Theoretical Computer Science 608 (2015) 219–230

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Rendezvous of heterogeneous mobile agents in edge-weighted networks ☆

Dariusz Dereniowski [a], Ralf Klasing [b], Adrian Kosowski [c], Łukasz Kuszner [a,*]

[a] Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
[b] LaBRI, CNRS and University of Bordeaux, 351, cours de la Libération, 33405 Talence cedex, France
[c] GANG Project, Inria Paris and LIAFA, CNRS and Paris Diderot University, Case 7014, 75205 Paris Cedex 13, France

ARTICLE INFO

Article history:
Received 27 November 2014
Received in revised form 6 May 2015
Accepted 9 May 2015
Available online 6 June 2015

Keywords:
Distributed algorithm
Mobile agents
Rendezvous

ABSTRACT

We introduce a variant of the deterministic rendezvous problem for a pair of heterogeneous agents operating in an undirected graph, which differ in the time they require to traverse particular edges of the graph. Each agent knows the complete topology of the graph and the initial positions of both agents. The agent also knows its own traversal times for all of the edges of the graph, but is unaware of the corresponding traversal times for the other agent. The goal of the agents is to meet on an edge or a node of the graph. In this scenario, we study the time required by the agents to meet, compared to the meeting time T_{OPT} in the offline scenario in which the agents have complete knowledge about each others' speed characteristics. When no additional assumptions are made, we show that rendezvous in our model can be achieved after time $O(nT_{\mathrm{OPT}})$ in an n-node graph, and that such time is essentially in some cases the best possible. However, we prove that the rendezvous time can be reduced to $\Theta(T_{\mathrm{OPT}})$ when the agents are allowed to exchange $\Theta(n)$ bits of information at the start of the rendezvous process. We then show that under some natural assumption about the traversal times of edges, the hardness of the heterogeneous rendezvous problem can be substantially decreased, both in terms of time required for rendezvous without communication, and the communication complexity of achieving rendezvous in time $\Theta(T_{\mathrm{OPT}})$.

1. Introduction

Solving computational tasks using teams of agents deployed in a network gives rise to many problems of coordinating actions of multiple agents. Frequently, the communication capabilities of agents are extremely limited, and the exchange of large amounts of information between agents is only possible while they are located at the same network node. In the rendezvous problem, two identical mobile agents, initially located in two nodes of a network, move along links from node

☆ An extended abstract of this work appeared in Proc. 21st International Colloquium on Structural Information and Communication Complexity (SIROCCO 2014) [15]. Research partially supported by the Polish National Science Centre grant DEC-2011/02/A/ST6/00201 and by the ANR project DISPLEXITY (ANR-11-BS02-014). This study has been carried out in the frame of the "Investments for the future" Programme IdEx Bordeaux – CPU (ANR-10-IDEX-03-02). Dariusz Dereniowski was partially supported by a scholarship for outstanding young researchers funded by the Polish Ministry of Science and Higher Education.

* Corresponding author.
E-mail addresses: deren@eti.pg.gda.pl (D. Dereniowski), ralf.klasing@labri.fr (R. Klasing), adrian.kosowski@inria.fr (A. Kosowski), kuszner@eti.pg.gda.pl (Ł. Kuszner).

http://dx.doi.org/10.1016/j.tcs.2015.05.055
0304-3975/

to node, with the goal of occupying the same node at the same time. Such a question has been studied in various models, contexts and applications [1].

In this paper we focus our attention on heterogeneous agents in networks, where the time required by an agent to traverse an edge of the network depends on the properties of the traversing agent. In the most general case we consider, the traversal time associated with every edge and every agent operating in the graph may be different. Scenarios in which traversal times depend on the agent are easy to imagine in different contexts. In a geometric setting, one can consider a road connection network, with agents corresponding to different types of vehicles moving in an environment. One agent may represent a typical road vehicle which performs very well on paved roads, but is unable to traverse other types of terrain. By contrast, the other agent may be a specialized mobile unit, such as a vehicle on caterpillars or an amphibious vehicle, which is able to traverse different types of terrain with equal ease, but without being capable of developing a high speed. In a computer network setting, agents may correspond to software agents with different structure, and the transmission times of agents along links may depend on several parameters of the link being traversed (transmission speed, transmission latency, ability to handle data compression, etc.).

In general, it may be the case that one agent traverses some links faster than the other agent, but that it traverses other links more slowly. We will also analyze more restricted cases, where we are given some a priori knowledge about the structure of the problem. Specially, we will be interested in the case of *ordered agents*, i.e., where we assume that one agent is always faster than the other one, and the case of *ordered edges*, where we assume that if in a fixed pair of links, one agent takes more time to traverse the first link, the same will also be true for the other agent.

We study the rendezvous problem under the assumption that each agent knows the complete topology of the graph and its traversal times for all edges, but knows nothing about the traversal times or the initial location of the other agent. In all of the considered cases, we will ask about the best possible time required to reach rendezvous, compared to that in the "offline scenario", in which each of the agents also has complete knowledge of the parameters of the other agent. We will also study how this time can be reduced by allowing the agents to communicate (exchange a certain number' of bits at a distance) at the start of the rendezvous process.

1.1. The model and the problem

Let us consider a simple graph $G = (V, E)$ and its weight functions $w_A : E \mapsto \mathbb{N}_+$ and $w_B : E \mapsto \mathbb{N}_+$, where $\mathbb{N}_+$ is the set of positive integers. Let $s_A, s_B \in V$, $s_A \neq s_B$, be two distinguished nodes of G – the agents' A and B starting nodes. We assume that initially an agent $K \in \{A, B\}$ knows the graph G, s_A, s_B and w_K. Thus, A knows w_A but it does not know w_B, and B knows w_B but it does not now w_A. We assume that the nodes of G have unique identifiers and that G is given to each agent together with the identifiers. The latter in particular implies that the agents have unique identifiers – they can 'inherit' the identifiers of the nodes s_A and s_B. Also, the agents do not see each other unless they meet. Both agents are non-oblivious, i.e., equipped with memories which persist over successive time steps.

The weight functions indicate the time required for A and B to move along edges. That is, given an edge $e = \{u, v\}$, an agent $K \in \{A, B\}$ needs $w_K(e)$ units of time to move along e (in any direction). We assume that both agents start their computation at time 0 by exchanging messages. The time required to send and to receive a message is negligible.

Once an agent $K \in \{A, B\}$ is located at a node v, it can do one of the following *actions*:

- the agent can wait $t \in \mathbb{N}_+$ units of time at v; after time t the agent will decide on performing another action,
- the agent can start a movement from v to one of its neighbors u; in such case the agent moves with the uniform speed from v to u along the edge $\{u, v\}$ and after $w_K(\{v, u\})$ units of time K arrives at u and then performs its next action.

While an agent is performing its local computations preceding an action, it has access to all messages sent by the other agent at time 0. We assume that the time of agent's computations preceding an action is negligible.

We say that A and B *rendezvous at time t* (or simply *meet*) if they share the same location at time t,

- they both are located at the same node at time t, or
- $K \in \{A, B\}$ started a movement from u_K to v_K at time $t_K < t$, $u_A = v_B$, $v_A = u_B$, $e = \{u_A, v_A\}$, $t_K + w_K(e) < t$ and $\frac{t-t_A}{w_A(e)} = 1 - \frac{t-t_B}{w_B(e)}$ (informally speaking, the agents 'pass' each other on e as they start from opposite endpoints of e), or
- $K \in \{A, B\}$ started a movement from u to v at time $t_K < t$, $e = \{u, v\}$, $t_K + w_K(e) < t$ and $\frac{t-t_A}{w_A(e)} = \frac{t-t_B}{w_B(e)}$ (informally speaking, both agents start at the same endpoint but the one of them 'catches up' the other: $t_A < t_B$ and $w_A(e) > w_B(e)$, or $t_A > t_B$ and $w_A(e) < w_B(e)$).

Observe that the last case is not possible in an optimum offline solution, as the agents could rendezvous earlier in the vertex u.

We are interested in the following problem:

> Given two integers b and t, does there exist a deterministic algorithm whose execution by A and B guarantees that the agents send to each other at time 0 messages consisting of at most b bits in total, and A and B meet after time at most t?

Given an algorithm for the agents, we refer to the *rendezvous time* of an algorithm as the minimum time length t such that the agents meet at time t as a result of the execution of the algorithm. The total number of bits sent between the agents is the *communication complexity* of the algorithm.

1.2. Related work

The rendezvous problem has been thoroughly studied in the literature in different contexts. In a general setting, the rendezvous problem was first mentioned in [28]. Authors investigating rendezvous (cf. [1] for an extensive survey) considered either the geometric scenario (rendezvous in an interval of the real line, see, e.g., [5,6,21], or in the plane, see, e.g., [2,3]) or the graph scenario (see, e.g., [16,20,25]). A natural extension of the rendezvous problem is that of gathering [19,23,26,30], when more than two agents have to meet in one location.

Rendezvous in anonymous graphs In the anonymous graph model, the agents rely on local knowledge of the graph topology, only. Nodes have no unique identifiers, and maintain only a local labeling of outgoing edges (ports) leading to their neighbors. When studying the feasibility and efficiency of deterministic rendezvous in anonymous graphs, a key problem which needs to be resolved is that of breaking symmetry. Without resorting to marking nodes, this can be achieved by taking advantage of the different labels of agents [27,16,25]. Labeled agents allowed to mark nodes using whiteboards were considered in [31]. Rendezvous of labeled agents using variants of Universal Exploration Sequences was also investigated in [25,29] in the synchronous model, who showed that such meeting can be achieved in time polynomial in the number of nodes of the graph and in the length of the smaller of the labels of the agents. For the case of unlabeled agents, rendezvous is not always feasible when the agents move in synchronous rounds and are allowed only to meet on nodes. However, for any feasible starting configuration, rendezvous of anonymous agents can be achieved in polynomial time, and even more strongly, using only logarithmic memory space of the agent [13]. In the asynchronous scenario, it has recently been shown that agents can always meet within a polynomial number of moves if they have unique labels [17]. For the case of anonymous agents, the class of instances for which asynchronous rendezvous is feasible is quite similar to that in the synchronous case, though under the assumption that agents are also allowed to meet on edges (which appears to be indispensable in the asynchronous scenario), certain configurations with a mirror-type symmetry also turn out to be gatherable [22].

Location-aware rendezvous The anonymous scenario may be sharply contrasted with the case in which the agent has full knowledge of the map of the environment, and knows its position within it. Such assumption, partly fueled by the availability and the expansion of the Global Positioning System (GPS), is sometimes called the *location awareness* of agents or nodes of the network. Thus, the only unknown variable is the initial location of the other agent. In [10,4] the authors study the rendezvous problem of location-aware agents in the asynchronous case. The authors of [10] introduced the concept of covering sequences that permitted location aware agents to meet along the route of polynomial length in the initial distance d between the agents for the case of multi-dimensional grids. Their result was further advanced in [4], where the proposed algorithm provides a route, leading to rendezvous, of length being only a polylogarithmic factor away from the optimal rendezvous trajectory. The synchronous case of location-aware rendezvous was studied in [9], who provided algorithms working in linear time with respect to the initial distance d for trees and grids, also showing that for general networks location-aware rendezvous carried a polylogarithmic time overhead with respect to n, regardless of the initial distance d.

Problems for heterogeneous agents Recently, scenarios with agents having different capabilities have been also studied. In [14] the authors considered multiple colliding robots with different velocities traveling along a ring with a goal to determine their initial positions and velocities. The authors of [18] studied the case of two robots moving around an anonymous cycle, where different speeds allowed them to break symmetry. Mobile agents with different speeds were also studied in the context of patrolling a boundary, see e.g. [12,24]. In [11] agents capable of traveling in two different modes that differ with maximal speeds were considered in the context of searching a line segment. We also mention that speed, although very natural, is not the only attribute that can be used to differentiate the agents. For example, authors in [7] studied robots with different ranges or, in other words, with different battery sizes limiting the distance that a robot can travel.

1.3. Additional notation

Let $T_K(u, v, w)$, $K \in \{A, B\}$, denote the minimum time required by agent K to move from u to v in G with a weight function w. If $w = w_K$, then we write $T_K(u, v)$ in place of $T_K(u, v, w_K)$, $K \in \{A, B\}$. In other words $T_K(u, v)$ is the length of the shortest path from u to v in G with weight function w_K, where the length of a path composed of edges $e_1, \ldots, e_l$ is $\sum_{j=1}^{l} w_K(e_j)$. We use the symbol $T_{\mathrm{OPT}}(s_A, s_B)$ to denote the minimum time for rendezvous in the off-line setting where agents that are initially placed on s_A and s_B know all parameters. We will skip starting positions if it will not lead to confusion writing simply T_{OPT}. Denote also $M_K := \max\{w_K(e) \mid e \in E\}$, $K \in \{A, B\}$, and let $M := \max\{M_A, M_B\}$. All logarithms have base 2, i.e., we write for brevity log in place of $\log_2$.

The following lemma, informally speaking, implies that we do not have to consider scenarios in which rendezvous occurs on edges, and by doing so we restrict ourselves to solutions among which there exists one that is within a constant factor from an optimal one. Let

Table 1
Summary of results (n is the number of nodes of the input graph).

	Communication complexity for rendezvous in time $\Theta(T_{\mathrm{OPT}})$	Rendezvous time in case of no communication
Case 1: arbitrary	$O(n \cdot (\log\log(M \cdot n)))$ (Theorem 2) $\Omega(n)$ (Theorem 3)	$\Theta(n \cdot T_{\mathrm{OPT}})$ (Theorems 9, 10)
Case 2: ordered edges	$O(\log\log M + \log^2 n)$ (Theorem 4) $\Omega(\log n)$ (Theorem 5)	$O(n \cdot T_{\mathrm{OPT}})$ (Theorem 9) $\Omega(\sqrt{n} \cdot T_{\mathrm{OPT}})$ (Theorem 11)
Case 3: ordered agents	none (Theorem 8)	$\Theta(T_{\mathrm{OPT}})$ (Theorem 8)

$$T_{\mathrm{RV}}(s_A, s_B, v) = \max\{T_A(s_A, v), T_B(s_B, v)\}$$

denote the minimum time for rendezvous at v.

Let any node u that minimizes the $T_{\mathrm{RV}}(s_A, s_B, u)$ be called a *rendezvous node*.

Lemma 1. *For each graph $G = (V, E)$ and for each $s_A, s_B \in V$, if $u \in V$ is the rendezvous node, then $T_{\mathrm{RV}}(s_A, s_B, u) \leq 2T_{\mathrm{OPT}}(s_A, s_B)$.*

Proof. If the two agents can achieve rendezvous on a node in time $T_{\mathrm{OPT}}(s_A, s_B)$, then the lemma follows and hence we assume in the following that rendezvous occurs on an edge. For $K \in \{A, B\}$, let v_K be the last node visited by K prior to rendezvous that the two agents achieve in time $T_{\mathrm{OPT}}(s_A, s_B)$. Observe that $v_A \neq v_B$ and $e = \{v_A, v_B\} \in E$.

In an optimum solution at least one of the agents traversed at least half of e, so

$$2T_{\mathrm{OPT}}(s_A, s_B) \geq \min\{T_A(s_A, v_B), T_B(s_B, v_A)\}. \tag{1}$$

Moreover, $T_A(s_A, v_B) > T_B(s_B, v_B)$ and $T_B(s_B, v_A) > T_A(s_A, v_A)$, so

$$\min\{T_A(s_A, v_B), T_B(s_B, v_A)\} \tag{2a}$$

$$= \min\{\max\{T_A(s_A, v_B), T_B(s_B, v_B)\}, \max\{T_B(s_B, v_A), T_A(s_A, v_A)\}\}$$

$$= \min\{T_{\mathrm{RV}}(s_A, s_B, v_B), T_{\mathrm{RV}}(s_A, s_B, v_A)\}. \tag{2b}$$

If u is a rendezvous node, then

$$\min\{T_{\mathrm{RV}}(s_A, s_B, v_B), T_{\mathrm{RV}}(s_A, s_B, v_A)\} \geq T_{\mathrm{RV}}(s_A, s_B, u).$$

This, together with (1) and (2b), prove the lemma. □

1.4. Possible restrictions on weight functions

Arbitrary weight functions might cause very bad performance of rendezvous (see Theorems 3 and 10). Thus, beside the arbitrary case, we will be interested in restricted cases, namely:

1. w_A and w_B are *arbitrary functions*,
2. $\forall_{e_1,e_2 \in E}\ w_A(e_1) < w_A(e_2) \iff w_B(e_1) < w_B(e_2)$,
3. $\forall_{e \in E}\ w_A(e) \leq w_B(e)$ or $\forall_{e \in E}\ w_B(e) \leq w_A(e)$.

Case 1 reflects the situation where both agents and edges are not related in terms of time needed to move along them. Whenever two functions have the property case 2, we will refer to the problem instance as the case of *ordered edges*. Informally, in such scenario both agents obtain the same ordering of edges (up to resolving ties) with respect to their weights. The last case reflects the situation where one of the agents is always at least as fast as the other one. Instances with this property are referred to as the cases of *ordered agents*.

1.5. Our results

In this work we analyze the following two extreme scenarios. In the first scenario (the middle column in Table 1) we consider the communication complexity of algorithms that guarantee that rendezvous occurs in time $\Theta(T_{\mathrm{OPT}})$ regardless of the starting positions. In the second scenario (the third column) we provide bounds on the rendezvous time in case when the agents send no messages to each other.

2. Communication complexity for $\Theta(T_{\mathrm{OPT}})$ time

In this section we determine upper and lower bounds for communication complexity of algorithms that achieve rendezvous in asymptotically optimal time. Section 2 is subdivided into three parts reflecting the three cases of weight functions we consider.

2.1. The case of arbitrary functions

We start by giving an upper bound on communication complexity of asymptotically optimal rendezvous. Our method is constructive, i.e., we provide an algorithm for the agents (see proof of Theorem 2). Then, (cf. Theorem 3) we give the corresponding lower bound.

Theorem 2. *There exists an algorithm that guarantees rendezvous in $\Theta(T_{\mathrm{OPT}})$ time and has communication complexity $O(n(\log\log(M\cdot n)))$ for arbitrary functions.*

Proof. We start the proof with an informal description of the algorithm described below. Agent A sends to B the approximate value of its distance from each node v_j; this approximate value differs by a factor at most two from $T_A(s_A, v_j)$. Since the agent B sends to A his approximate distances from nodes as well, both agents use these values to estimate approximate rendezvous times at all nodes and select one that minimizes it.

Let $I_0 = [0, 1]$, and for $j > 0$ let $I_j = (2^{j-1}, 2^j]$. Denote $V = \{v_1, \ldots, v_n\}$, where the vertices are ordered according to their identifiers. We first formulate an algorithm and then we prove that it has the required properties. We assume that A is the executing agent and B is the other agent (the algorithm for B is analogous).

1. For each $j = 1, \ldots, n$ (in this order) send to B the integer $r(A, j)$ such that $T_A(s_A, v_j) \in I_{r(A,j)}$.
2. After receiving the corresponding messages from B, construct $T' : V \mapsto \mathbb{N}_+$ such that
$$T'(v_j) := \max\{2^{r(A,j)}, 2^{r(B,j)}\}, \quad j \in \{1, \ldots, n\}.$$
3. Find a node v_ρ with minimum value of $T'(v_\rho)$. If more than one such node v_ρ exists, then take v_ρ to be the one with minimum identifier.
4. Go to v_ρ along a shortest path and stop.

Note that both agents compute the same function T'. This in particular implies that the same vertex v_ρ, to which each agent goes, is selected by both agents. Hence, the agents rendezvous at v_ρ. The transmission of $r(K, j)$ requires $O(\log\log(M\cdot n))$ bits because $r(K, j) = O(\log(M\cdot n))$ for each $K \in \{A, B\}$ and $j \in \{1, \ldots, n\}$. Thus, the communication complexity of the algorithm is $O(n\log\log(M\cdot n))$.

We now give an upper bound on the rendezvous time at v_ρ. By definition, for each $j \in \{1, \ldots, n\}$ and for each $K \in \{A, B\}$ we have

$$2^{r(K,j)-1} \le T_K(s_K, v_j) \le 2^{r(K,j)}.$$

Thus, having in mind that

$$T_{\mathrm{RV}}(s_A, s_B, v) = \max\{T_A(s_A, v), T_B(s_B, v)\},$$

we obtain:

$$\frac{1}{2}T'(v_j) \le T_{\mathrm{RV}}(s_A, s_B, v_j) \le T'(v_j), \quad j \in \{1, \ldots, n\}. \tag{3}$$

Now, let u be a rendezvous node. By (3), the choice of index ρ, again by (3) and by Lemma 1 we obtain

$$T_{\mathrm{RV}}(s_A, s_B, v_\rho) \le T'(v_\rho) \le T'(u) \le 2T_{\mathrm{RV}}(s_A, s_B, u) \le 4T_{\mathrm{OPT}}(s_A, s_B),$$

which completes the proof. □

Theorem 3. *Every algorithm that guarantees rendezvous in time $o(\frac{n}{\log n}T_{\mathrm{OPT}})$ has communication complexity $\Omega(n)$ for some n-node graphs.*

Proof. Let $\mathcal{G}$ be the class of graphs with weights such that each $G \in \mathcal{G}$ is a complete bipartite graph $K_{2,n}$ with

$$V = \{s_A, s_B, v_1, v_2, \ldots, v_n\}$$

and $E = E_A \cup E_B$, where

$$E_K = \big\{\{s_K, v_j\} \,\big|\, j \in \{1, 2, \ldots, n\}\big\}, \quad K \in \{A, B\},$$

and, for each $K \in \{A, B\}$, $w_K(e) = X$ for each $e \in E \setminus E_K$ and $w_K(e) \in \{1, X\}$ for each $e \in E_K$, where X is a sufficiently big integer, e.g., $X = n$.

Note that for each $G \in \mathcal{G}$, $T_{\mathrm{OPT}} \in \{1, X\}$. Moreover, $T_{\mathrm{OPT}} < X$ if and only if there exists an index $j \in \{1, \ldots, n\}$ such that $w_A(\{s_A, v_j\}) = w_B(\{s_B, v_j\}) = 1$. Deciding the existence of such an index j solves the well-known problem of set

disjointness (cf. e.g. [8] for a recent survey of results) and requires more than n bits to be transmitted between the two players in a deterministic communication complexity scenario.

Indeed, suppose for a contradiction that there exists an algorithm $\hat{\mathcal{A}}$ such that its time complexity is $o(\frac{n}{\log n}T_{\mathrm{OPT}})$ and its communication complexity is $o(n)$. Since $T_{\mathrm{OPT}} = 1$, we have that $\hat{\mathcal{A}}$ finishes in time $o(\frac{n}{\log n})$, thus the total number of vertices visited by agents A and B during the operation of the algorithm is $o(\frac{n}{\log n})$. The visited vertices are also the only candidates for rendezvous after the communication phase of the algorithm, thus we can obtain a solution to the disjoint set problem as follows. Each of the players, Alice (A) and Bob (B), given their respective set $S_A, S_B \subseteq \{1, \ldots, n\}$, simulates the operation of the corresponding agent $K \in \{A, B\}$ in algorithm $\hat{\mathcal{A}}$ on the graph $G = K_{2,n}$ with weights $w_K(\{s_K, v_j\}) = 1$ for edges with all indices such that $j \in S_K$, and weight X for all other edges. Algorithm $\hat{\mathcal{A}}$ is run under the assumption that each agent K starts from node v_K and every message exchanged between agents A and B in the operation of algorithm $\hat{\mathcal{A}}$ is simulated by a message transmitted between Alice and Bob (in the corresponding direction). Each party K then sends an additional $o(n)$ bits to the other, representing the list of identifiers of vertices visited by their agent K during simulation of the $o(\frac{n}{\log n})$ steps of algorithm $\hat{\mathcal{A}}$, under the assumption of lack of rendezvous. A correct answer to the set disjointness problem is obtained by giving a negative reply if and only if one of the locations v_j visited by Bob's agent corresponds to an index $j \in S_A$. In this way, we have obtained a protocol for solving set disjointness in $o(n)$ communicated bits, a contradiction. □

2.2. The case of ordered edges

Theorem 4. *There exists an algorithm that guarantees rendezvous in $\Theta(T_{\mathrm{OPT}})$ time and has communication complexity $O(\log\log M + \log^2 n)$ in case of ordered edges.*

Proof. Let $I_0 = [0, 1]$, and for $j > 0$ let $I_j = (2^{j-1}, 2^j]$. For $K \in \{A, B\}$ and a function $w_K : E \mapsto \mathbb{N}_+$ let $m(w_K)$ be the maximum integer such that the removal of all edges from G with weights greater than $m(w_K)$ disconnects G in such a way that s_A and s_B belong to different connected components. For $K \in \{A, B\}$ and $j \geq 0$, define $r_j^K = |\{e \in E \mid w_K(e) \in I_j\}|$.

We now give some intuitions regarding the algorithm described below. The agents aim at constructing approximations $\widetilde{w}_A$ and $\widetilde{w}_B$ of their weight functions. This is achieved by informing the other agent how many edges have weights that belong to each interval I_j. Additionally, in order to improve the communication complexity further, the agents determine the value of $c = \min\{c_A, c_B\}$ and restrict their communication to sending each other the number of edges that fall into intervals $I_{c-\lceil\log n\rceil}, \ldots, I_{c+\lceil\log n\rceil}$ and into the interval $[0, 2^{c-\lceil\log n\rceil-1}]$. As we prove later, the edges with weights in the latter interval can be used by agents without any additional limits while the edges with weights that do not fall into any of the above mentioned intervals should not be used at all. Finally, these approximations $\widetilde{w}_A$ and $\widetilde{w}_B$ possessed by both agents, and not the original functions w_A and w_B, are used to select the rendezvous node.

We now give a statement of an algorithm with communication complexity $O(\log\log M + \log^2 n)$. Then, we prove that its execution by each agent guarantees rendezvous in time $\Theta(T_{\mathrm{OPT}})$.

1. Let A be the executing agent and let B be the other agent (the statement for B is analogous). Send to B the index c_A such that $m(w_A) \in I_{c_A}$ – this requires sending $\log c_A \leq \log\log m(w_A) = O(\log\log M)$ bits. Set $c := \min\{c_A, c_B\}$ (c_B is in the corresponding message received from B).
2. Send to B the value of r_c^A.
3. For each $j \in \{1, \ldots, \lceil\log n\rceil\}$ send to B the values of r_{c+j}^A and r_{c-j}^A (this requires sending $O(\log^2 n)$ bits in total).
4. Send to B the value of $r^A := r_0^A + r_1^A + \cdots + r_{c-\lceil\log n\rceil-1}^A$ (this requires sending $O(\log n)$ bits).
5. After receiving the corresponding messages from B construct a weight function $\widetilde{w}_B : E \mapsto \mathbb{N}_+$ as follows. First, sort the edges so that $w_A(e_j) \leq w_A(e_{j+1})$ for each $j \in \{1, \ldots, |E| - 1\}$. Denote $E_0^B = \{e_1, \ldots, e_{r^B}\}$ and $E_\infty^B = E \setminus \{e \mid \widetilde{w}_B(e) \in I_0 \cup \cdots \cup I_{c+\lceil\log n\rceil}\}$. Then, $\widetilde{w}_B(e) := 0$ for each $e \in E_0^B$; $\widetilde{w}_B(e) = +\infty$ for each $e \in E_\infty^B$; and for each edge $e \in E \setminus (E_0^B \cup E_\infty^B)$ set $\widetilde{w}_B(e) := 2^{j'-1}$ if $e \in I_{j'}$ (this can be deduced from messages received from B).
6. Calculate the function $\widetilde{w}_A$ (i.e., the function that B constructs based on the information sent to B).
7. Find a node $v_\rho \in V$ such that

$$\max\{T_A(s_A, v_\rho, \widetilde{w}_A), T_B(s_B, v_\rho, \widetilde{w}_B)\}$$

   is minimum. If more than one such node exists, then take v_ρ to be the one with minimum identifier.
8. Go to v_ρ along a shortest path and stop.

Note that the communication complexity of the above algorithm is $O(\log\log M + \log^2 n)$. Also, both agents calculate $\widetilde{w}_A$ and $\widetilde{w}_B$ and hence the node v_ρ is the same for both agents, which implies that the algorithm guarantees rendezvous.

Therefore, it remains to prove that

$$\max\{T_A(s_A, v_\rho), T_B(s_B, v_\rho)\} = O(T_{\mathrm{OPT}}(s_A, s_B)).$$

Due to Lemma 1, it is enough to show that

$$\max\{T_A(s_A, v_\rho), T_B(s_B, v_\rho)\} = O(T_{\mathrm{RV}}(s_A, s_B, u)), \tag{4}$$

where u is a rendezvous node. For $K \in \{A, B\}$ and $x \in \{u, v_\rho\}$, let P_K^x be the set of edges of a shortest path from s_K to x in G with weight function w_K and let $\widetilde{P}_K^x$ be the set of edges of a shortest path from s_K to x in G with weight function $\widetilde{w}_K$. Note that (4) follows from

$$\max\left\{w_A(P_A^{v_\rho}), w_B(P_B^{v_\rho})\right\} = O\left(\max\{w_A(P_A^u), w_B(P_B^u)\}\right). \tag{5}$$

Hence, we focus on proving the latter equation. First note that for each $K \in \{A, B\}$ and $X \subseteq E$ it holds:

$$\begin{aligned} w_K(X \cap E_0^K) &\le |X| \cdot 2^{c-\lceil \log n \rceil - 1} \\ &\le \frac{|X|}{n} \cdot 2^{c-1} \le \frac{|X|}{n} \cdot \min\{m(w_A), m(w_B)\}. \end{aligned} \tag{6}$$

Thus, for each $K \in \{A, B\}$,

$$\begin{aligned} w_K(\widetilde{P}_K^{v_\rho} \cap E_0^K) &\le \frac{|\widetilde{P}_K^{v_\rho} \cap E_0^K|}{n} \cdot \min\{m(w_A), m(w_B)\} \\ &< \min\{m(w_A), m(w_B)\} \end{aligned} \tag{7}$$

because the edges in $\widetilde{P}_K^{v_\rho}$ form a path which in particular gives $|\widetilde{P}_K^{v_\rho} \cap E_0^K| < n$. Inequality (7) implies that, for each $K \in \{A, B\}$,

$$w_K(\widetilde{P}_K^{v_\rho}) \le \min\{m(w_A), m(w_B)\} + w_K(\widetilde{P}_K^{v_\rho} \setminus E_0^K). \tag{8}$$

Note that

$$P_K^u \cap E_\infty^K = \emptyset, \quad K \in \{A, B\}. \tag{9}$$

Indeed, otherwise $w_K(P_K^u) > 2^{c+\lceil \log n \rceil} \ge n2^c$ contradicting that u is a rendezvous node (the agent K with $c_K = c$ can reach the starting position of the other agent in time not greater than $(n-1)2^{c_K} = (n-1)2^c$).

Denote $Y_K = \{e \in E \mid w_K(e) > 2^{c-1}\}$ for each $K \in \{A, B\}$. We prove that

$$Y_A \cap P_A^u \neq \emptyset \quad \text{or} \quad Y_B \cap P_B^u \neq \emptyset. \tag{10}$$

Suppose for a contradiction that $Y_A \cap P_A^u = \emptyset$ and $Y_B \cap P_B^u = \emptyset$. Assume without loss of generality that $|Y_A| \le |Y_B|$ (the analysis in the opposite case is analogous). By definition of Y_A, $w_A(e) \le 2^{c-1}$ for each $e \in P_A^u$. Since we consider the case of ordered edges, $|Y_A| \le |Y_B|$ implies that $w_A(e) \le 2^{c-1}$ for each $e \in P_B^u$. This however means that a subset of edges in $P_A^u \cup P_B^u$ gives a path whose each edge e satisfies $w_A(e) \le 2^{c-1} \le 2^{c_A - 1}$, contradicting the choice of c_A. This completes the proof of (10). Note that (10) implies

$$\min\{m(w_A), m(w_B)\} \le 2^c \le 2\max\left\{w_A(P_A^u), w_B(P_B^u)\right\}. \tag{11}$$

By the definition of $P_K^{v_\rho}$ and by (8) we have

$$\begin{aligned} w_K(P_K^{v_\rho}) &\le w_K(\widetilde{P}_K^{v_\rho}) \\ &\le w_K(\widetilde{P}_K^{v_\rho} \setminus E_0) + \min\{m(w_A), m(w_B)\}. \end{aligned} \tag{12}$$

Recall that $w_K(e) \le 2\widetilde{w}_K(e)$ for each $e \in E \setminus E_0^K$ and $K \in \{A, B\}$. Thus,

$$w_K(\widetilde{P}_K^{v_\rho} \setminus E_0) \le 2\widetilde{w}_K(\widetilde{P}_K^{v_\rho} \setminus E_0). \tag{13}$$

Since $\widetilde{w}_K(e) = 0$ for each $e \in E_0^K$ and $K \in \{A, B\}$,

$$\widetilde{w}_K(\widetilde{P}_K^{v_\rho} \setminus E_0) = \widetilde{w}_K(\widetilde{P}_K^{v_\rho}). \tag{14}$$

Combining (12), (13) and (14) we

$$w_K(P_K^{v_\rho}) \le 2\widetilde{w}_K(\widetilde{P}_K^{v_\rho}) + \min\{m(w_A), m(w_B)\}. \tag{15}$$

By definition of $\widetilde{P}_K^u$ and (9),

$$\widetilde{w}_K(\widetilde{P}_K^u) \le \widetilde{w}_K(P_K^u) \le w_K(P_K^u), \quad K \in \{A, B\}. \tag{16}$$

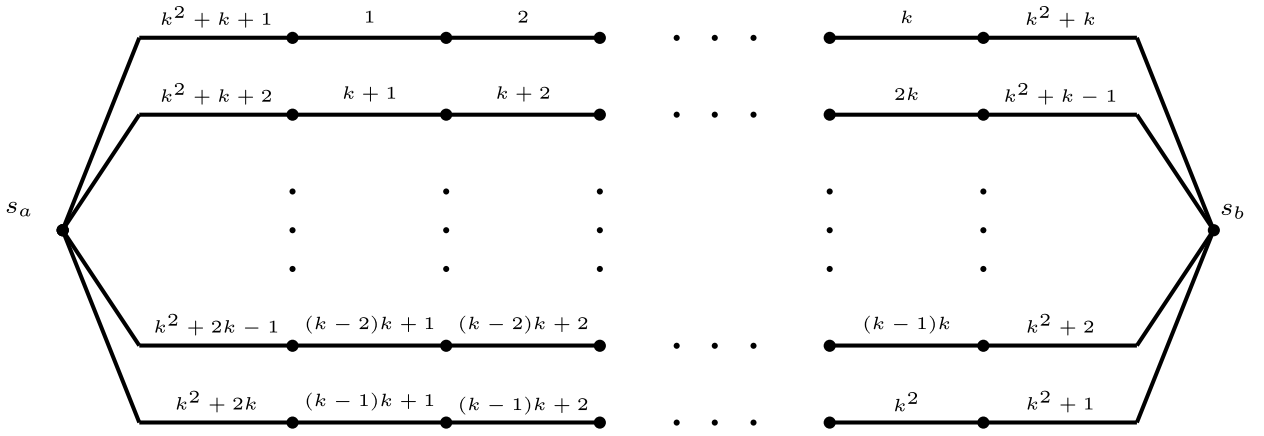

Fig. 1. The structure of the graphs in the proof of Theorems 5 and 11; the numbers give ordering of edges with respect to agents' weight functions.

By (15), the choice of v_ρ, (16) and (11),

$$\begin{aligned}
&\max\left\{w_A(P_A^{v_\rho}), w_B(P_B^{v_\rho})\right\}\\
&\quad\le \min\{m(w_A), m(w_B)\} + 2\max\left\{\widetilde{w}_A(\widetilde{P}_A^{v_\rho}), \widetilde{w}_B(\widetilde{P}_B^{v_\rho})\right\}\\
&\quad\le \min\{m(w_A), m(w_B)\} + 2\max\left\{\widetilde{w}_A(\widetilde{P}_A^{u}), \widetilde{w}_B(\widetilde{P}_B^{u})\right\}\\
&\quad\le \min\{m(w_A), m(w_B)\} + 2\max\left\{w_A(P_A^{u}), w_B(P_B^{u})\right\}\\
&\quad\le 4\max\left\{w_A(P_A^{u}), w_B(P_B^{u})\right\}.
\end{aligned}$$

This proves (5) and completes the proof of the theorem. □

Theorem 5. *In case of ordered edges each algorithm that guarantees rendezvous in time $\Theta(T_{\mathrm{OPT}})$ has communication complexity $\Omega(\log n)$ for some n-node graphs.*

Proof. Let k be a positive integer. We first define a family of graphs $\mathcal{G} = \{G_1, \ldots, G_k\}$. Each graph in $\mathcal{G}$ has the same structure, namely, the vertices s_A and s_B are connected with k node-disjoint paths but the graphs in $\mathcal{G}$ have different weight functions associated with them. Each of those paths consists of exactly $k+2$ edges (see Fig. 1).

Thus, each graph in $\mathcal{G}$ has k^2+k+2 nodes and $m = k^2+2k$ edges. The edges are denoted by $e_1, \ldots, e_m$. The location of each edge in $\mathcal{G}$ is shown in Fig. 1, where for improving the presentation we write i in place of e_i for each $i \in \{1, \ldots, m\}$. We will set the labels of the edges so that $w_K(e_1) < \cdots < w_K(e_m)$ for each $K \in \{A, B\}$. Now, for each graph in $\mathcal{G}$, we put

$$w_A(e_i) := X + i \quad \text{for each } i \in \{1, \ldots, m\},$$

where $X = k^4$. For $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, k\}$, we set the weight function w_B for G_j as follows:

$$w_B(e_i) := \begin{cases} i, & \text{for } i \le jk,\\ X+i, & \text{for } jk < i \le k^2+k-j+1,\\ kX+i, & \text{for } k^2+k-j+1 < i. \end{cases}$$

Note that $w_A(e_1) < \cdots < w_B(e_m)$ and $w_B(e_1) < \cdots < w_B(e_m)$ in each graph $G_j \in \mathcal{G}$ which ensures that all problem instances are cases of ordered edges.

Let $G_j \in \mathcal{G}$. Denote by $H_1, \ldots, H_k$ the k edge-disjoint paths connecting s_A and s_B, where $H_{j'}$ is the path containing the edge $e_{k^2+k+j'}$ incident to s_A, $j' \in \{1, \ldots, k\}$. We argue that if A and B rendezvous on a path $H_{j'}$ in time at most $kX/2$, then $j' = j$. First note that A is not able to reach any vertex adjacent to s_B in time $kX/2$. Also, $j' < j$ is not possible for otherwise B would traverse one of the edges $e_{k^2+k-j+2}, \ldots, e_{k^2+k}$, each of weight at least kX – a contradiction. Now, suppose for a contradiction that $j' > j$. Then, one of the agents traverses at least half of the path $H_{j'}$, i.e., it traverses at least $k/2+1$ of its edges. If this agent is A, then clearly rendezvous occurs not earlier than $(k/2+1)X$ – a contradiction. If this agent is B, then it does not traverse any of the edges $e_1, \ldots, e_{jk}$, since those belong to paths $H_1, \ldots, H_j$ and we have $j' > j$. Hence, by the definition of w_B, we also have rendezvous after more than $kX/2$ time units, which gives the required contradiction. We have proved that, in H_j, rendezvous is obtained before time $kX/2$ only if it occurs on the path H_j.

Observe that for n large enough and for each $G_j \in \mathcal{G}$ it holds

$$T_{\mathrm{OPT}}(s_A, s_B) < 2X.$$

To see that, let A traverse the edge e_{k^2+k+j}, and let B traverse the remaining edges of H_j, i.e.,

$$e_{(j-1)k+1}, e_{(j-1)k+2}, \ldots, e_{(j-1)k+k} \text{ and } e_{k^2+k-j+1}.$$

We have $w_A(e_{k^2+k+j}) = X + \Theta(k^2)$ and

$$w_B(e_{k^2+k-j+1}) + \sum_{l=1}^{k} w_B(e_{(j-1)k+l}) = X + k^2 + k - j + 1 + \sum_{l=1}^{k}((j-1)k+l) = X + O(k^3).$$

Since $X = k^4$, we obtain that $T_{\mathrm{OPT}}(s_A, s_B) < 2X$ for n large enough.

Suppose for a contradiction that there exists an algorithm $\mathcal{A}$ that guarantees rendezvous in time $\Theta(T_{\mathrm{OPT}})$ and has communication complexity $o(\log n)$. Let C be such a constant that the rendezvous time guaranteed by $\mathcal{A}$ is bounded by CT_{OPT}. We will show that for $n > C^4$, the algorithm $\mathcal{A}$ that sends at most $C_2 \log n$ bits, where $C_2 = \frac{3}{8}$, cannot guarantee rendezvous in time CT_{OPT}, which will give the desired contradiction.

Note that in each algorithm, and thus in particular in $\mathcal{A}$, if the agent A receives the same message from B in two different graphs in $\mathcal{G}$, then A must traverse the same sequence of edges for both graphs.

The number of all possible messages that A might receive using $C_2 \log n$ bits is at most $2^{C_2 \log n}$. Observe that $2^{C_2 \log n} < \frac{k}{C}$. Indeed,

$$C_2 = \frac{3}{8} = \frac{\log n^{3/4}}{2 \log n} < \frac{\log(n/C)}{2 \log n} < \frac{\log(k/C)}{\log n}$$

implies the required inequality. Thus, there must exist $\mathcal{G}'$, a subset of $\mathcal{G}$, with at least $C+1$ elements such that A traverses the same sequence of edges for all graphs in $\mathcal{G}'$.

To traverse an edge adjacent to s_A from H_i, the agent A requires $X + i > X$ units of time. In order to traverse any of the remaining edges, A requires to traverse back the mentioned edge first using $X + i > X$ units of time again. Hence, during $CT_{\mathrm{OPT}} < 2CX$ time, A is able to traverse at most C edges adjacent to s_A. It implies that there must exist an index j such that $G_j \in \mathcal{G}'$ and agent A does not traverse an edge from H_j adjacent to s_A. It means that A has not met B in time $2CX > CT_{\mathrm{OPT}}$, a contradiction. □

2.3. The case of ordered agents

Now we will present a general solution which achieves $\Theta(T_{\mathrm{OPT}}(s_A, s_B))$ time without communication for the case of ordered agents. This property allows us to obtain our asymptotically tight bounds both for communication complexity of optimal-time rendezvous and for optimal rendezvous time with no communication. We point out that, unlike in previous cases, the algorithms for the agents are different, i.e., A and B perform different (asymmetric) actions. We assume that the algorithm for the agent A (respectively, B) is executed by the agent whose starting node has smaller (bigger, respectively) identifier. Note that, since both agents know the graph and both starting nodes, they can correctly decide on executing an algorithm.

2.3.1. Simple algorithm

Tasks of agent A:

1. wait $T_A(s_A, s_B)$ units of time,
2. go to s_B along an arbitrarily chosen shortest path (according to the weight function w_A) from s_A to s_B, and return to s_A along the same path and stop.

Tasks of agent B:

1. wait $T_B(s_A, s_B)$ units of time,
2. go to s_A along an arbitrarily chosen shortest path (according to the weight function w_B) from s_B to s_A and stop.

Lemma 6. *Simple algorithm guarantees rendezvous in time*

$$6 \min\{T_A(s_A, s_B), T_B(s_A, s_B)\}.$$

Proof. For sure, A and B will eventually rendezvous, as both of them reach s_A and stay there. Let y be the time point at which the agents rendezvous. Let us consider agent A. It might meet B while:

1. waiting $T_A(s_A, s_B)$ units of time at s_A. In this case $y = 2T_B(s_A, s_B) \leq T_A(s_A, s_B)$.
2. moving towards s_B or on the way back to s_A. Clearly $y \leq 3T_A(s_A, s_B)$. Also, agent B at time point y is either at s_B or is moving from s_B to s_A. Thus, $y \leq 2T_B(s_A, s_B)$.
3. arriving at s_A, i.e., rendezvous occurs at s_A at the moment when A returns to s_A. It is clear that $y \leq 3T_A(s_A, s_B)$. As the agents have not met at s_A before A started moving, we have

$$T_A(s_A, s_B) \leq 2T_B(s_A, s_B).$$

So, $y \leq 6T_B(s_A, s_B)$.
4. waiting for B at s_A after the path traversals. Clearly, $y = 2T_B(s_A, s_B)$. In this case, as the agents have not met at s_B, we have

$$T_B(s_A, s_B) \leq 2T_A(s_A, s_B).$$

So, $y \leq 4T_A(s_A, s_B)$. □

We remark that the constant 6 from Lemma 6 might be reduced to $2\sqrt{2}+3$ if we would allow both agents A and B to wait a little longer in the initial state: $\sqrt{2}T_A(s_A, s_B)$ and $\sqrt{2}T_B(s_A, s_B)$ respectively.

Lemma 7. *In the case of ordered agents we have*

$$\min\{T_A(s_A, s_B), T_B(s_A, s_B)\} \leq 2T_{\mathrm{OPT}}(s_A, s_B)$$

Proof. Suppose that both agents rendezvous at x after $T_{\mathrm{OPT}}(s_A, s_B)$ units of time. If rendezvous does not occur at a node, then with a slight abuse of notation we write $T_K(u, x)$ to denote the time an agent K needs to go from a node u to x. Suppose without loss of generality that A is a 'faster' agent, i.e., $w_A(e) \leq w_B(e)$ for each edge e. This implies that $T_A(s_A, s_B) \geq T_B(s_A, s_B)$ and hence it remains to provide the upper bound on $T_A(s_A, s_B)$. Moreover, by first using the triangle inequality we have $T_A(s_A, s_B) \leq T_A(s_A, x) + T_A(x, s_B) \leq T_A(s_A, x) + T_B(x, s_B) \leq 2T_{\mathrm{OPT}}(s_A, s_B)$. □

Now, due to Lemmas 6 and 7, we are ready to conclude:

Theorem 8. *In the case of ordered agents (case 3) there exists an algorithm that guarantees rendezvous in time $\Theta(T_{\mathrm{OPT}}(s_A, s_B))$ without performing any communication.*

3. Rendezvous with no communication

3.1. The case of arbitrary functions

Theorem 9 below gives the upper bound on rendezvous time without communication. Then, Theorem 10 provides our lower bound for this case.

Theorem 9. *There exists an algorithm that without performing any communication guarantees rendezvous in time $O(n \cdot T_{\mathrm{OPT}}(s_A, s_B))$, where n is the number of nodes of the network.*

Proof. We start by giving an algorithm. Its first step is an initialization and the remaining steps form a loop. Denote $V = \{v_1, \ldots, v_n\}$.

1. Let initially $x := 1$. Let K be the executing agent.
2. For each $j \in \{1, \ldots, n\}$ do:
   2.1. If $T_K(s_K, v_j) \leq x$, then set $x' := T_K(s_K, v_j)$ and go to v_j along a shortest path. Otherwise, set $x' := 0$.
   2.2. Wait $x - x'$ time units at the current node.
   2.3. Return to s_K along a shortest path. (This step is vacuous if $x' = 0$.)
   2.4. Wait $x - x'$ time units.
3. Set $x := 2x$ and return to Step 2.

Let us introduce some notation regarding the above algorithm. We divide the time into *phases*, where the p-th phase, $p \geq 0$, consists of all time units in which both agents were performing actions determined in Step 2 for $x = 2^p$. Then, each phase is further subdivided into *stages*, where the s-th stage, $s \in \{1, \ldots, n\}$, of the p-th phase consists of all time units in which both agents were performing actions determined in Step 2 for $x = 2^p$ and $j = s$. Note that these definitions are correct since both agents simultaneously start at time 0.

First observe, by a simple induction on the total number of stages, that at the beginning of each stage each agent $K \in \{A, B\}$ is present at s_K. We now prove that both agents are guaranteed to rendezvous at a rendezvous node v in the p-th phase, where

$$2^p \geq \max\{T_A(s_A, v), T_B(s_B, v)\}.$$

Consider the s-th stage of p-th phase such that $v_s = v$. Since

$$2^p \geq \max\{T_A(s_A, v), T_B(s_B, v)\},$$

both agents reach v in at most 2^p moves. Due to the waiting time of $2^p - T_K(s_K, v)$ of agent $K \in \{A, B\}$ after reaching v, we obtain that both agents are present at v at the end of the 2^p-th time unit of the s-stage in the p-th phase. This completes the proof of the correctness of our algorithm.

It remains to bound the time in which the agents rendezvous. The duration of the p-th phase is $O(n2^p)$. The total number of phases is at most

$$P = \lceil \log \max\{T_A(s_A, v), T_B(s_B, v)\} \rceil.$$

Thus, the agents rendezvous in time

$$O(n \sum_{p=1}^{P} 2^p) = O(n2^P) = O\left(n \cdot \max\{T_A(s_A, v), T_B(s_B, v)\}\right).$$

Lemma 1 implies that the agents rendezvous in time $O(n \cdot T_{\mathrm{OPT}}(s_A, s_B))$ as required. □

Theorem 10. *Any algorithm that without performing any communication guarantees rendezvous uses time $\Omega(n \cdot T_{\mathrm{OPT}}(s_A, s_B))$, where n is the number of nodes of the network.*

Proof. Let us consider the complete bipartite graph G given in the proof of Theorem 3 with the set $V(G) = \{s_A, s_B, v_1, v_2, \ldots v_n\}$ and $E = E_A \cup E_B$, where

$$E_K = \left\{\{s_K, v_j\} \mid j \in \{1, 2, \ldots n\}\right\}, \quad K \in \{A, B\}.$$

Let $w_A(e) = X$ for each $e \in E_B$ and $w_A(\{s_A, v_i\}) = 1$ for each $e \in E_A$, where X is some sufficiently big integer, say $X = n$. We will now give a partial definition of w_B, starting with $w_B(e) = X$ for each $e \in E_A$. This weight functions will be constructed in such a way that rendezvous at time 1 is possible. Informally, we will set only one edge in E_B to have weight 1 for the agent B while the remaining edges will have weight X. This is done by analyzing possible moves of the agent A.

Now, we consider an arbitrary sequence of moves of agent A during the first n time units. Clearly, after this time, agent A is not able to reach s_B. There also exists an edge $\{s_A, v_j\} \in E_A$ that agent A performed no move along it, i.e., A did not visit v_j. We set $w_B(\{s_A, v_j\}) := 1$ and $w_B(\{s_A, v_i\}) := X$ for all $i \neq j$.

It is easy to observe that $T_{\mathrm{OPT}}(s_A, s_B)$ is equal to 1 and this time can be achieved only by a meeting at v_j. However, A and B did not rendezvous during the first n time units. Thus, there exists no algorithm that guarantees rendezvous in time $o(n \cdot T_{\mathrm{OPT}}(s_A, s_B))$. □

3.2. Lower bound for the case of ordered edges without communication

Theorem 11. *In the case of ordered edges, any algorithm that guarantees rendezvous without performing any communication uses time $\Omega(\sqrt{n} \cdot T_{\mathrm{OPT}}(s_A, s_B))$, where n is the number of nodes of the network.*

Proof. We will use the same family of graphs $\mathcal{G}$ as constructed in the proof of Theorem 5; see also Fig. 1. Recall that if A and B rendezvous on a path $H_{j'}$ in time at most $kX/2$, then $j' = j$ and $T_{\mathrm{OPT}} = O(X)$ for each $G_j \in \mathcal{G}$.

Note that, the agent A has the same input for each graph in $\mathcal{G}$ since w_A is the same for all graphs in $\mathcal{G}$. Thus, for any algorithm $\mathcal{A}$, the agent A traverses the same sequence of edges for each graph in $\mathcal{G}$. Moreover, rendezvous time bounded by $kX/2$ (see the proof of Theorem 5) implies that there exist edges adjacent to s_A that A does not traverse. In other words, there exists $j \in \{1, \ldots, k\}$ such that A traverses no edge of H_j. Therefore, we obtain that A and B cannot rendezvous in G_j in time less than $kX/2$. Since $k = \Theta(\sqrt{n})$ and rendezvous can be achieved in time $O(X)$ for each graph in $\mathcal{G}$ the proof has been completed. □

4. Final remarks

It seems that the most interesting and challenging case among the three analyzed in the paper is the one of ordered edges without communication. There is still a substantial gap between the lower and the upper bounds we have provided and we leave it an interesting open question whether there exists an algorithm with a better approximation ratio than that of $O(nT_{\mathrm{OPT}})$. It is also interesting if the upper bound M on the weights of the edges affects the communication complexity for arbitrary functions and the cases of ordered edges.

Another interesting research direction is to analyze scenarios in which we allow agents to communicate at any time. To point out an advantage that the agents may gain in such case, note that the agents can rendezvous very quickly in graphs that we used for a lower bound in the proof of Theorem 5. Indeed, transmitting just one bit in the moment correlated with the index of the preferred (optimal for rendezvous) path would help the agents to learn which path they should follow.

References

[1] S. Alpern, S. Gal, The Theory of Search Games and Rendezvous, International Series in Operations Research and Management Science, Kluwer Academic Publishers, Boston, Dordrecht, London, 2003.
[2] E. Anderson, S. Fekete, Asymmetric rendezvous on the plane, in: Proceedings of 14th Annual ACM Symposium on Computational Geometry, SoCG, 1998, pp. 365–373.
[3] E. Anderson, S. Fekete, Two-dimensional rendezvous search, Oper. Res. 49 (1) (2001) 107–118.
[4] E. Bampas, J. Czyzowicz, L. Gasieniec, D. Ilcinkas, A. Labourel, Almost optimal asynchronous rendezvous in infinite multidimensional grids, in: N.A. Lynch, A.A. Shvartsman (Eds.), DISC, in: LNCS, vol. 6343, Springer, 2010, pp. 297–311.
[5] V. Baston, S. Gal, Rendezvous on the line when the players' initial distance is given by an unknown probability distribution, SIAM J. Control Optim. 36 (6) (1998) 1880–1889.
[6] V. Baston, S. Gal, Rendezvous search when marks are left at the starting points, Naval Res. Logist. 48 (8) (2001) 722–731.
[7] J. Chalopin, S. Das, M. Mihalák, P. Penna, P. Widmayer, Data delivery by energy-constrained mobile agents, in: ALGOSENSORS, 2013, pp. 111–122.
[8] A. Chattopadhyay, T. Pitassi, The story of set disjointness, SIGACT News 41 (3) (2010) 59–85.
[9] A. Collins, J. Czyzowicz, L. Gasieniec, A. Kosowski, R.A. Martin, Synchronous rendezvous for location-aware agents, in: D. Peleg (Ed.), DISC, in: LNCS, vol. 6950, Springer, 2011, pp. 447–459.
[10] A. Collins, J. Czyzowicz, L. Gasieniec, A. Labourel, Tell me where I am so I can meet you sooner, in: S. Abramsky, C. Gavoille, C. Kirchner, F. Meyer auf der Heide, P.G. Spirakis (Eds.), ICALP (2), in: LNCS, vol. 6199, 2010, pp. 502–514.
[11] J. Czyzowicz, L. Gasieniec, K. Georgiou, E. Kranakis, F. MacQuarrie, The beachcombers' problem: walking and searching with mobile robots, in: M. Halldórsson (Ed.), SIROCCO, in: LNCS, vol. 8576, 2014, pp. 23–36.
[12] J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, Boundary patrolling by mobile agents with distinct maximal speeds, in: ESA, 2011, pp. 701–712.
[13] J. Czyzowicz, A. Kosowski, A. Pelc, How to meet when you forget: log-space rendezvous in arbitrary graphs, Distrib. Comput. 25 (2) (2012) 165–178.
[14] J. Czyzowicz, E. Kranakis, E. Pacheco, Localization for a system of colliding robots, in: ICALP (2), 2013, pp. 508–519.
[15] D. Dereniowski, R. Klasing, A. Kosowski, Ł. Kuszner, Rendezvous of heterogeneous mobile agents in edge-weighted networks, in: M. Halldórsson (Ed.), SIROCCO, in: LNCS, vol. 8576, 2014, pp. 311–326.
[16] A. Dessmark, P. Fraigniaud, D.R. Kowalski, A. Pelc, Deterministic rendezvous in graphs, Algorithmica 46 (1) (2006) 69–96.
[17] Y. Dieudonné, A. Pelc, V. Villain, How to meet asynchronously at polynomial cost, in: PODC, 2013, pp. 92–99.
[18] O. Feinerman, A. Korman, S. Kutten, Y. Rodeh, Fast rendezvous on a cycle by agents with different speeds, in: ICDCN, in: LNCS, vol. 8314, 2014, pp. 1–13.
[19] P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous robots with limited visibility, Theoret. Comput. Sci. 337 (1–3) (2005) 147–168.
[20] P. Fraigniaud, A. Pelc, Deterministic rendezvous in trees with little memory, in: Proceedings of 22nd International Conference on Distributed Computing, DISC, in: LNCS, vol. 5218, 2008, pp. 242–256.
[21] S. Gal, Rendezvous search on the line, Oper. Res. 47 (6) (1999) 974–976.
[22] S. Guilbault, A. Pelc, Asynchronous rendezvous of anonymous agents in arbitrary graphs, in: A.F. Anta, G. Lipari, M. Roy (Eds.), OPODIS, in: LNCS, vol. 7109, Springer, 2011, pp. 421–434.
[23] A. Israeli, M. Jalfon, Token management schemes and random walks yield self-stabilizing mutual exclusion, in: C. Dwork (Ed.), PODC, ACM, 1990, pp. 119–131.
[24] A. Kawamura, Y. Kobayashi, Fence patrolling by mobile agents with distinct speeds, in: ISAAC, 2012, pp. 598–608.
[25] D.R. Kowalski, A. Malinowski, How to meet in anonymous network, Theoret. Comput. Sci. 399 (1–2) (2008) 141–156.
[26] W. Lim, S. Alpern, Minimax rendezvous on the line, SIAM J. Control Optim. 34 (5) (1996) 1650–1665.
[27] G.D. Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoret. Comput. Sci. 355 (3) (2006) 315–326.
[28] T. Schelling, The Strategy of Conflict, Oxford University Press, Oxford, 1960.
[29] A. Ta-Shma, U. Zwick, Deterministic rendezvous, treasure hunts and strongly universal exploration sequences, in: Proceedings of 18th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA, 2007, pp. 599–608.
[30] L. Thomas, Finding your kids when they are lost, J. Oper. Res. Soc. 43 (6) (1992) 637–639.
[31] X. Yu, M. Yung, Agent rendezvous: a dynamic symmetry-breaking problem, in: Proceedings of 23rd International Colloquium on Automata, Languages and Programming, ICALP, in: LNCS, vol. 1099, 1996, pp. 610–621.