On asynchronous rendezvous in general graphs

Evangelos Bampas[1], Lélia Blin[2], Jurek Czyzowicz[3], David Ilcinkas[4,*], Arnaud Labourel[1,**], Maria Potop-Butucaru[5], and Sébastien Tixeuil[5]

[1] LIS, Aix-Marseille Université & CNRS, France
evangelos.bampas@gmail.com, arnaud.labourel@lif.univ-mrs.fr
[2] Sorbonne Université, Université d'Evry Val d'Essonne, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005, Paris, France
lelia.blin@lip6.fr
[3] Université du Québec, Gatineau, Canada
Jurek.Czyzowicz@uqo.ca
[4] LaBRI, CNRS and Univ. Bordeaux, France
david.ilcinkas@labri.fr
[5] Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005, Paris, France
{maria.potop-butucaru, sebastien.tixeuil}@lip6.fr

Abstract. A pair of agents (robots) are moving in a graph with the goal of meeting at the same node or while traversing the same edge. An asynchronous adversary knows the prescribed walks of the two agents and is in complete control of the speed of each agent during its walk. We provide a complete characterization of pairs of walks that enforce rendezvous against an asynchronous adversary after traversing a given number of edges. The characterization is efficient in that it can be checked in polynomial time. We argue that the certificate of rendezvous enforcement that is produced by the checking algorithm contains a wealth of information on why rendezvous is enforced.

1   Introduction

The *rendezvous* problem concerns two or more autonomous mobile entities (agents), which start from distinct points in some search space and have to meet. This problem has been studied extensively under a variety of assumptions, the most important lying in the nature of the search space (subsets of Euclidean space vs. graphs), the mode of movement of the agents (deterministic vs. randomized), and the power of the adversary (synchronous vs. asynchronous). Other aspects of the problem include the amount of knowledge that may be available to the agents about the space where they are moving, the amount of interactions with the environment, the quantity of computational resources that the agents can use in order to decide their moves, etc.

   In this paper, we focus on asynchronous rendezvous of exactly two agents in a network modeled by an undirected simple graph. We use a classical model of asynchrony that has been used in several recent papers [5,11,13,16,22,32]. According to this model, the complete sequences of edges to be followed by each agent are known to the adversary, who can vary at will the speed of the execution of that sequence. The graph where the agents are moving is assumed to be embedded in the three-dimensional Euclidean space in such a way that graph nodes correspond to points of the space and graph edges correspond to pairwise disjoint curves connecting these points (every graph admits such an embedding). The agents are modeled as points moving inside the embedding. The speed of an agent traversing an edge is completely controlled by the adversary, as long as it remains non-negative and it allows the agent to complete the edge traversal in finite time. In fact, some authors make the assumption that agents can also be moved back and forth on the curve representing an edge, but as observed by Stachowiak [32], this does not give the adversary any additional power. Underlying this model is the assumption that agents can meet not only on nodes of the graph, but also in the interior points of edges. One is forced to make this assumption, since otherwise two agents are unable to enforce asynchronous rendezvous even in the graph that consists of a single edge. The *cost* of rendezvous is measured in terms of the total number of edges traversed by both agents until they meet.

Related work   The synchronous rendezvous in graphs or *gathering* (if there are more than two agents in the system) has been extensivelly studied in the past few years. A survey of these results is proposed by Pelc in [30]. In the most recent work in this area [18], Dieudonné and Pelc propose two universal deterministic algorithms for synchronous gathering both polynomial in the size of the network.

   De Marco et al. [16] initiated the study of asynchronous deterministic rendezvous in graphs, providing algorithms for labeled agents operating in the infinite line, in a ring of unknown size, and in arbitrary graphs with known upper bound on the size. Later, Stachowiak [32] improved the algorithm for the infinite line. Czyzowicz et al. [13] proved the feasibility of rendezvous of labeled agents in graphs without knowledge on the size, and in connected terrains on the plane. Guilbault and Pelc [22] characterized the starting positions from which two anonymous agents can achieve deterministic rendezvous in an unknown graph, and gave a rendezvous algorithm that works for all such starting posi-

tions. Moreover, they presented a randomized algorithm that achieves rendezvous from all starting positions with probability 1. Collins et al. [11] gave an almost optimal algorithm for two-dimensional grids, which was generalized and further improved by Bampas et al. [5]. A deterministic asynchronous rendezvous algorithm with cost polynomial in the size of the network and in the length of the smaller label was proposed by Dieudonné et al. [19]. Also, as an application of their algorithm [19], they solve several fundamental problems such as: counting, leader election, renaming, and gossiping.

In a different model of asynchrony, that has been used mainly in the context of rendezvous of more than two agents, i.e., gathering, agents operate in Look-Compute-Move cycles. In a Look operation, an agent obtains a snapshot of the current positions of the agents in the network, then in a Compute operation it computes its next move, and finally in the Move operation it executes the move it computed. The time that elapses between consecutive operations of each agent is controlled by the adversary, thus an agent may compute a move based on information that is outdated by the time it executes that move. Gathering has been studied both on graphs [6,14,15,24,25,26,27] and on the Euclidean plane [1,3,4,7,8,9,10,12,17,20,21,23,31].

An extensive overview of results concerning mainly randomized rendezvous in continuous search spaces is contained in the book by Alpern and Gal [2]. The survey by Kranakis et al. [29] discusses and compares many of the models developed by the theoretical computer science community for rendezvous in graphs. A more recent survey by Pelc [30] contains an overview of results in the deterministic network setting. A monograph by Kranakis et al. [28] focuses on results on the rendezvous problem in ring networks. Finally, the book by Flocchini et al. [20] includes an extensive presentation of the gathering results in both continuous and discrete spaces.

Our contribution The goal of this paper is not to give an algorithm for rendezvous in a particular setting, but instead to give an algorithm to check whether two given walks in a given graph enforce rendezvous against an asynchronous adversary with a cost of at most a given integer k.

Thus, the main result of this paper is a complete characterization of pairs of walks of two agents that enforce rendezvous against an asynchronous adversary with at most a given cost. The characterization is efficient in that it can be checked in time polynomial in the length of the given walks. The certificate of rendezvous enforcement that is produced by the checking algorithm contains a wealth of information on why rendezvous is enforced. Note that the characterization we propose is different from the characterization for asynchronous rendezvous given by Guibault and Pelc [22]. Indeed, our paper answers the question of whether the adversary can avoid rendezvous given two agent walks, whereas Guibault and Pelc [22] study the possibility of deterministic rendezvous given two agent starting positions.

The outline of the paper is the following. In Section 2, we prove an equivalence between the general continuous adversary and a more restricted discrete adversary, which proves useful for our analysis. Section 3 is devoted to proving our characterization of pairs of walks that enforce rendezvous. We conclude with Section 4, where we discuss the implications of our results on the design of asynchronous rendezvous algorithms.

2 Discretization of asynchronous continuous schedulings

We consider agents moving asynchronously on anonymous, undirected, simple graphs. Let $G = (V, E)$ be such a graph. An agent can move from node $u \in V$ to node $v \in V$ if and only if there is an undirected edge $\{u, v\} \in E$.

Definition 1 (Walk). *A* walk $\mathcal{P}$ *on graph* $G = (V, E)$ *is a sequence of nodes* $(p_i)_{0 \leq i \leq r}$, *such that for each* i *in the range* $0 \leq i \leq r - 1$, $\{p_i, p_{i+1}\} \in E$. *For any non-negative integer* $k \leq r$, $\mathcal{P}[k]$ *denotes the* residual walk $(p_i)_{k \leq i \leq r}$. *The* length $|\mathcal{P}| = r$ *of the walk is either a non-negative integer for finite walks, or positive infinity for infinite walks.*

Let $\mathcal{P} = (p_i)_{0 \leq i \leq r}$ and $\mathcal{Q} = (q_i)_{0 \leq i \leq s}$ be two walks on G. We propose to establish an equivalence between two adversarial models for asynchronous rendezvous. In the first model, we assume an arbitrary non-crossing embedding η of G in $\mathbb{R}^3$.

Definition 2 (Trajectory). *If* $\mathcal{P}$ *is a walk on* G *then the corresponding* trajectory *is a curve* $\gamma_{\mathcal{P}} : \ell_{\mathcal{P}} \to \mathbb{R}^3$, *where* $\ell_{\mathcal{P}} = \mathbb{R}_+$ *if* $\mathcal{P}$ *is infinite, or* $\ell_{\mathcal{P}} = [0, r]$ *if* $\mathcal{P}$ *is of length* r. *Furthermore,* $\gamma_{\mathcal{P}}(0) = \eta(p_0)$ *and every edge is traversed in unit time with uniform speed (this enforces that for all* $k \in \mathbb{N}$, $\gamma_{\mathcal{P}}(k) = \eta(p_k)$).

Definition 3 (Rendezvous points). *Let* $\mathcal{P}$ *and* $\mathcal{Q}$ *be two walks on* G. *The* rendezvous points *of* $\mathcal{P}$ *and* $\mathcal{Q}$ *are the elements of the set* $\{(x, y) : \gamma_{\mathcal{P}}(x) = \gamma_{\mathcal{Q}}(y)\}$.

Definition 4 (Continuous adversary).

1. *A* continuous scheduling $\mathcal{S}_c$ *for* $\mathcal{P}$ *and* $\mathcal{Q}$ *is a pair of continuous, monotone, and surjective functions* $\lambda_{\mathcal{P}} : \ell \to \ell_{\mathcal{P}}$, $\lambda_{\mathcal{Q}} : \ell \to \ell_{\mathcal{Q}}$, *where* $\ell = \mathbb{R}_+$ *if either* $\mathcal{P}$ *or* $\mathcal{Q}$ *is infinite, or* $\ell = [0, 1]$ *if* $\mathcal{P}$ *and* $\mathcal{Q}$ *are both finite. Furthermore,* $\lambda_{\mathcal{P}}(0) = \lambda_{\mathcal{Q}}(0) = 0$.
2. *We say that* rendezvous occurs *under* $\mathcal{S}_c$ *at time* t *if and only if* $(\lambda_{\mathcal{P}}(t), \lambda_{\mathcal{Q}}(t))$ *is a rendezvous point of* $\mathcal{P}$ *and* $\mathcal{Q}$.
3. *The* cost of rendezvous *for* $\mathcal{P}$ *and* $\mathcal{Q}$ *under* $\mathcal{S}_c$ *is*

$$\min\left\{\lambda_{\mathcal{P}}(t) + \lambda_{\mathcal{Q}}(t) : \gamma_{\mathcal{P}}(\lambda_{\mathcal{P}}(t)) = \gamma_{\mathcal{Q}}(\lambda_{\mathcal{Q}}(t))\right\}$$

We interpret such a scheduling as the adversary moving the agents on the embedding of G in such a way that at each time $t \geq 0$, the first agent is on the point $\gamma_{\mathcal{P}}(\lambda_P(t))$ and the second agent is on the point $\gamma_{\mathcal{Q}}(\lambda_{\mathcal{Q}}(t))$. Note that, under this definition for a continuous scheduling, we assume that the adversary has the power to arbitrarily vary the speeds used by the two agents to execute their prescribed walks, provided that these speeds remain non-negative at all times. The adversary may well immobilize an agent even in the midst of traversing an edge of the graph, but cannot turn an agent back on its prescribed walk nor prevent it to reach each node of the walk in finite time. The cost of rendezvous corresponds to the sum of the number of edges traversed by each agent until the time of meeting (this number may be fractional if rendezvous occurs in the interior of an edge). We can visualize a continuous scheduling by looking at the functions $\lambda_{\mathcal{P}}$ and $\lambda_{\mathcal{Q}}$ as parametric equations defining a curve in $\ell_{\mathcal{P}} \times \ell_{\mathcal{Q}}$ (see e.g. Figure 1). Due
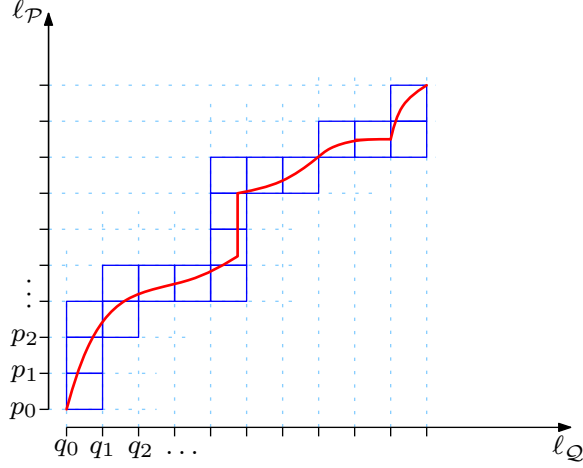
Fig. 1. Continuous scheduling.

to the monotonicity of $\lambda_{\mathcal{P}}$ and $\lambda_{\mathcal{Q}}$, the slope of this curve is always between 0 and $\frac{\pi}{2}$, inclusive.

In the second model, let $\tilde{G}$ be the graph produced by subdividing each edge of G into two parts. Let $\tilde{\mathcal{P}} = (\tilde{p}_i)_{0 \leq i \leq 2r}$ and $\tilde{\mathcal{Q}} = (\tilde{q}_i)_{0 \leq i \leq 2s}$ be the walks on $\tilde{G}$ that correspond to $\mathcal{P}$ and $\mathcal{Q}$ on G.

Definition 5 (Discrete adversary).

1. *A* discrete scheduling $\mathcal{S}_d$ *for* $\mathcal{P}$ *and* $\mathcal{Q}$ *is a sequence* $((u_i, v_i))_{0 \leq i \leq 2(r+s)}$ *of pairs of nodes of* $\tilde{G}$, *such that* $(u_0, v_0) = (\tilde{p}_0, \tilde{q}_0)$ *and, for every integer* $t \geq 0$, *if* $(u_t, v_t) = (\tilde{p}_i, \tilde{q}_j)$, *then either* $(u_{t+1}, v_{t+1}) = (\tilde{p}_{i+1}, \tilde{q}_j)$ *or* $(u_{t+1}, v_{t+1}) = (\tilde{p}_i, \tilde{q}_{j+1})$.
2. *We say that* rendezvous occurs *under* $\mathcal{S}_d$ *at time* t *if and only if* $u_t = v_t$.
3. *The* cost of rendezvous *for* $\mathcal{P}$ *and* $\mathcal{Q}$ *under* $\mathcal{S}_d$ *is* $\min\left\{\frac{t}{2} : u_t = v_t\right\}$.

The intended interpretation of a discrete scheduling is that at each time step $t \geq 0$, the adversary allows the two agents to instantaneously hop onto the nodes u_t and v_t (respectively) of $\tilde{G}$. Of course, by definition of the scheduling, only one agent really moves at each time step while the other stays on the same node. Alternatively, we can imagine that the adversary moves the agents in $\eta(G)$ as follows: in each time interval of the form $\left[t, t + \frac{1}{2}\right]$, where t is half-integral, the agent whose position changes between (u_{2t}, v_{2t}) and (u_{2t+1}, v_{2t+1}) – suppose without loss of generality that this is the agent performing the walk $\tilde{\mathcal{P}}$, moving from $\tilde{p}_i$ to $\tilde{p}_{i+1}$ – is moved monotonously from $\eta(\tilde{p}_i)$ to $\eta(\tilde{p}_{i+1})$, while the other agent does not move. Here we assume that, for odd $i = 2k + 1$, $\eta(\tilde{p}_i)$ is exactly the middle point of the embedding of the edge $(\tilde{p}_{2k}, \tilde{p}_{2k+2})$.

By the preceding discussion, it is not hard to see that any discrete scheduling can be converted to an equivalent continuous scheduling that preserves the occurrence or non-occurrence of rendezvous and also the cost of rendezvous. We now prove the converse, thus establishing the promised equivalence between the two adversarial models:

4

Theorem 1. *For every continuous scheduling $\mathcal{S}_c = (f, g)$ for $\mathcal{P}$ and $\mathcal{Q}$, there exists a discrete scheduling $\mathcal{S}_d = ((u_i, v_i))_{0 \leq i \leq 2(r+s)}$, such that:*

1. *If rendezvous occurs under $\mathcal{S}_c$ with cost $C(\mathcal{S}_c)$, then rendezvous also occurs under $\mathcal{S}_d$ with cost in the range $C(\mathcal{S}_c) - 1 < C(\mathcal{S}_d) < C(\mathcal{S}_c) + 1$.*
2. *If rendezvous does not occur under $\mathcal{S}_c$, then rendezvous does not occur under $\mathcal{S}_d$.*

Proof. We consider a grid that covers $\ell_{\mathcal{P}} \times \ell_{\mathcal{Q}}$, aligned at the points of $\ell_{\mathcal{P}}$ and $\ell_{\mathcal{Q}}$ that correspond to embeddings of nodes of G. Thus, each square in the grid is of the form $[\tilde{p}_{2j}, \tilde{p}_{2j+2})_{\ell_{\mathcal{P}}} \times [\tilde{q}_{2k}, \tilde{q}_{2k+2})_{\ell_{\mathcal{Q}}}$. We use the terms *boundary* and *interior* of a square with their usual topological meaning. For any square, the diagonal with slope $-\frac{\pi}{4}$ is called *primary*, and the diagonal with slope $\frac{\pi}{4}$ is called *secondary*. We say that a square is *traversed* by $\mathcal{S}_c$ if the corresponding curve contains any point of the square. By the monotonicity properties of the scheduling, the *entry point* of a square is always on the left or bottom boundary of the square, whereas the *exit point* of a square is always on the top or right boundary. Consequently, the next square traversed is always upward, rightward, or diagonally upward and rightward with respect to the current one.

Let $(R_i)_{i \geq 0}$ be the sequence of squares traversed by $\mathcal{S}_c$. We construct a new continuous scheduling $\mathcal{S}'_c$ by normalizing the traversal of each square R_i, in such a way that in the new scheduling the agents never move simultaneously (thus the curve of $\mathcal{S}'_c$ is a sequence of horizontal and vertical segments) and, furthermore, during the movement of one agent the other one is immobilized either on a node or in the middle point of some edge (thus the curve of $\mathcal{S}'_c$ is aligned to a finer grid that consists of squares of the form $[\tilde{p}_i, \tilde{p}_{i+1})_{\ell_{\mathcal{P}}} \times [\tilde{q}_j, \tilde{q}_{j+1})_{\ell_{\mathcal{Q}}}$). Scheduling $\mathcal{S}'_c$ can be immediately converted into a discrete scheduling $\mathcal{S}_d$ that results in rendezvous if and only if $\mathcal{S}'_c$ results in rendezvous, and the cost of rendezvous of $\mathcal{S}'_c$ and of $\mathcal{S}_d$ is exactly the same. What remains to be shown is, therefore, the equivalence between $\mathcal{S}_c$ and $\mathcal{S}'_c$.

Definition of $\mathcal{S}'_c$: We first define the entry and exit points of $\mathcal{S}'_c$ for each R_i. If the entry (resp. exit) point of $\mathcal{S}_c$ lies in the interior of some boundary edge of the square, then the entry (resp. exit) point of $\mathcal{S}'_c$ is exactly at the middle of the same edge. If the entry (resp. exit) point of $\mathcal{S}_c$ is at a corner of the square, then it coincides with the entry (resp. exit) point of $\mathcal{S}'_c$. Observe that by the continuity of $\mathcal{S}_c$, the exit point of $\mathcal{S}'_c$ from square R_i always coincides with the entry point of $\mathcal{S}'_c$ into square R_{i+1}.

Let $R_i = [\tilde{p}_{2j}, \tilde{p}_{2j+2})_{\ell_{\mathcal{P}}} \times [\tilde{q}_{2k}, \tilde{q}_{2k+2})_{\ell_{\mathcal{Q}}}$ be a square traversed by $\mathcal{S}_c$, and let $e = (u, v) = (\tilde{p}_{2j}, \tilde{p}_{2j+2})$ and $e' = (w, x) = (\tilde{q}_{2k}, \tilde{q}_{2k+2})$ be the edges in G that define R_i. The general rule for constructing the normalized traversal of R_i is the following:

> If $\mathcal{S}_c$ traverses at least one point of the interior of R_i, then $\mathcal{S}'_c$ goes from entry point to exit point, *passing through* the center of the square. The precise choice of route is arbitrary, as long as $\mathcal{S}'_c$ respects this condition, the monotonicity conditions, and is a sequence of horizontal and vertical segments aligned to the finer grid mentioned earlier.
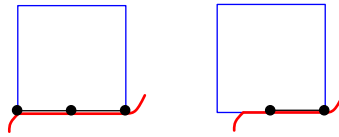
There is one exception to this rule, *viz.* if $e = e'$ and scheduling $\mathcal{S}_c$ does not touch the secondary diagonal of R_i including its endpoints, then $\mathcal{S}'_c$ goes from entry point to exit
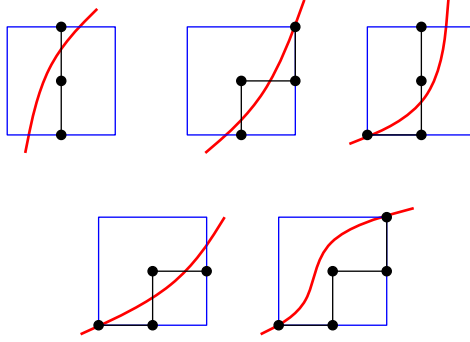
Fig. 2. Normalization of a continuous scheduling, general case.
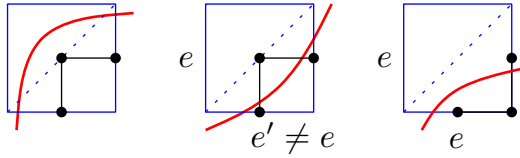
Fig. 3. Normalization of a continuous scheduling that enters from the bottom edge and exits through the right edge.

Fig. 4. Normalization of a continuous scheduling when it traverses only boundary points of the square.

point *avoiding* the center of the square. If $\mathcal{S}_c$ traverses only boundary points of R_i, then $\mathcal{S}'_c$ uses the unique route that connects its entry point to its exit point.

We illustrate the cases when the entry point lies on the bottom edge of R_i in Figures 2, 3, and 4. The remaining cases can be easily obtained by symmetry.

We call a point in R_i or on the boundary of R_i a *rendezvous point of R_i* if whenever a scheduling passes through this point, rendezvous occurs. We prove that in any R_i, the scheduling $\mathcal{S}_c$ traverses a rendezvous point if and only if the scheduling $\mathcal{S}'_c$ traverses a rendezvous point. We consider the following cases:

– $\mathcal{S}_c$ stays on the boundary of R_i, or $\mathcal{S}_c$ traverses an interior point of R_i and $e \neq e'$: The only possible rendezvous points are the corners of R_i, and by construction $\mathcal{S}'_c$ passes through exactly the same corners as $\mathcal{S}_c$.

6

– $\mathcal{S}_c$ *traverses an interior point of* R_i *and touches the secondary diagonal and* $e = e'$: In this case, rendezvous occurs under $\mathcal{S}_c$. Moreover, $e = e'$ implies that $\tilde{p}_{2j+1} = \tilde{q}_{2k+1}$, and since $\mathcal{S}'_c$ goes through the center of the square, rendezvous also occurs under $\mathcal{S}'_c$.

– $\mathcal{S}_c$ *traverses an interior point of* R_i *and does not touch the secondary diagonal and* $e = e'$: In this case, $\mathcal{S}'_c$ goes through either (v, w) or (u, x). If edge e is traversed in the same direction by both walks then $u = w$ and $v = x$ and rendezvous does not occur under $\mathcal{S}_c$ nor under $\mathcal{S}'_c$. Lastly, If edge e is traversed in opposite directions by the walks, then $u = x$ and $v = w$ and rendezvous occurs under $\mathcal{S}_c$ and under $\mathcal{S}'_c$.

The above case analysis establishes the equivalence between $\mathcal{S}'_c$ and $\mathcal{S}_c$ with respect to the occurrence of rendezvous.

As far as the cost of rendezvous is concerned, note that in all cases, if $\mathcal{S}_c$ has a rendezvous point in a square R_i, then there exists a corresponding rendezvous point of $\mathcal{S}'_c$ in the same square R_i. Furthermore, the horizontal distance between the two rendezvous points is strictly smaller than half the length of the side of R_i (in either direction), and likewise for the vertical distance. This establishes the desired inequality between the cost of rendezvous of $\mathcal{S}'_c$ and of $\mathcal{S}_c$. $\qquad\square$

3 A characterization of rendezvous-enforcing pairs of walks

In the following, we only consider walks $\tilde{\mathcal{P}}, \tilde{\mathcal{Q}}$ on a subdivided graph $\tilde{G}$ that correspond to valid walks in the original graph G. We drop the "tilde" marks for convenience, but keep in mind that the walks under consideration are walks on some subdivided graph. Therefore, every other node corresponds to the middle of some edge in the original graph, and every occurrence of such a node is immediately preceded and succeeded by nodes that correspond to the endpoints of this edge in the original graph.

Definition 6. *We say that walks* $\mathcal{P}$ *and* $\mathcal{Q}$ *enforce rendezvous with cost at most* C *if and only if any discrete scheduling for* $\mathcal{P}$ *and* $\mathcal{Q}$ *has cost at most* C.

Definition 7. *Let* $\mathcal{P} = (p_i)_{0 \le i \le r}$ *and* $\mathcal{Q} = (q_i)_{0 \le i \le s}$ *be two walks on* G. *We define the corresponding* obstruction graph $H(\mathcal{P}, \mathcal{Q}) = (U, D)$ *to be the directed graph with node set* $U = \{(i, j) : p_i = q_j\} \cup \{x, y\}$, *where* x *and* y *are two special nodes, and edge set* D *containing the following edges:*

regular edges: $((i, j), (k, l)) \in D$, *for all* $(i, j), (k, l) \in U$ *such that* $|i - k| = |j - l| = 1$.
vertical jump edges: $((i, j), (i, j')) \in D$, *for all* $(i, j), (i, j') \in U$ *such that* $j' > j$ *and* $p_i \notin \{q_{j+1}, \ldots, q_{j'-1}\}$.
horizontal jump edges: $((i, j), (i', j)) \in D$, *for all* $(i, j), (i', j) \in U$ *such that* $i' < i$ *and* $q_j \notin \{p_{i'+1}, \ldots, p_{i-1}\}$.

Furthermore, there exists an edge from node x *to each node of the form* $(0, j)$ *or* (i, s), *and also an edge from each node of the form* (r, j) *or* $(i, 0)$ *to node* y.

We visualize the obstruction graph embedded in what we call an *obstruction diagram*; see for example Figures 5 and 6. We use the term *row* k as a shorthand for the set

of nodes $U \cap \{(i, k) : 0 \leq i \leq r\}$. Similarly, *column k* stands for the set of nodes $U \cap \{(k, j) : 0 \leq j \leq s\}$. A discrete scheduling for $\mathcal{P}$ and $\mathcal{Q}$ is represented by a path from grid point $(0, 0)$ to grid point (r, s), in the same manner as in Figure 1. Rendezvous occurs if and only if the path that corresponds to the scheduling uses one of the grid points that belong to the node set of $H(\mathcal{P}, \mathcal{Q})$.

Definition 8 (Barrier). *A* barrier W *in* $H(\mathcal{P}, \mathcal{Q})$ *is a directed path from node x to node y. The* height h *of W is defined as* $h = \max \{i + j : (i, j) \in W\}$.

Remark 1. It is not hard to see that if $H(\mathcal{P}, \mathcal{Q})$ contains a barrier $W = (x, (a_0, b_0), \ldots, (a_t, b_t), y)$ of height h, then the path $\overline{W} = (x, (b_t, a_t), \ldots, (b_0, a_0), y)$ is a barrier of height h in $H(\mathcal{Q}, \mathcal{P})$. This symmetry justifies the terminology "$\mathcal{P}$ and $\mathcal{Q}$ form a barrier of height h," which we use in the sequel.

Remark 2. If $H(\mathcal{P}, \mathcal{Q})$ contains a barrier of height h, then $H(\mathcal{P}, \mathcal{Q})$ contains a barrier $(x, (a_0, b_0), \ldots, (a_t, b_t), y)$ of height at most h such that $a_i > 0$ for all $i > 0$.

In general, a barrier in $H(\mathcal{P}, \mathcal{Q})$ may be of one of the following four types (the letters "L", "R", "T", and "B" stand for "left", "right", "top", and "bottom", respectively):

- LB barrier: if its first edge connects x to a node in column 0 and its last edge connects a node in row 0 to y.
- LR barrier: if its first edge connects x to a node in column 0 and its last edge connects a node in column r to y.
- TB barrier: if its first edge connects x to a node in row s and its last edge connects a node in row 0 to y.
- TR barrier: if its first edge connects x to a node in row s and its last edge connects a node in column r to y.

The remainder of this section is devoted to proving that the minimum height of any barrier formed by two walks essentially corresponds to the maximum cost of these walks when enforcing rendezvous.

Theorem 2. *If $\mathcal{P}$ and $\mathcal{Q}$ form a barrier of height h, then $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most h.*

Proof. We prove the theorem by induction on the height of the barrier. If $h = 0$, then the barrier and thus the obstruction graph $H(\mathcal{P}, \mathcal{Q})$ necessarily contain the node $(0, 0)$. By definition this implies that $p_0 = q_0$, therefore any scheduling for $\mathcal{P}$ and $\mathcal{Q}$ results in rendezvous with cost 0.

Now, assume that the statement holds for all heights up to and including h, and consider two walks $\mathcal{P}$ and $\mathcal{Q}$ that form a barrier of height $h + 1$, for some $h \geq 0$. Fix an arbitrary scheduling $\mathcal{S}$ for $\mathcal{P}$ and $\mathcal{Q}$. Without loss of generality, let $\mathcal{S}$ be such that $(u_1, v_1) = (p_1, q_0)$. In other words, the adversary first moves the agent on the walk $\mathcal{P}$ and thus the residual walks for both agents after the first step of the scheduling are now $\mathcal{P}[1]$ and $\mathcal{Q}$. We claim that $\mathcal{P}[1]$ and $\mathcal{Q}$ form a barrier of height at most h. This implies, by

the inductive hypothesis, that any scheduling for $\mathcal{P}[1]$ and $\mathcal{Q}$ results in rendezvous with cost at most h, and therefore $\mathcal{S}$ results in rendezvous with cost at most $h + 1$.

To prove the claim, let $U_H = \{(i, j)_H : p_i = q_j\} \cup \{x_H, y_H\}$ be the node set of the obstruction graph $H = H(\mathcal{P}, \mathcal{Q})$ and let $W = (x_H, (a_0, b_0)_H, \ldots, (a_t, b_t)_H, y_H)$ be a barrier of height $h + 1$ in $H(\mathcal{P}, \mathcal{Q})$, such that $a_k > 0$ for all $k > 0$ (from Remark 2). Similarly, let $U_{H'} = \{(i, j)_{H'} : p_{i+1} = q_j\} \cup \{x_{H'}, y_{H'}\}$ be the node set of the obstruction graph $H' = H(\mathcal{P}[1], \mathcal{Q})$. Since for $i > 0$, $(i, j)_H \in U_H$ if and only if $(i - 1, j)_{H'} \in U_{H'}$, the path $W' = (x_{H'}, (a_1 - 1, b_1)_{H'}, \ldots, (a_t - 1, b_t)_{H'}, y_{H'})$ must be a barrier in H'. Furthermore, the height of W' is: $\max_{1 \leq k \leq t} (a_k - 1 + b_k) \leq \max_{0 \leq k \leq t} (a_k + b_k) - 1 \leq h$, where the last inequality follows from the fact that the height of W is at most $h + 1$. □

Before proceeding to the converse of Theorem 2, we prove the following technical lemma.

Lemma 1. *If $\mathcal{P}$ and $\mathcal{Q}[1]$ form a barrier of height at most h_1 and $\mathcal{P}[1]$ and $\mathcal{Q}$ form a barrier of height at most h_2, then $\mathcal{P}$ and $\mathcal{Q}$ form a barrier of height at most $\max(h_1, h_2) + 1$.*

Proof. Consider the obstruction graph $H(\mathcal{P}, \mathcal{Q})$. Let W_1 be a path in $H(\mathcal{P}, \mathcal{Q})$ that starts from a node in row s or in column 0, and ends at a node in row 1 or in column r, without using any node in row 0. Such a path must exist, since $\mathcal{P}$ and $\mathcal{Q}[1]$ form a barrier. Furthermore, since the height of this barrier is at most h_1, we must have $\max\{i + j : (i, j) \in W_1\} \leq h_1 + 1$.

Similarly, since $\mathcal{P}[1]$ and $\mathcal{Q}$ form a barrier, there must exist a path W_2 in $H(\mathcal{P}, \mathcal{Q})$ that starts from a node in row s or in column 1, and ends at a node in row 0 or in column r, without using any node in column 0. For this path we must have $\max\{i + j : (i, j) \in W_2\} \leq h_2 + 1$.

We consider every possible combination of types for W_1 and W_2. In each case, we construct a barrier W in $H(\mathcal{P}, \mathcal{Q})$ of height at most $\max(h_1, h_2) + 1$.

If W_1 corresponds to an LR barrier in $H(\mathcal{P}, \mathcal{Q}[1])$, then clearly W_1 itself is also an LR barrier in $H(\mathcal{P}, \mathcal{Q})$ and, by the former inequality, its height is at most $h_1 + 1$. Similarly, if W_2 corresponds to a TB barrier in $H(\mathcal{P}[1], \mathcal{Q})$, then W_2 itself is also a TB barrier in $H(\mathcal{P}, \mathcal{Q})$ and its height is at most $h_2 + 1$ by the latter inequality. If either of W_1 or W_2 corresponds to a TR barrier, then it also forms a TR barrier for $\mathcal{P}$ and $\mathcal{Q}$ and its height is at most $\max(h_1, h_2) + 1$.

Now, assume that W_1 and W_2 both correspond to LB barriers in $H(\mathcal{P}, \mathcal{Q}[1])$ and $H(\mathcal{P}[1], \mathcal{Q})$, respectively. Let $(0, a)$, $a \geq 1$, and $(1, b)$, $b \geq 0$, be the first nodes of W_1 and W_2, respectively.

We first consider the case where $a > 1$ and $b > 0$. In this case, we construct an LB barrier W in $H(\mathcal{P}, \mathcal{Q})$ as follows: follow W_1 up to the last node $(i^\star, j^\star)$ such that $i^\star = 1$ or $j^\star = b$. Then, follow as many horizontal jump edges or vertical jump edges as needed, in order to reach node $(1, b)$. Finally, follow W_2 until the end. By construction, the height of W is $\max(h_1, h_2) + 1$, since only nodes from W_1 or W_2 can be maximum-height nodes for W. The construction of W is correct in view of the following two observations:

9

(a) The node $(i^\star, j^\star)$ is well defined: Indeed, there exists at least one node (i, j) in W_1 such that $i = 1$ or $j = b$. Note that W_1 starts at row $a > 1$ and it reaches eventually row 1. The only edges that can take a walk to a lower row are regular edges, therefore W_1 contains at least one regular edge. The *first* regular edge of W_1 takes the walk from column 0 to column 1, hence to a node (i, j) with $i = 1$.

(b) It is possible to reach $(1, b)$ from $(i^\star, j^\star)$ via vertical or horizontal jump edges. Indeed, if $j^\star = b$ then $i^\star$ cannot be equal to 0, for otherwise the mere fact that $(0, b)$ and $(1, b)$ are nodes of $H(\mathcal{P}, \mathcal{Q})$ would imply that $p_0 = q_b$ and $p_1 = q_b$, therefore $p_0 = p_1$, and we do not permit our walks to stay on the same node for two consecutive steps (cf. Definition 1). We know, then, that W_1 reaches a node $(i^\star, b)$ with $i^\star \geq 1$, therefore there exists a sequence of horizontal jump edges that lead from $(i^\star, b)$ to $(1, b)$.

On the other hand, if $i^\star = 1$ then we must have $j^\star \leq b$. Indeed, if we suppose that $j^\star > b$ then there must exist a subsequent node of W_1 with $j = b$, due to the fact that only regular edges can take a walk to a lower row, therefore this contradicts the definition of $(i^\star, j^\star)$ as the *last* node with $i = 1$ or $j = b$. We know, then, that W_1 reaches a node $(1, j^\star)$ with $j^\star \leq b$, therefore there exists a sequence of vertical jump edges that lead from $(1, j^\star)$ to $(1, b)$.

If $a = 1$, i.e. W_1 corresponds to a zero-height barrier in $H(\mathcal{P}, \mathcal{Q}[1])$, then we have $a = 1$, $p_0 = q_1 = u$ (because $(0, 1)$ is a node), and $p_1 = q_b = v$ (because $(1, b)$ is a node). If, additionally, $b = 0$, i.e. W_2 also corresponds to a zero-height barrier in $H(\mathcal{P}[1], \mathcal{Q})$, then $b = 0$ and both $(0, 1)$ and $(1, 0)$ are nodes of the graph $H(\mathcal{P}, \mathcal{Q})$, thus there is a barrier of height 1 in $H(\mathcal{P}, \mathcal{Q})$. Now, assume that $b > 0$. In this case, we must actually have $b \geq 2$, because $b = 1$ would imply $p_0 = q_1 = p_1$ and a walk cannot stay on the same node in two consecutive steps (cf. Definition 1). Since u and v are visited in sequence by the walk $\mathcal{P}$, exactly one of them must correspond to the middle of some edge on the original graph where the walks are defined. We distinguish the two cases:

– If u is the middle of an edge, then either $q_0 = v$ or $q_2 = v$. If $q_0 = v$ then we have a barrier of height 1 in $H(\mathcal{P}, \mathcal{Q})$ as before. If $q_2 = v$, then $p_1 = q_2 = q_b = v$, therefore $(1, 2)$ is a node of $H(\mathcal{P}, \mathcal{Q})$ and we construct a barrier W as follows: start from node $(0, 1)$, follow a regular edge to node $(1, 2)$, then follow vertical jump edges all the way up to node $(1, b)$, $b \geq 2$, and then follow W_2 until the end.

– If v is the middle of an edge, then either $q_{b-1} = u$ or $q_{b+1} = u$ (if q_b is the last node of $\mathcal{Q}$, then W_2 is a TB barrier and we fall back to one of the previous cases). In any case, we construct a barrier W as follows: start from node $(0, 1)$, follow vertical jump edges all the way up to node $(0, b-1)$ or $(0, b+1)$, then follow a regular edge to node $(1, b)$, and then follow W_2 until the end.

Finally, the case $a > 1$, $b = 0$ can be handled symmetrically. This concludes the construction of an LB barrier in $H(\mathcal{P}, \mathcal{Q})$ when W_1 and W_2 are LB barriers in $H(\mathcal{P}, \mathcal{Q}[1])$ and $H(\mathcal{P}[1], \mathcal{Q})$, respectively.

To conclude the proof of the Lemma, we observe that in the above argument, the crucial property that we exploited was that the W_1 walk eventually reached row 1 and

that the W_2 walk started from column 1. Consequently, the same argument as above can cover all of the remaining cases where W_1 is of type sB, $s \in \{L, T\}$, and W_2 is of type Lt, $t \in \{B, R\}$, giving in each case a barrier of type st for $H(\mathcal{P}, \mathcal{Q})$. □

We are now ready to prove the converse of Theorem 2.

Theorem 3. *If $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most h, then $\mathcal{P}$ and $\mathcal{Q}$ form a barrier of height at most h.*

Proof. We prove the theorem by induction on h. The base case is immediate. Assume that the theorem holds for all values up to some $h \geq 0$ and assume that two walks $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most $h + 1$. Let us further assume that both walks are of positive length. (The proof is otherwise immediate.)

Any discrete scheduling for $\mathcal{P}$ and $\mathcal{Q}$ has cost at most $h + 1$. In particular, this is the case for all discrete schedulings for $\mathcal{P}$ and $\mathcal{Q}$ that first move the agent on the walk $\mathcal{Q}$. Therefore, any discrete scheduling for $\mathcal{P}$ and $\mathcal{Q}[1]$ has cost at most h, or, equivalently, $\mathcal{P}$ and $\mathcal{Q}[1]$ enforce rendezvous with cost at most h. By the induction hypothesis, $\mathcal{P}$ and $\mathcal{Q}[1]$ form a barrier of height at most h. By considering the discrete schedulings that first move the agent on the walk $\mathcal{P}$, we similarly prove that $\mathcal{P}[1]$ and $\mathcal{Q}$ form a barrier of height at most h. Applying Lemma 1 concludes the proof. □

Remark 3 (On the efficiency of the characterization). Given two finite walks $\mathcal{P} = (p_i)_{0 \leq i \leq r}$, $\mathcal{Q} = (q_i)_{0 \leq i \leq s}$ and an integer k, we can construct the subgraph of $H(\mathcal{P}, \mathcal{Q})$ that contains only nodes with height at most k, and check whether there is a barrier, in time $\mathcal{O}(rs)$. Therefore, in view of Theorems 2 and 3, we can decide in time $\mathcal{O}(rs)$ whether $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most k. If k is not given, we can compute the minimum k such that $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most k (or decide that they do not enforce rendezvous) by performing a binary search on k in time $\mathcal{O}(rs \log(r + s))$. Similarly, we can compute the maximum number of moves the adversary can delay rendezvous.

In the case where one or both of the walks are infinite, we can consider the prefixes $\mathcal{P}_k$, $\mathcal{Q}_k$ of length k of the walks $\mathcal{P}$ and $\mathcal{Q}$, respectively. Then, $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most k if and only if $H(\mathcal{P}_k, \mathcal{Q}_k)$ contains a barrier of height at most k, which can be decided in time $\mathcal{O}(k^2)$. If k is not given, then we can keep doubling our candidate k until we find that $H(\mathcal{P}_k, \mathcal{Q}_k)$ contains a barrier. If $\mathcal{P}$ and $\mathcal{Q}$ do not enforce rendezvous, this process never terminates. However, if $\mathcal{P}$ and $\mathcal{Q}$ do enforce rendezvous, then we can compute the minimum k such that $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most k in total time $\mathcal{O}(k^2)$.

4 Discussion

Let $\mathcal{P} = (p_i)_{0 \leq i \leq r}$ and $\mathcal{Q} = (q_i)_{0 \leq i \leq s}$ be the walks prescribed by a rendezvous algorithm for two agents in some graph G. Referring to the illustration in Figure 6, one could check whether the adversary can prevent rendezvous simply by considering the underlying grid shown in the figure and checking whether there exists a path from $(0, 0)$ to (r, s) that avoids the nodes of the obstruction graph $H(\mathcal{P}, \mathcal{Q})$ and has no downward or leftward

moves. Every such path corresponds to a scheduling that avoids rendezvous, because it stays on the "non-rendezvous" grid points and consumes both of the walks (cf. the green (thick) path in Figure 6).

Formally, let $N(\mathcal{P}, \mathcal{Q})$ be a directed graph with node set $\{(i,j) : p_i \neq q_j, 0 \leq i \leq r, 0 \leq j \leq s\}$ and edge set $\{((i,j),v) : (i < r \wedge v = (i+1,j)) \vee (j < s \wedge v = (i,j+1))\}$. One can verify that the adversary can prevent rendezvous for $\mathcal{P}$ and $\mathcal{Q}$ for at least k moves if and only if there exists a directed path of length at least k in $N(\mathcal{P}, \mathcal{Q})$ starting from $(0,0)$, and this condition can be checked in time polynomial in the length of the two walks.

However, in view of the characterization that we obtained in the previous section, a different way to check whether $\mathcal{P}$ and $\mathcal{Q}$ enforce rendezvous with cost at most k is by checking whether there exists a barrier of height at most k in the corresponding obstruction graph $H(\mathcal{P}, \mathcal{Q})$. This property can also be checked in polynomial time. However, we argue that the certificate of enforcement of rendezvous obtained in this manner, i.e. the barrier in the obstruction graph, is much more informative than the non-existence of a certain directed path in $N(\mathcal{P}, \mathcal{Q})$.

Indeed, the barrier reveals a lot of information explaining *why* the two walks enforce rendezvous. Consider for example the case of a straightline LB barrier that consists of regular edges only. This barrier tells us that there is something special about the beginning of the two walks, which enforces rendezvous. A closer examination reveals that such a barrier implies that there is a prefix of $\mathcal{P}$ and a prefix of $\mathcal{Q}$ that are exact reverses of each other.

It is intuitively clear and has been observed by Czyzowicz et al. [13] that, if $\mathcal{P}$ contains a prefix U and at the same time $\mathcal{Q}$ contains a prefix that is the reverse of U, then rendezvous of the two agents under this pair of walks is actually enforced, regardless of the actions of the adversary. The situation resembles two trains entering a tunnel in opposite directions. No matter how their speed varies, the trains eventually cross each other. This notion of *tunnel* was exploited by Czyzowicz et al. [13] and by Guilbault and Pelc [22]. In both papers, it is guaranteed that, for any starting positions of the agents, their walks eventually create a tunnel, i.e., a prefix of one is the reverse of a prefix of the other, and thus rendezvous is achieved. In fact, it is equally easy to see that the two prefixes need not be exact reverses of each other: the agents meet, even if one or both of them perform arbitrary zigzags (moving back and forth *inside* the tunnel) that are not exactly mirrored in the movement of the other agent. This more general notion of tunnel is implicitly used in various algorithms [5,11,16,32].

Taking this one step further, one could allow the trains in the tunnel to instantaneously *jump backward* arbitrarily long, each in the direction of its respective starting point. Even with this new ability, it should still be intuitively clear that two trains that start from opposite endpoints of a tunnel have to cross each other. In the agent setting, jumping corresponds to the tunnel itself being a non-simple path on the graph, therefore enabling an agent to instantaneously "jump" from one occurrence of a particular node to a different occurrence of the same node, at any position in the tunnel. Such a jump is backward, resp. forward, if the latter occurrence is closer, resp. further, to the starting

point of the agent than the former occurrence. Unfortunately, it turns out that rendezvous may be enforced even though there does not exist any tunnel such that agents are only allowed zigzags and backward jumps (see Fig. 7 and 8 for an example). Therefore, even a limited capacity of *forward* jumping can be allowed, although it is a non-trivial task to characterize exactly the forward jumps that can be allowed so that rendezvous is still enforced.

One naturally wonders if that's all there is to asynchronous rendezvous. Are we forced to design algorithms that create some kind of tunnel that is traversed by the agents in opposite directions (allowing for zigzags and jumps obeying certain rules)? Or could we employ a fundamentally different idea in order to ensure that the agents meet despite best efforts on the part of the adversary? It seems that a barrier in the obstruction graph $H(\mathcal{P}, \mathcal{Q})$ contains all the information that would enable us to recover the particular path on the graph that serves as a "tunnel" for $\mathcal{P}$ and $\mathcal{Q}$. If this is true, then the former question would be answered affirmatively. We therefore leave the following as an open question and an interesting direction for further research: Given a barrier, how do we recover a path on the graph that the agents want to traverse in opposite directions (allowing for zigzags and jumps)? Also, what are the correct rules for forward jumps?
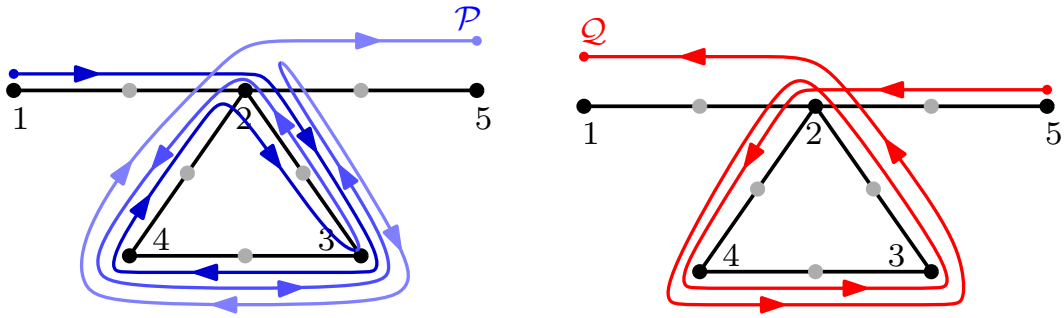
Acknowledgment
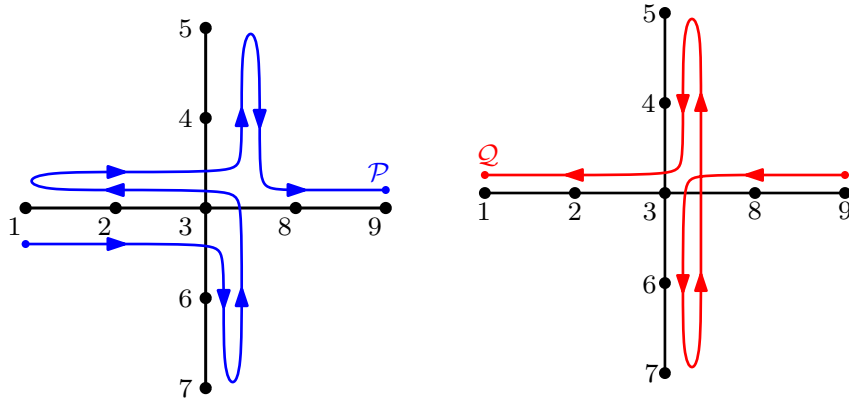
Fig. 5. Two walks on the same graph: $\mathcal{P} = (1, 2, 3, 6, 7, 6, 3, 2, 1, 2, 3, 4, 5, 4, 3, 8, 9)$ and $\mathcal{Q} = (9, 8, 3, 6, 7, 6, 3, 4, 5, 4, 3, 2, 1)$.
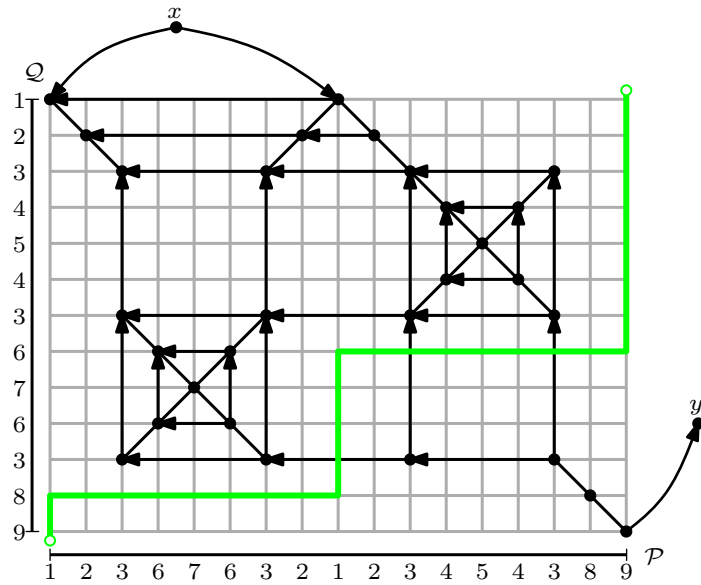
Fig. 6. The obstruction diagram corresponding to the walks in Figure 5. For simplicity, bidirectional regular edges are illustrated without arrows. The green (thick) path corresponds to a scheduling for $\mathcal{P}$ and $\mathcal{Q}$ that avoids rendezvous.

14

Fig. 7. Two walks on the same graph: $\mathcal{P} = (1, 2, 3, 4, 2, 3, 2, 4, 3, 2, 3, 4, 2, 5)$ and $\mathcal{Q} = (5, 2, 4, 3, 2, 4, 3, 2, 1)$.
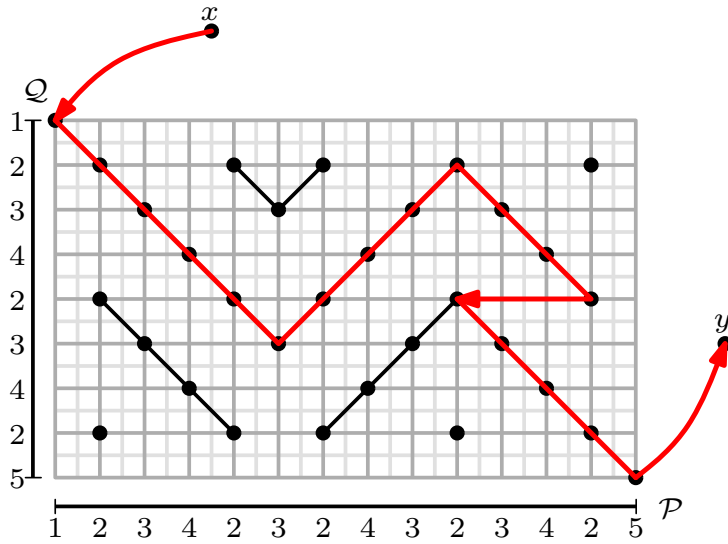
Fig. 8. The simplified obstruction diagram corresponding to the walks in Figure 7. The red (thick) $x \rightsquigarrow y$ path corresponds to a minimum-height barrier for $\mathcal{P}$ and $\mathcal{Q}$ that proves rendezvous. For clarity, jump edges are not illustrated, except the one used in the barrier.

References

1. Agmon, N., Peleg, D.: Fault-tolerant gathering algorithms for autonomous mobile robots. SIAM J. Comput. 36(1), 56–82 (2006)
2. Alpern, S., Gal, S.: The theory of search games and rendezvous, International Series in Operations Research & Management Science, vol. 55. Kluwer Academic Publishers, Dordrecht (2003)
3. Ando, H., Oasa, Y., Suzuki, I., Yamashita, M.: Distributed memoryless point convergence algorithm for mobile robots with limited visibility. IEEE Trans. Robot. Autom. 15(5), 818–828 (oct 1999)
4. Balabonski, T., Delga, A., Rieg, L., Tixeuil, S., Urbain, X.: Synchronous gathering without multiplicity detection: A certified algorithm. In: Bonakdarpour, B., Petit, F. (eds.) Stabilization, Safety, and Security of Distributed Systems - 18th International Symposium, SSS 2016, Lyon, France, November 7-10, 2016, Proceedings. Lecture Notes in Computer Science, vol. 10083, pp. 7–19 (2016), `https://doi.org/10.1007/978-3-319-49259-9_2`
5. Bampas, E., Czyzowicz, J., Gąsieniec, L., Ilcinkas, D., Labourel, A.: Almost optimal asynchronous rendezvous in infinite multidimensional grids. In: Lynch, N.A., Shvartsman, A.A. (eds.) DISC. LNCS, vol. 6343, pp. 297–311. Springer (2010)
6. Bonnet, F., Potop-Butucaru, M., Tixeuil, S.: Asynchronous gathering in rings with 4 robots. In: Mitton, N., Loscrì, V., Mouradian, A. (eds.) Ad-hoc, Mobile, and Wireless Networks - 15th International Conference, ADHOC-NOW 2016, Lille, France, July 4-6, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9724, pp. 311–324. Springer (2016), `https://doi.org/10.1007/978-3-319-40509-4_22`
7. Bouzid, Z., Das, S., Tixeuil, S.: Gathering of mobile robots tolerating multiple crash faults. In: IEEE 33rd International Conference on Distributed Computing Systems, ICDCS 2013, 8-11 July, 2013, Philadelphia, Pennsylvania, USA. pp. 337–346. IEEE Computer Society (2013), `https://doi.org/10.1109/ICDCS.2013.27`
8. Bramas, Q., Tixeuil, S.: Wait-free gathering without chirality. In: Scheideler, C. (ed.) Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings. Lecture Notes in Computer Science, vol. 9439, pp. 313–327. Springer (2015), `https://doi.org/10.1007/978-3-319-25258-2_22`
9. Cieliebak, M., Flocchini, P., Prencipe, G., Santoro, N.: Distributed computing by mobile robots: Gathering. SIAM J. Comput. 41(4), 829–879 (2012)
10. Cohen, R., Peleg, D.: Convergence properties of the gravitational algorithm in asynchronous robot systems. SIAM J. Comput. 34(6), 1516–1528 (2005)
11. Collins, A., Czyzowicz, J., Gąsieniec, L., Labourel, A.: Tell me where I am so I can meet you sooner (asynchronous rendezvous with location information). In: et al. Cyril Gavoille, S.A., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP (2). LNCS, vol. 6199, pp. 502–514. Springer (2010)
12. Courtieu, P., Rieg, L., Tixeuil, S., Urbain, X.: Certified universal gathering in $\mathbb{R}^2$ for oblivious mobile robots. In: Gavoille, C., Ilcinkas, D. (eds.) Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9888, pp. 187–200. Springer (2016), `https://doi.org/10.1007/978-3-662-53426-7_14`
13. Czyzowicz, J., Pelc, A., Labourel, A.: How to meet asynchronously (almost) everywhere. ACM Transactions on Algorithms 8(4), 37 (2012)
14. D'Angelo, G., Di Stefano, G., Klasing, R., Navarra, A.: Gathering of robots on anonymous grids and trees without multiplicity detection. Theor. Comput. Sci. 610, 158–168 (2016), `https://doi.org/10.1016/j.tcs.2014.06.045`
15. D'Angelo, G., Di Stefano, G., Navarra, A.: How to gather asynchronous oblivious robots on anonymous rings. In: DISC. pp. 326–340 (2012)
16. De Marco, G., Gargano, L., Kranakis, E., Krizanc, D., Pelc, A., Vaccaro, U.: Asynchronous deterministic rendezvous in graphs. Theor. Comput. Sci. 355(3), 315–326 (2006)
17. Défago, X., Gradinariu, M., Messika, S., Parvédy, P.R.: Fault-tolerant and self-stabilizing mobile robots gathering. In: DISC. pp. 46–60 (2006)
18. Dieudonné, Y., Pelc, A.: Anonymous meeting in networks. Algorithmica 74(2), 908–946 (2016), `https://doi.org/10.1007/s00453-015-9982-0`

19. Dieudonné, Y., Pelc, A., Villain, V.: How to meet asynchronously at polynomial cost. SIAM J. Comput. 44(3), 844–867 (2015), `https://doi.org/10.1137/130931990`
20. Flocchini, P., Prencipe, G., Santoro, N.: Distributed Computing by Oblivious Mobile Robots. Morgan and Claypool Publishers (2012)
21. Flocchini, P., Prencipe, G., Santoro, N., Widmayer, P.: Gathering of asynchronous robots with limited visibility. Theor. Comput. Sci. 337(1-3), 147–168 (2005)
22. Guilbault, S., Pelc, A.: Asynchronous rendezvous of anonymous agents in arbitrary graphs. In: Anta, A.F., Lipari, G., Roy, M. (eds.) OPODIS. LNCS, vol. 7109, pp. 421–434. Springer (2011)
23. Honorat, A., Potop-Butucaru, M., Tixeuil, S.: Gathering fat mobile robots with slim omnidirectional cameras. Theor. Comput. Sci. 557, 1–27 (2014), `https://doi.org/10.1016/j.tcs.2014.08.004`
24. Kamei, S., Lamani, A., Ooshita, F., Tixeuil, S.: Asynchronous mobile robot gathering from symmetric configurations without global multiplicity detection. In: Kosowski, A., Yamashita, M. (eds.) Structural Information and Communication Complexity - 18th International Colloquium, SIROCCO 2011, Gdansk, Poland, June 26-29, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6796, pp. 150–161. Springer (2011), `https://doi.org/10.1007/978-3-642-22212-2_14`
25. Kamei, S., Lamani, A., Ooshita, F., Tixeuil, S.: Gathering an even number of robots in an odd ring without global multiplicity detection. In: Rovan, B., Sassone, V., Widmayer, P. (eds.) Mathematical Foundations of Computer Science 2012 - 37th International Symposium, MFCS 2012, Bratislava, Slovakia, August 27-31, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7464, pp. 542–553. Springer (2012), `https://doi.org/10.1007/978-3-642-32589-2_48`
26. Klasing, R., Kosowski, A., Navarra, A.: Taking advantage of symmetries: Gathering of many asynchronous oblivious robots on a ring. Theor. Comput. Sci. 411(34-36), 3235–3246 (2010)
27. Klasing, R., Markou, E., Pelc, A.: Gathering asynchronous oblivious mobile robots in a ring. Theor. Comput. Sci. 390(1), 27–39 (2008)
28. Kranakis, E., Krizanc, D., Markou, E.: The mobile agent rendezvous problem in the ring, Synthesis Lectures on Distributed Computing Theory, vol. 1. Morgan & Claypool (2010)
29. Kranakis, E., Krizanc, D., Rajsbaum, S.: Mobile agent rendezvous: A survey. In: Flocchini, P., Gąsieniec, L. (eds.) SIROCCO. LNCS, vol. 4056, pp. 1–9. Springer (2006)
30. Pelc, A.: Deterministic rendezvous in networks: A comprehensive survey. Networks 59(3), 331–347 (2012)
31. Prencipe, G.: Impossibility of gathering by a set of autonomous mobile robots. Theor. Comput. Sci. 384(2-3), 222–231 (2007)
32. Stachowiak, G.: Asynchronous deterministic rendezvous on the line. In: Nielsen, M., Kucera, A., Miltersen, P.B., Palamidessi, C., Tuma, P., Valencia, F.D. (eds.) SOFSEM. LNCS, vol. 5404, pp. 497–508. Springer (2009)