \cdot \log \log n)$. Note that if rendezvous takes place for some $j' \leq \lceil \log \tau \rceil$, where $j' \leq j_B$, the total time complexity is bounded by $\sum_{j=1}^{j'} O(2^{3j} \cdot \log \log n) = O(2^{3 \cdot j'} \cdot \log \log n)$. And in turn since $2^{3 \cdot j'} \leq 2^{3 \cdot j_B} \leq 2 \cdot k_B^3 = O((k_A + k_B)^3)$, the total time complexity is O($(k_A + k_B)^3 \cdot \log \log n$).

□

2.3.2 Blind rendezvous in time $O((k_A + k_B)^2 \log n)$

We start with the proof of the following lemma.

Lemma 4. *One can impose a cyclic order $\pi(X)$ on nodes of a spanning tree $T(V_X)$, s.t., the walking distance (the number of edges to be visited) between two consecutive nodes in order $\pi(X)$ is at most* 3.

Proof. We say that the nodes located at an even distance from the root s_X are on an even level and all the remaining nodes are on an odd level. The ordering of nodes π is created according to the following principle. Starting from the root s_X we visit all nodes in $T(V_X)$ using depth-first search algorithm. The root gets label 0. When we arrive (from the parent) to an even level the currently visited node gets the next available label. In other words at even levels we use *pre-order numbering principle*. And when we arrive (from the last child) to an odd level the currently visited node gets the next available label. I.e., at odd levels we follow *post-order numbering principle*

We now show that the labeling (ordering) procedure given above generates at least one new label in three consecutive steps. And indeed, if we follow the route determined by the depth-first search algorithm and we visit for the first time a node v at an even level (when the new label is generated): (case 1) if the first child of v has a child w then w (which is at distance 2 from v) gets the new label; (case 2) if the first child of v is a leaf this child (which is at distance 1 from v) gets the new label; (case 3) if the node v is a leaf but not the last child of its parent the next label goes to the (next) sibling of v (which is at distance 2); and (case 4) if v is the last child the next label goes to its parent (which is at distance 1).

Similarly, if v is visited for the last time on an odd level it gets a new label. Now (case 5) if v is the last child and its parent w is not the last child the next sibling of the parent (which is at distance 3 from v) gets the new label; (case 6) if v is the last child and its parent w is also the list child then the parent of w (at distance 2 from v) gets the new

1. **Algorithm RV4**($X \in \{A, B\}$)
2. **Step 1** Determine j_X, the threshold $\tau = \frac{\log n}{\log \log n}$, and label $b_i^{(X)}$ of s_X;
3. **Step 2 for** $j = \lceil \log \tau \rceil, \ldots, \lceil \log n \rceil$ **do**
4.   **if** ($j \geq j_X$) {*active stage*}
5.   (**walk** to and **wait** in s_X) in 2^j time steps;
6.   **for** $l = 0, 1, \ldots, \lceil \log n \rceil$ **do** {test all bits}
7.   **if** ($b_i^{(X)}[l] = 1$) {walk all the time}
8.   **for** $2^{2j} \times 3$ time steps **do**
9.   **walk** to the next node in order $\pi(X)$;
10.   **else repeat** 2^j times {walk and wait for another}
11.   (**walk** to the next node in order $\pi(X)$
12.   and **wait** there) in $2^j \times 3$ time steps;
13.   **else wait** the relevant $3 \cdot 2^{2j} \lceil \log n \rceil$ time steps in place.

Figure 5: Pseudo-code of algorithm for BR+ variant of rendezvous

label; (case 7) and if v is the last child and its parent is the root, the cyclic order is established (and the next label is at distance 1). In the remaining cases when v is not the last child (case 8) if its next sibling (at distance 2) is a leaf it gets the new label; and (case 9) if the next sibling of v has children the next label go to the first child (at distance 3 from v) of this sibling. □

The last rendezvous algorithm **RV4** presented in this section operates on the following principle. At the start of each active stage robot X returns (if moved before) to the starting point s_X. If the two starting points in V_A and in V_B coincide rendezvous is accomplished. Otherwise the algorithm controls further movement of robots, s.t., during long enough ($\geq 2^j \times 3$ time steps) interval of an active stage j one of the robots, say w.l.o.g. A, visits all nodes in V_A in the cyclic order $\pi(A)$ with frequency of one visit per three time steps. While the other robot B visits consecutive nodes with frequency of $2^j \times 3$ time steps. So when eventually robot B resides in the node that belongs to $V_A \cap V_B$ there is enough time for robot A to arrive in this node before B moves away. If at any time step the two robots A and B meet, the rendezvous is accomplished and the two robots *halt*.

Theorem 6. *If $k_A + k_B \geq \tau = \frac{\log n}{\log \log n}$ and $V_A \cap V_B \neq \emptyset$ (rendezvous is feasible), Algorithm* **RV4** *admits meeting in BR+ variant in time $O((k_A + k_B)^2 \log n)$.*

Proof. Lets consider the first stage active for robots A and B, i.e., when $j = j_B$. Note that line 13 of the pseudo-code, see Figure 5, accommodates for the waiting time needed for two robots to stay synchronized prior to this stage. In this active stage loop for in line 6 compares consecutive bits of labels $b_i^{(A)}$ adopted by A and $b_{i'}^{(B)}$ adopted by B. There must be at least one position l on which the two labels differ. In consequence, there is a period of $2^{2j} \times 3$ time steps during which one of the robots, say w.l.o.g. A, with the bit $b_i^{(A)}[l] = 1$, visits periodically all nodes in V_A with frequency of 3 time steps per node. During the same period the other robot B with the bit $b_{i'}^{(B)}[l] = 0$ waits long ($\geq 2^j \times 3$ time steps) periods of time in every node of V_B. So when eventually robot B visits the node that belongs to $V_A \cap V_B$ the other robot A has enough time to arrive in this node before B moves on.

The time complexity of this first active stage is $O(2^{2j_B} \cdot \log n) = O(k_B^2 \log n)$. Since the duration of stages grows exponentially we conclude that the total time complexity is also $O(k_B^2 \log n) = O((k_A + k_B)^2 \log n)$. □

Corollary 2. *In the enhanced BR+ variant of rendezvous two robots can meet in time $O(\min\{(k_A + k_B)^3 \log \log n, (k_A + k_B)^2 \log n\})$.*

Proof. The result follows directly from the superposition of **RV3** and **RV4**. □

3 Asynchronous Rendezvous

In this section we focus on the asynchronous model in which movement of robots is determined by predefined trajectories along which progression of robots is governed freely by the adversary. We show some infeasibility results and we design and analyse several algorithms for the considered variants of rendezvous.

3.1 Edge Monotonic (EM) Variant

In this variant nodes in $V = \{v_1, ..., v_n\}$ and edges in $E = \{e_1, ..., e_m\}$ are ordered, where the order on edges is consistent with their weight restrictions, i.e., $w(e_1) \leq w(e_2) \leq \ldots w(e_m)$, see Section 2.1. For all $l = 1, 2, \ldots m$, let $G(l)$ be the subgraph of G induced by edges $e_l, ..., e_m$ and $\bar{T}(l)$ be the spanning forest in $G(l)$ computed by the Kruskal's minimum spanning tree algorithm, with the weight on each edge e_i set to $-w(e_i)$.

Lemma 5. *For any $i < j$ the forest $\bar{T}(j)$ is a subforest of $\bar{T}(i)$ in $G(i)$.*

Proof. As $E(j) \subset E(i)$, the thesis follows directly from Kruskal's algorithm. □

The main idea behind our solution is the following recursively constructed universal *walk* $W(\bar{T}(1))$ which visits all nodes in V, and satisfies two conditions:

(C1) For each $l \in \{1, ..., m\}$, the walk $W(e_l)$ associated with edge e_l is a tour visiting this edge exactly once in each direction, where the starting point $W(e_l)^S$ and the finishing point $W(e_l)_F$ of walk $W(e_l)$ coincide with the endpoint of e_l with the smaller index in V.

(C2) For any level $l \in \{1, ..., m-1\}$, assume forest $\bar{T}(l)$ consists of $k(l)$ trees $T_1, ..., T_{k(l)}$ where only one T_i contains edge e_l.

- If T_i is a single edge $W(e_l)$ becomes $W(T_i)$, where $W(T_i)^S = W(e_l)^S$ and $W(T_i)_F = W(e_l)_F$.
- If T_i is formed of some tree $T'_i \in \bar{T}(l+1)$ extended by e_l, in order to create $W(T_i)$ we adopt $W(T_i)^S = W(T'_i)^S, W(T_i)_F = W(e_l)_F$, and we connect $W(T'_i)_F$ with $W(e_l)^S$ by the relevant simple path in tree T_i.
- If T_i is formed of two trees $T'_i, T''_i \in \bar{T}(l+1)$ connected by e_l, then to create $W(T_i)$ we take $W(T_i)^S = W(T'_i)^S, W(T_i)_F = W(T''_i)_F$, and we connect $W(T'_i)_F$ with $W(e_l)^S$ and $W(e_l)_F$ with $W(T''_i)^S$ by the relevant simple paths in T_i.

1. **Algorithm RV5(**$X \in \{A, B\}$**)**
2. **Step 1** Compute $W(T_X)$;
3. **Step 2** Walk full length of $W(T_X)$;
4. **Step 3** *Halt.*

Figure 6: Pseudo-code of asynchronous algorithm for EM variant of rendezvous

We need the following lemmas.

Lemma 6. *For any level $l \in \{1, ..., m\}$ and tree $T_i \in \bar{T}(l)$, $|W(T_i)| = O(|T_i|^2)$.*

Proof. The proof is done by induction on (decreasing) level l. If T_i is a single edge the proof is immediate. If T_i is formed at level l from tree T'_i, and edge e_l we know that $|T_i| = |T'_i| + 1$ and we can assume $|W(T'_i)| = O((T'_i)^2)$. As the length of a simple path connecting $W(T'_i)_F$ with $W(e_l)^S$ is not longer than $|T'_i| + 1$ we also get $|W(T_i)| = O((T'_i)^2) + |T'_i| + 2 = O(|T_i|^2)$. In the remaining case, we need two simple paths not longer than *max*$(|W(T'_i)| + 1, |W(T''_i)| + 1)$, thus also in this case we can conclude that $|W(T_i)| = O(|T_i|^2)$. □

Let l_X be the smallest l for which all edges in $G(l)$ are traversable by robot X. Please note that $G(l_X)$ may have several connected components but G_X is the component containing s_X. Finally, let T_X be the minimum spanning tree computed by Kruskal's algorithm in G_X .

Lemma 7. *If rendezvous is feasible, $W(T_A)$ is a contiguous sub-route in $W(T_B)$.*

Proof. In this variant rendezvous is feasible iff $s_A \in G_A \subseteq G_B$ which in turn implies that T_B is a supertree of T_A. As walk $W(T_A)$ is fully constructed before $W(T_B)$ is completed the lemma follows from the construction governed by conditions (C1) and (C2). □

A pseudo-code of rendezvous Algorithm **RV5** can be found in Figure 6. The following theorem holds.

Theorem 7. *If rendezvous is feasible, i.e., $s_A \in G_B$, Algorithm* **RV5** *allows robots to meet along trajectory of length $O((k_A + k_B)^2)$.*

Proof. The meeting is forced directly by Lemma 7. As $W(T_A)$ is a contiguous fragment of $W(T_B)$ robot B while walking full length of $W(T_B)$ must also visit all nodes (one by one) on walk $W(T_A)$, leaving no room for robot A to escape. The total length of the adopted trajectories reflect the sizes of $W(T_A)$ and $W(T_B)$ and is limited to $O((k_A + k_B)^2)$. □

3.2 Node Inclusion (NI) versus Edge Inclusion (EI) Variant

We start this section with the proof that in Node Inclusion variant asynchronous rendezvous is not feasible. Please note that the proof holds also for networks with explicit labels.

Theorem 8. *In Node Inclusion variant asynchronous rendezvous is not possible even if nodes are equipped with unique labels.*

Proof. We start with a short observation. When the map of a robot is formed of a single edge any successful rendezvous protocol cannot ask the robot to stay permanently at the starting node awaiting another robot. If this was the case, the adversary could reduce the map of the other robot to a single node on the opposite side of the edge, and rendezvous would never take place.

Assume the network G is formed of six nodes $V = \{v_1, ..., v_6\}$ in which we consider different maps for robots A, B, C and D. The first map $G_A(V_A, E_A)$ is a tree rooted in v_1, where $V_A = V$, $E_A = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5), (v_1, v_6)\}$, and the starting point $s_A = v_1$. The second is a smaller tree $G_B(V_B, E_B)$ rooted in v_2, where $V_B = V \setminus \{v_1\}$ and $E_B = \{(v_2, v_3), (v_2, v_4), (v_2, v_5), (v_2, v_6)\}$, and the starting point $s_B = v_2$. The remaining maps G_C and G_D are formed of single edges (v_3, v_6) and (v_4, v_5) with staring points $s_C = v_3$ and $s_D = v_4$ respectively.

Note that sets of edges in maps are mutually exclusive. Also from the observation above we conclude that any proper rendezvous procedure must alter position of robots in maps G_C and G_D at least once. We consider two cases.

1. The rendezvous protocol asks each robot A and B to move between nodes in their maps indefinitely. In such case, since G_A, G_B do not share edges, it is enough for the adversary to prevent robots from meeting on the nodes. This can be done by not allowing (delaying) one robot to enter the next node on its route for as long as the other robot resides at this node. In this case rendezvous between robots A and B is not feasible.

2. The rendezvous protocol instructs, e.g., robot A to terminate at some node $v \in V_A$. Note that v cannot belong to both V_C and V_D and w.l.o.g. assume the latter. In such case the adversary instructs robot D to move from the starting point $s_D = v_4$ and wait on the edge until robot A arrives eventually at its final destination v. In this case rendezvous between robots A and D is not feasible.

This concludes the impossibility proof.

□

We show now that if either $E_A \subseteq E_B$ or $E_B \subseteq E_A$ rendezvous is possible. In such defined *Edge Inclusion* variant EI the two robots may have to walk a very long distance to meet.

Theorem 9. *Asynchronous rendezvous in Edge Inclusion variant, i.e., when $E_A \subseteq E_B$ or $E_B \subseteq E_A$ is feasible.*

Proof. In what follows we provide a brief description of the solution which utilises contiguous walks introduced in Section 3.1.

First note that given a particular map G all robots would construct exactly the same rendezvous walk. This is because they are indistinguishable. Thus the main challenge is to construct walks $W(G_A), W(G_B)$ for maps $G_A = (V_A, E_A), G_B = (V_B, E_B)$ respectively where $E_A \subseteq E_B$, s.t., the initial bounded in size fragment of the walk constructed for G_B contain as contiguous subwalk the walk constructed for G_A.

We propose the following recursive construction of the rendezvous walk for any input map $G_X = (V_X, E_X)$.

(1) For each $e \in E_X$ the walk $W(e)$ associated with edge e is a tour visiting this edge exactly once in each direction, where the starting point $W(e)^S$ and the finishing point $W(e)_F$ of walk $W(e)$ coincide with the endpoint of e with the smaller index in V.

(2) Consider any connected subgraph $G' = (V', E')$ of map G_X with $|E'| = k$. Assume inductively that all its k connected subgraphs $G'(1), ..., G'(k)$ of G' with $k - 1$ edges (ordered by edges removed in the lexicographical order) have already walks constructed $W(G'(1)), ..., W(G'(k))$ respectively.

The walk $W(G')$ is formed of the walks $W(G'(1)), ..., W(G'(k))$ where for all $i = 1, ..., k - 1$ $W(G'(i))_F$ is connected with $W(G'(i + 1))^S$ by the lexicographically earliest shortest path in G'.

The recursive construction admits a unique walk $W(G')$ for each subset G' of G_X, and ensures that for any subset $W(G')$ is a contiguous subwalk of walk $W(G_X)$. And finally, by adopting a rendezvous algorithm analogous to **RV5** one can secure the meeting eventually.

Note One should emphasise here that walks proposed above can be very long. For example, if $G_X = G$ is a complete graph with $|V_X| = n$ and $|E_X| = \binom{n}{2}$ the length of $W(G_X)$ is $\Omega(\binom{n}{2} \cdot (\binom{n}{2} - 1) \cdot (\binom{n}{2} - 2) \cdot ... \cdot 3 \cdot 2)$ which is at least exponential in n. □

4 Conclusion

In this paper we studied deterministic rendezvous in networks in which accessibilty to nodes and edges of participating robots may differ. We considered several variants of restricted accessibility for both synchronous and asynchronous models.

Several problems remains open. For example, whether the rendezvous protocols proposed in Section 2.2 and later are optimal. This includes the question whether the length of rendezvous walks introduced in Section 3.2 can be reduced to polynomial in the size of considered maps.

One can also consider models in which maps are not known to the robots. Another interesting question refers to better understanding of gathering more than two robots. In this setting while some robots could eventually meet in pairs, one mutually accessible location for gathering may not be available.

References

[1] C. Agathangelou, C. Georgiou, and M. Mavronicolas, A distributed algorithm for gathering many fat mobile robots in the plane, In Proc. PODC 2013, pp. 250–259.

[2] N. Agmon, and D. Peleg, Fault-Tolerant Gathering Algorithms for Autonomous Mobile Robots, SIAM J. Comput. (36), pp. 56–82, 2006.

[3] S. Alpern, The rendezvous search problem, SIAM J. Control and Optimization (33), pp. 673–683, 1995.

[4] S. Alpern, Rendezvous search on labeled networks, Naval Reaserch Logistics (49), pp. 256–274, 2002.

[5] S. Alpern, R. Fokkink, L. Gąsieniec, R. Lindelauf, and V.S. Subrahmanian, Search Theory, A Game Theoretic Perspective, Springer 2013.

[6] S. Alpern and S. Gal, The Theory of Search Games and Rendezvous, Kluwer Academic Publisher, 2002.

[7] E. Anderson and S. Fekete, Asymmetric rendezvous on the plane, In Proc. Symp. on Computational Geometry 1998, pp. 365–373.

[8] E. Anderson and S. Fekete, Two-dimensional rendezvous search, Operations Research (49:1), pp. 107–118, 2001.

[9] E. Anderson and R. Weber, The rendezvous problem on discrete locations, Journal of Applied Probability (28), pp. 839–851, 1990.

[10] V. Baston and S. Gal, Rendezvous on the line when the players' initial distance is given by an unknown probability distribution, SIAM J. Control and Optimization (36), pp. 1880–1889, 1998.

[11] V. Baston and S. Gal, Rendezvous search when marks are left at the starting points. Naval Reaserch Logistics (48), pp. 722–731, 2001.

[12] S. Chen, A. Russell, A. Samanta, and R. Sundaram, Deterministic Blind Rendezvous in Cognitive Radio Networks, In Proc. ICDCS 2014, pp. 358–367.

[13] A. Collins, J. Czyżowicz, L. Gąsieniec, and A. Labourel, Tell Me Where I Am So I Can Meet You Sooner, In Proc. ICALP 2010, pp. 502–514.

[14] A. Collins, J. Czyżowicz, L. Gąsieniec, A. Kosowski, and R.A. Martin, Synchronous Rendezvous for Location-Aware Agents, In Proc. DISC 2011, pp. 447–459.

[15] J. Czyżowicz, L. Gąsieniec, and A. Pelc, Gathering few fat mobile robots in the plane, Theoretical Computer Science, (410:6–7), pp. 481–499, 2009.

[16] J. Czyżowicz, A. Kosowski, and A. Pelc, How to meet when you forget: Log-space rendezvous in arbitrary graphs, Distributed Computing (25), pp. 165–178, 2012.

[17] J. Czyżowicz, A. Labourel, and A. Pelc, How to meet asynchronously (almost) everywhere, ACM Transactions on Algorithms (8), article 37, 2012.

[18] G. D'Angelo, G. Di Stefano, and A. Navarra, Gathering on rings under the look-compute-move model, Distributed Computing, 27(4):255-285, 2014.

[19] G. D'Angelo, G. Di Stefano, R. Klasing, and A. Navarra, Gathering of robots on anonymous grids and trees without multiplicity detection. Theoretical Computer Science 610, 158âĂŞ 168, 2016.

[20] S. Das, D. Dereniowski, A. Kosowski, and P. Uznanski, Rendezvous of Distance-Aware Mobile Agents in Unknown Graphs, In Proc. SIROCCO 2014, pp. 295–310.

[21] S. Das, F. L. Luccio, and E. Markou, Mobile Agents Rendezvous in Spite of a Malicious Agent, In Proc. ALGOSENSORS 2015, pp. 211–224.

[22] B. Degener, B. Kempkes, F. Meyer auf der Heide, A local $O(n^2)$ gathering algorithm, In Proc. SPAA 2010, pp. 217–223.

[23] D. Dereniowski, R. Klasing, A. Kosowski, and Ł. Kuszner, Rendezvous of Heterogeneous Mobile Agents in Edge-Weighted Networks, Theoretical Computer Science (608), pp. 219–230, 2015.

[24] Y. Dieudonné, A. Pelc, and D. Peleg, Gathering Despite Mischief, ACM Trans. Algorithms (11:1), pp. 1–28, 2014.

[25] G. Di Stefano and A. Navarra, Gathering of oblivious robots on infinite grids with minimum traveled distance. Information and Computation, 254:377-391, 2017.

[26] G. Di Stefano and A. Navarra, Optimal gathering of oblivious robots in anonymous graphs and its application on trees and rings, Distributed Computing, 30(2):75-86, 2017.

[27] S. Elouasbi, and A. Pelc, Deterministic Rendezvous with Detection Using Beeps. In Proc. ALGOSENSORS 2015, pp. 85–97.

[28] S. Elouasbi, and A. Pelc, Deterministic meeting of sniffing agents in the plane. In Proc. SIROCCO 2016, pp. 212–227.

[29] A. Farrugia, L. Gąsieniec, Ł. Kuszner, and E. Pacheco, Deterministic Rendezvous in Restricted Graphs, In Proc. SOFSEM 2015, pp. 189–200.

[30] O. Feinerman, A. Korman, S. Kutten, and Y. Rodeh. Fast rendezvous on a cycle by agents with different speeds. In *ICDCN*, volume 8314 of *LNCS*, pages 1–13, 2014.

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

[32] B. Katreniak, Convergence with Limited Visibility by Asynchronous Mobile Robots, In Proc. SIROCCO 2011, pp. 125–137.

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

[34] E. Kranakis, D. Krizanc, S. Rajsbaum, Mobile Agent Rendezvous: A Survey, In Proc. SIROCCO 2006, pp. 1–9.

[35] Z. Lin, H. Liu, X. Chu, and Y-W. Leung, Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks, In Proc. INFOCOM 2011, pp. 2444–2452.

[36] A. Miller and A. Pelc, Tradeoffs between cost and information for rendezvous and treasure hunt, J. Parallel Distrib. Comput. (83), pp. 159–167, 2015.

[37] A. Miller and A. Pelc, Time versus cost tradeoffs for deterministic rendezvous in networks, Distributed Computing (29:1), pp. 51-64, 2016.

[38] A. Pelc, Deterministic rendezvous in networks: A comprehensive survey, Networks (59), pp. 331–347, 2012.

[39] T. Schelling, The Strategy of Conflict, Harvard Univ. Press, Cambridge, 1960.

[40] Y. Yamauchi, T. Izumi and S. Kamei, Mobile Agent Rendezvous on a Probabilistic Edge Evolving Ring, In Proc. ICNC 2012, pp. 103–112.