Asynchronous deterministic rendezvous in bounded terrains

Jurek Czyzowicz[1,⋆], David Ilcinkas[2,⋆⋆],
Arnaud Labourel[2,⋆⋆,⋆⋆⋆], and Andrzej Pelc[1,†]

[1] Département d'informatique, Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada. E-mails: jurek@uqo.ca, pelc@uqo.ca
[2] LaBRI, CNRS & Université de Bordeaux, 33405 Talence, France. E-mails: david.ilcinkas@labri.fr, labourel.arnaud@gmail.com,

Abstract. Two mobile agents (robots) have to meet in an a priori unknown bounded terrain modeled as a polygon, possibly with polygonal obstacles. Robots are modeled as points, and each of them is equipped with a compass. Compasses of robots may be incoherent. Robots construct their routes, but the actual walk of each robot is decided by the adversary that may, e.g., speed up or slow down the robot. We consider several scenarios, depending on three factors: (1) obstacles in the terrain are present, or not, (2) compasses of both robots agree, or not, (3) robots have or do not have a map of the terrain with their positions marked. The cost of a rendezvous algorithm is the worst-case sum of lengths of the robots' trajectories until their meeting. For each scenario we design a deterministic rendezvous algorithm and analyze its cost. We also prove lower bounds on the cost of any deterministic rendezvous algorithm in each case. For all scenarios these bounds are tight.

keywords: mobile agent, rendezvous, deterministic, polygon, obstacle

1 Introduction

The problem and the model. Two mobile agents (robots) modeled as points starting at different locations of an a priori unknown bounded terrain have to meet. The terrain is represented as a polygon possibly with a finite number of polygonal obstacles. We assume that the boundary of the terrain is included in it. Thus, formally, a terrain is a set $\mathcal{P}_0 \setminus (\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k)$, where $\mathcal{P}_0$ is a closed polygon and $\mathcal{P}_1, \ldots, \mathcal{P}_k$ are disjoint open polygons included in $\mathcal{P}_0$. We assume that a robot knows if it is at an interior or at a boundary point, and in the

[⋆] Partially supported by NSERC discovery grant.

[⋆⋆] Partially supported by the ANR project ALADDIN, the INRIA project CEPAGE and by a France-Israel cooperation grant (Multi-Computing project).

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

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

latter case it is capable of walking along the boundary in both directions (i.e., it knows the slope(s) of the boundary at this point). However, a robot cannot sense the terrain or the other robot at any vicinity of its current location. Meeting (rendezvous) is defined as the equality of points representing robots at some moment of time.

We assume that each robot has a unit of length (not necessarily the same for the two agents) and a compass. Compasses of robots may be incoherent, however we assume that robots have the same (clockwise) orientation of their system of coordinates. An additional tool, which may or may not be available to the robots, is a map of the terrain. The map available to a robot is scaled (i.e., it accurately shows the distances), distinguishes the starting positions of this robot and the other one, and is oriented according to the compass of the robot. (Hence maps of different robots may have different North.)

All our considerations concern deterministic algorithms. The crucial notion is the *route* of the robot which is a finite polygonal path in the terrain. The adversary initially places a robot at some point in the terrain. The robot constructs its route in steps in the following way. In every step, the robot starts at some point v; in the first step, v is the starting point chosen by the adversary. The robot chooses a direction α, according to its compass, and a distance x. If the segment of length x in direction α starting in v does not intersect the boundary of the terrain, the step ends when the robot reaches point u at distance x from v in direction α. Otherwise, the step ends at the closest point of the boundary in direction α. If the starting point v in a step is in a segment of the boundary of the terrain, the robot has also an option (in this step) to follow this segment of the boundary in any of the two directions until its end or for some given distance along it. Steps are repeated until rendezvous, or until the route of the robot is completed.

We consider the *asynchronous* version of the rendezvous problem. The asynchrony of the robots' movements is captured by the assumption that the actual walk of each robot is decided by the adversary: the movement of the robot can be at arbitrary speed, the robot may sometimes stop or go back and forth, as long as the walk of the robot in each segment of its route is continuous, does not leave it and covers all of it.[1] More formally, the route in a terrain is a sequence $(S_1, S_2, \ldots, S_k)$ of segments, where $S_i = [a_i, a_{i+1}]$ is the segment corresponding to step i. In our algorithms the route is always finite. This means that the robot stops at some point, regardless of the moves of the other robot. We now describe the walk f of a robot on its route. Let $R = (S_1, S_2, \ldots, S_k)$ be the route of a

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

robot. Let $(t_1, t_2, \ldots, t_{k+1})$, where $t_1 = 0$, be an increasing sequence of reals, chosen by the adversary, that represent points in time. Let $f_i : [t_i, t_{i+1}] \rightarrow [a_i, a_{i+1}]$ be any continuous function, chosen by the adversary, such that $f_i(t_i) = a_i$ and $f_i(t_{i+1}) = a_{i+1}$. For any $t \in [t_i, t_{i+1}]$, we define $f(t) = f_i(t)$. The interpretation of the walk f is as follows: at time t the robot is at the point $f(t)$ of its route and after time t_{k+1} the robot remains inert. This general definition of the walk and the fact that it is constructed by the adversary capture the asynchronous characteristics of the process. Throughout the paper, *rendezvous* means deterministic asynchronous rendezvous.

Robots with routes R and R' and with walks f and f' meet at time t, if points $f(t)$ and $f'(t)$ are equal. A rendezvous is guaranteed for routes R and R', if the robots using these routes meet at some time t, regardless of the walks chosen by the adversary. The trajectory of a robot is the sequence of segments on its route until rendezvous. (The last segment of the trajectory of a robot may be either the last segment of its route or any of its segments or a portion of it, if the other robot is met there.) The cost of a rendezvous algorithm is the worst case sum of lengths of segments of trajectories of both robots, where the worst case is taken over all terrains with the considered values of parameters, and all adversarial decisions.

We consider several scenarios, depending on three factors: (1) obstacles in the terrain are present, or not, (2) compasses of both robots agree, or not, (3) robots have or do not have a map of the terrain. Combinations of the presence or absence of these factors give rise to eight scenarios. For each scenario we design a deterministic rendezvous algorithm and analyze its cost. We also prove lower bounds on the cost of any deterministic rendezvous algorithm in each case. For all scenarios these bounds are tight.

One final clarification has to be made. For all scenarios except those with incoherent compasses and the presence of obstacles (regardless of the availability of a map), robots may be anonymous, i.e., they execute identical algorithms. By contrast, with the presence of obstacles and incoherent compasses, anonymity would preclude feasibility of rendezvous in some situations. Consider a square with one square obstacle positioned at its center. Consider two robots starting at opposite (diagonal) corners of the larger square, with compasses pointing to opposite North directions. If they execute identical algorithms and walk at the same speed, then at each time they are in symmetric positions in the terrain and hence rendezvous is impossible. The only way to break symmetry for a deterministic rendezvous in this case is to equip the robots with distinct labels (which are positive integers). Hence, this is the assumption we make for the scenarios with the presence of obstacles and incoherent compasses (both with and without a map). For any label μ, we denote by $|\mu|$ the length of the binary representation of the label, i.e., $|\mu| = \lfloor \log \mu \rfloor + 1$.

Our results. The cost of our algorithms depends on some of the following parameters (different parameters for different scenarios, see the discussion in Section 4): D is the distance between starting positions of robots in the terrain (i.e., the length of a shortest path between them included in the terrain), P is the perimeter of the terrain, (i.e., the sum of perimeters of all polygons $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_k$), x is the largest perimeter of an obstacle, and l and L are the smaller and larger labels of the two robots, respectively, for the two scenarios that require different labels, as remarked above., i.e., for the scenarios with the presence of obstacles and incoherent compasses.

Our rendezvous algorithms rely on two different ideas: either meeting in a uniquely defined point of the terrain, or meeting on a uniquely defined cycle. It turns out that a uniquely defined point can be found in all scenarios except those with the presence of obstacles and incoherent compasses. Apart from this exception even anonymous robots can meet. On the other hand, with the presence of obstacles and incoherent compasses, such a uniquely defined point may not exist, as witnessed by the above quoted example of a square with one square obstacle positioned at its center. For these scenarios we resort to the technique of meeting at a common cycle, breaking symmetry by different labels of robots.

We first summarize our results concerning rendezvous when each of the robots is equipped with a map showing its own position and that of the other robot. If compasses of the robots are coherent, then we show a rendezvous algorithm at cost D, which is clearly optimal. Otherwise, and if the terrain does not contain obstacles, then we show an algorithm whose cost is again D, and hence optimal. Finally, with incoherent compasses in the presence of obstacles, we show a rendezvous algorithm at cost $O(D|l|)$; in the latter scenario we show that cost $\Omega(D|l|)$ is necessary for some terrains.

Our results concerning rendezvous without a map are as follows. If compasses of the robots are coherent, then we show a rendezvous algorithm at cost $O(P)$. We also show a matching lower bound $\Omega(P)$ in this case. If compasses of the robots are incoherent, but the terrain does not contain obstacles, then we show a rendezvous algorithm at cost $O(P)$ and again a matching lower bound $\Omega(P)$. Finally, in the hardest of all scenarios (presence of obstacles, incoherent compasses and no map) we have a rendezvous algorithm at cost $O(P + x|L|)$ and a matching lower bound $\Omega(P + x|L|)$. Table 1 summarizes our results. Due to lack of space, some proofs are removed.

Related work. The rendezvous problem was first described in [24]. A detailed discussion of the large literature on rendezvous can be found in the excellent book [4]. Most of the results in this domain can be divided into two classes: those

Rendezvous with a map			Rendezvous without a map		
compasses obstacles	coherent	incoherent	compasses obstacles	coherent	incoherent
no	D	D	no	$\Theta(P)$	$\Theta(P)$
yes		$\Theta(D\vert l\vert)$	yes		$\Theta(P+x\vert L\vert)$

Table 1. Summary of results

considering the geometric scenario (rendezvous in the line, see, e.g., [16, 25], or in the plane, see, e.g., [7, 8]), and those discussing rendezvous in graphs, e.g., [2, 5]. Some of the authors, e.g., [2, 3, 6] consider the probabilistic scenario where inputs and/or rendezvous strategies are random. Randomized rendezvous strategies use random walks in graphs, which were thoroughly investigated and applied also to other problems, such as, e.g., graph traversing [1]. A generalization of the rendezvous problem is that of gathering [15, 18, 19], when more than two robots have to meet in one location.

If graphs are unlabeled, deterministic rendezvous requires breaking symmetry, which can be accomplished either by allowing marking nodes or by labeling the robots. Deterministic rendezvous with anonymous robots working in unlabeled graphs but equipped with tokens used to mark nodes was considered e.g., in [21]. In [26] the authors studied the task of gathering many robots with unique labels. In [14, 20, 27] deterministic rendezvous in graphs with labeled robots was considered. However, in all the above papers, the synchronous setting was assumed. Asynchronous gathering under geometric scenarios has been studied, e.g., in [11, 15, 22] in different models than ours: robots could not remember past events, but they were assumed to have at least partial visibility of the scene. The first paper to consider deterministic asynchronous rendezvous in graphs was [12]. The authors concentrated on complexity of rendezvous in simple graphs, such as the ring and the infinite line. They also showed feasibility of deterministic asynchronous rendezvous in arbitrary finite connected graphs with *known* upper bound on the size. Further improvements of the above results for the infinite line were proposed in [25]. Gathering many robots in a graph, under a different asynchronous model and assuming that the whole graph is seen by each robot, has been studied in [18, 19].

2 Rendezvous with a map

We start by describing the following procedure that finds a unique shortest path from the starting position of one robot to the other. The procedure works in all scenarios in which robots have a map of the terrain with their positions indicated.

```
Procedure path UniquePath(point $v$, point $w$)
    point $u := v$; path $p := \{v\}$;
    $\mathcal{S} = \{p_s \mid p_s$ is a shortest path between $v$ and $w\}$;
    while ($u \neq w$) do
        $\mathcal{U}$ :=all paths $p_s$ of $\mathcal{S}$ such that the first segment of the subpath of $p_s$
            leading from $u$ to $w$ is the first in clockwise order around $u$
            starting from the direction $vw$;
        $p' := \bigcap_{p_s \in \mathcal{U}} p_s$;
        extend $p$ with the connected part of $p'$ containing $u$;
        $u :=$ new end of path $p$;
    return $p$;
```

Lemma 1. *Procedure* `UniquePath` *computes a unique shortest path from v to w, independent of the robot computing it.*

2.1 Coherent compasses

If robots have a map and coherent compasses, then they can easily agree on one of their two starting positions and meet at this point at cost D, which is optimal. This is done by the following Algorithm RVCM (rendezvous with a map and coherent compasses).

Algorithm *RVCM*
Let v be the northernmost of the two starting positions of the robots. If both robots have the same latitude, let v be the easternmost of them. Let w be the other starting position. The robot starting at v remains inert. The robot starting at w computes the path $p =$ `UniquePath`(w, v) and moves along p until v.

Theorem 1. *Algorithm RVCM guarantees rendezvous at cost D, for any two robots with a map and coherent compasses, in any terrain.*

2.2 Incoherent compasses

Terrains without obstacles.
In an empty polygon there is a unique shortest path between starting positions of the robots [9], and robots with a map can meet in the middle of this path at cost D, which is optimal. This is done by Algorithm RVM (rendezvous with a map, without obstacles).

Algorithm *RVM*
The robot computes the (unique) shortest path between the starting positions of the two robots. Then, it moves along this shortest path until the middle of it.

Theorem 2. *Algorithm RVM guarantees rendezvous at cost D for any two robots with a map, in any terrain without obstacles.*

Terrains with obstacles.
This is the first of the two scenarios where robots cannot always predetermine a meeting point. Therefore they compute a common embedding of a ring on which they are initially situated, and then each robot executes the rendezvous procedure from [12] for this ring. For the sake of completeness, this procedure is briefly described below. It consists of two parts: Label Transformation and Label Execution. The Label Transformation part takes the label μ of an agent and produces the label μ^* in the following way. First produce label μ' consisting of a string of $|\mu|$ zeros, followed by a 1 and then followed by the string μ. The label μ^*, called the *transformed label* of the agent, is obtained by replacing in μ' each 0 by 01 and each 1 by 10. The Label Execution part is divided into phases numbered 1,2,... For a given agent, we define the execution of bit 0 (resp. 1) in phase a as performing 3^a steps left (resp. right), according to the agent's local orientation. For an agent with label μ, phase a consists of consecutive executions of all bits of μ^* from left to right.

Using the above procedure, rendezvous with a map, with obstacles is performed by the following Algorithm RVMO. Recall that in this scenario robots have distinct labels, hence the procedure from [12] can be applied. Rendezvous is guaranteed to occur on the ring, but the meeting point depends on the walks of the robots determined by the adversary.

Algorithm $RVMO$
Phase 1: computation of the embedding[1] $\mathcal{R}$ of a ring of size 4.
Let v be the starting position of the robot and let w be the starting position of the other robot. The robot computes the embedding $\mathcal{R}$ of a ring, composed of four nodes v, a, w and b, where a is the midpoint of `UniquePath`(v, w), b is the midpoint of `UniquePath`(w, v), and the four edges are the respective halves of these paths.
Phase 2: rendezvous on $\mathcal{R}$.
This phase consists in applying the above described rendezvous procedure from [12] for ring $\mathcal{R}$, whose size (four) is known to the robots.

[a] This embedding is not necessarily homeomorphic with a circle, it may be degenerate.

Theorem 3. *Algorithm $RVMO$ guarantees rendezvous at cost $O(D|l|)$ for arbitrary two robots with a map, in any terrain.*

The following lower bound shows that the cost of Algorithm RVMO cannot be improved for some terrains. Indeed, it implies that for all $D > 0$, there exists a polygon with a single obstacle, for which the cost of any rendezvous algorithm for two robots, starting at distance D, is $\Omega(D|l|)$.

Theorem 4. *For any rendezvous algorithm A, for any $D > 0$, and for any integers $k_2 \geq k_1 > 0$, there exist two labels l_1 and l_2 of lengths at most k_1 and at most k_2, respectively, and a polygon with a single obstacle of perimeter $2D$, such that algorithm A executed by robots with labels l_1 and l_2 starting at distance D, requires cost $\Omega(Dk_1)$. This holds even if the two robots have a map.*

3 Rendezvous without a map

3.1 Coherent compasses

It turns out that robots can recognize the outer boundary of the terrain even without a map. Hence, if their compasses are coherent, they can identify a uniquely defined point on this boundary and meet in this point. This is done by Algorithm RVC (rendezvous with coherent compasses) at cost $O(P)$.

Algorithm *RVC*
From its starting position v, the robot follows the half-line α pointing to the North, as far as possible. When it hits the boundary of a polygon $\mathcal{P}$ (i.e., either the external boundary of the terrain or the boundary of an obstacle), it traverses the entire boundary of $\mathcal{P}$. Then, it computes the point u which is the farthest point from v in $\mathcal{P} \cap \alpha$. It goes around $\mathcal{P}$ until reaching u again and progresses on α, if possible. If this is impossible, the robot recognizes that it went around the boundary of $\mathcal{P}_0$. It then computes the northernmost points in $\mathcal{P}_0$. Finally, it traverses the boundary of $\mathcal{P}_0$ until reaching the easternmost of these points.

Theorem 5. *Algorithm RVC guarantees rendezvous at cost $O(P)$ for any two robots with coherent compasses, in any terrain.*

The following lower bound shows that the cost of Algorithm RVC is asymptotically optimal, for some polygons even without obstacles. This lower bound $\Omega(P)$ holds even if the distance D between starting positions of robots is bounded and if their compasses are coherent.

Theorem 6. *There exists a polygon of an arbitrarily large perimeter P, for which the cost of any rendezvous algorithm for two robots with coherent compasses starting at any distance $D > 0$, is $\Omega(P)$.*

Proof. Consider the polygon $\mathcal{P}'$ obtained by attaching to each side of a regular k-gon, whose center is at distance $D/8$ from its boundary, a rectangle of length

$3D/8$ and of height equal to the side length of the k-gon. The polygon $\mathcal{P}$ is the polygon obtained by gluing two copies of $\mathcal{P}'$ by the small side of one of the rectangles, as depicted in Fig. 1. Let P be the perimeter of the polygon $\mathcal{P}$. We choose $k = \Theta(P/D)$. There are two types of rectangles in $\mathcal{P}$, two *passing* ones (they share one side) and the $2k-2$ *normal* ones.

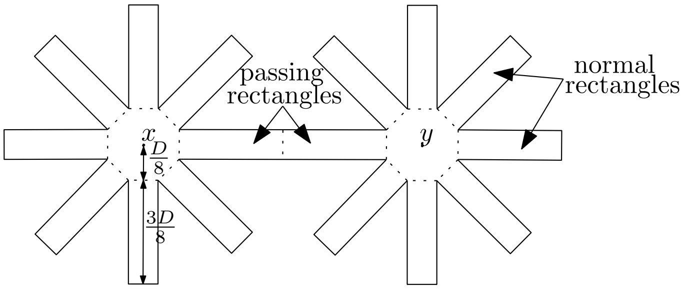

Fig. 1. Polygon $\mathcal{P}$

Consider all rotations of the polygon $\mathcal{P}$ around its center of symmetry by angles $2\pi i/k$, for $i = 0, \ldots, k-1$. We will prove that any deterministic rendezvous algorithm requires cost $\Omega(P)$ in at least one of the rotated polygons. Each robot starts in the center of a different k-gon. We say that a robot has *penetrated* a rectangle if it has moved at distance $D/8$ inside the rectangle. In order to accomplish rendezvous, at least one robot has to penetrate a passing rectangle. Each time one robot penetrates a rectangle, the adversary chooses a rotation, so that all previously penetrated rectangles, including the current one, are normal rectangles. This choice is coherent with the knowledge previously acquired by the robots, since normal rectangles are undistinguishable from each other and a robot needs to penetrate a rectangle in order to distinguish its type. Hence, the two robots have to penetrate a total of $k-1$ rectangles before the adversary cannot rotate the figure to prevent the penetration of a passing rectangle. It follows that at least one of the robots has to traverse a total distance of $\Omega(kD) = \Omega(P)$ before meeting. □

3.2 Incoherent compasses

Terrains without obstacles.
In this section, we use the notion of medial axis, proposed by Blum [10], to define a unique point of rendezvous inside the terrain. Observe that we cannot use the centroid for the rendezvous point since, as we also consider non-convex terrains, the centroid is not necessarily inside the terrain. The *medial axis* $M(\mathcal{P})$

of a polygon $\mathcal{P}$ is defined as the set of points inside $\mathcal{P}$ which have more than one closest point on the boundary of $\mathcal{P}$. Actually, $M(\mathcal{P})$ is a planar tree contained in $\mathcal{P}$, in which nodes are linked by either straight-line segment or arcs of parabolas [23]. We define the *medial point* of a polygon $\mathcal{P}$ as either the central node of $M(\mathcal{P})$ or the middle of the central edge of $M(\mathcal{P})$, depending on whether $M(\mathcal{P})$ has a central node or a central edge. Remark that the medial point of $\mathcal{P}$ is unique and is inside $\mathcal{P}$. The medial axis of a polygon $\mathcal{P}$ can be computed as in [13]. Algorithm RV (rendezvous without obstacles, without a map and with possibly incoherent compasses) determines the unknown (empty) polygon and guarantees meeting in its medial point.

Algorithm RV
At its starting position, the robot chooses an arbitrary half-line α which it follows until it hits the boundary of the polygon $\mathcal{P}_0$. It traverses the entire boundary of $\mathcal{P}_0$ and computes the medial point v of $\mathcal{P}_0$. Then, it moves to v by a shortest path and stops.

Theorem 7. *Algorithm RV guarantees rendezvous at cost $O(P)$ for any two robots, in any terrain without obstacles.*

The lower bound from Theorem 6 shows that the cost of Algorithm RV cannot be improved for some polygons.

Terrains with obstacles.
Our last rendezvous algorithm, Algorithm RVO, works for the hardest of all scenarios: rendezvous with obstacles, no map, and possibly incoherent compasses. Here again it may be impossible to predetermine a meeting point. Thus robots identify a common cycle and meet on this cycle. The difference between the present setting and that of Algorithm $RVMO$, where a map was available, is that now robots may start outside of the common cycle and have to reach it before attempting rendezvous on it. (Hence, in particular, the robots cannot use directly the procedure for rendezvous in a ring from [12], as was done in Algorithm $RVMO$.) Also the common cycle is different: rather than being composed of two shortest paths between initial positions of the robots (a map seems to be needed to find such paths), it is the boundary of a (possible) obstacle $\mathcal{O}$ in which the medial point of the outer polygon is hidden. These changes have consequences for the cost of the algorithm. The fact that the medial point of the outer polygon has to be found and the obstacle $\mathcal{O}$ has to be reached is responsible for the summand P in the cost. The only bound on the perimeter of this obstacle is x. Finally, the fact that the adversary may delay the robot with the smaller label and force the other robot to make its tours of obstacle $\mathcal{O}$ before the robot with the smaller label even reaches the obstacle, is responsible for the summand $x|L|$, rather than $x|l|$, in the cost.

A *cycle* is a polygonal path whose both extremities are the same point. A *tour* of a cycle $\mathcal{C}$ is any sequence of all the segments of $\mathcal{C}$ in either clockwise or counterclockwise order starting from a vertex of $\mathcal{C}$. By extension, a *partial tour* of $\mathcal{C}$ is a path which is a subsequence of a tour of $\mathcal{C}$ with the first or the last segment of the subsequence possibly replaced by a subsegment of it.

Algorithm RVO

Phase 1: Computation of the medial point of $\mathcal{P}_0$

At its starting position z, the robot chooses an arbitrary half-line α which it follows as far as possible. When it hits the boundary of a polygon $\mathcal{P}$, it traverses the entire boundary of $\mathcal{P}$. Then, it computes the point w which is the farthest point from z in $\mathcal{P} \cap \alpha$. It goes around $\mathcal{P}$ until reaching w again and progresses on α, if possible. If this is impossible, the robot recognizes that it went around the boundary of $\mathcal{P}_0$. The robot computes the medial point v of $\mathcal{P}_0$.

Phase 2: Moving to the medial point of $\mathcal{P}_0$

Let u be the current position of the robot. The robot follows the segment $\overline{uv}$ as far as possible. Similarly as in the first phase of the algorithm, if the robot hits a polygon $\mathcal{P}$, it traverses the entire boundary of $\mathcal{P}$. Then, it computes the point w which is the farthest point from u in $\mathcal{P} \cap \overline{uv}$. It goes around $\mathcal{P}$ until reaching w again and progresses on α, if possible. If this is impossible and if the point v has not been reached, the robot recognizes that v is inside an obstacle $\mathcal{O}$, and executes phase 3. If the robot reaches v, it does not enter phase 3 of the algorithm and stops.

Phase 3: Rendezvous around the medial obstacle of the terrain

The robot goes around the obstacle $\mathcal{O}$ until it reaches a vertex s. The robot produces the modified label μ^* consisting of the binary representation of the label μ of the robot followed by a 1 and then followed by $|\mu|$ zeros. This phase consists of $|\mu^*|$ stages. In stage i, the robot completes two tours of the boundary of $\mathcal{O}$, starting and ending in s, clockwise if the i-th bit of μ^* is 1 and counterclockwise otherwise.

Let $\overline{u_1u_2}$ and $\overline{u_2u_3}$ be consecutive segments in clockwise order (resp. counterclockwise order) of a cycle. For a given walk f of a robot a, we say that the robot *traverses in a clockwise way* (resp. *in a counterclockwise way*) a vertex u_2 of a cycle at time t if $f(t) = u_2$ and there exist positive reals ϵ_1 and ϵ_2 and points y and z such that $y = f(t - \epsilon_1)$ is an internal point of $\overline{u_1u_2}$, $z = f(t + \epsilon_2)$ is an internal point of $\overline{u_2u_3}$ and the robot walks in $\overline{u_1u_2} \cup \overline{u_2u_3}$ during the time period $[t - \epsilon_1, t + \epsilon_2]$.

Before establishing the correctness and cost of Algorithm RVO, we need to show the following two lemmas.

Lemma 2. *Consider two robots on cycle $\mathcal{C}$. Suppose that one robot executes a tour of $\mathcal{C}$ in some sense of rotation, starting and ending in v. If during the same* *period of time, the other robot either traverses v for the first time in the other sense of rotation or does not traverse it at all, then the two robots meet.*

Lemma 3. *Consider two robots on a cycle $\mathcal{C}$ and let $k \geq 0$ be an integer. If a robot executes either a partial tour of $\mathcal{C}$ followed by at most k tours of $\mathcal{C}$, or at most k tours of $\mathcal{C}$ followed by a partial tour of $\mathcal{C}$, while the second robot executes $k+2$ tours of $\mathcal{C}$, then the two robots meet.*

Theorem 8. *Algorithm RVO guarantees rendezvous at cost $O(P + x|L|)$ for any two robots in any terrain for which x is the largest perimeter of an obstacle.*

Proof. Let a_1 and a_2 be the two robots that have to meet. The first phase of the algorithm that consists in reaching $\mathcal{P}_0$ and making the tour of the boundary of $\mathcal{P}_0$ costs at most $3P$, since the boundary of each polygon of the terrain is traversed at most twice and the total length of parts of α inside the terrain is at most P. For the same reason as in phase 1, the total cost of phase 2 is at most $3P$.

If the medial point of $\mathcal{P}_0$ is inside the terrain, then the robots meet at the end of phase 2 at total cost of at most $12P$. Otherwise, both robots eventually enter phase 3 of the algorithm and they are on the boundary of the obstacle $\mathcal{O}$ containing the medial point of $\mathcal{P}_0$. The cost follows from the fact that each robot travels a distance $O(x|L|)$ in phase 3. Indeed, each robot executes at most $2|L|+1$ stages and each stage costs at most $2x$. Hence it remains to show that rendezvous occurs in this case as well.

Assume for contradiction that the two robots never meet. Notice that the modified label l^* cannot be the suffix of the modified label L^*. Indeed, if $|l^*| = |L^*|$ then the two labels are different since $l \neq L$, and otherwise the second part of l^*, consisting of 1 followed by $|l|$ zeros, cannot be the suffix of L^*. Hence, there exists an index i such that the $(|l^*| - i)$-th bit of l^* differs from the $(|L^*| - i)$-th bit of L^*. We call *important* stages the $(|l^*| - i)$-th stage of the robot with label l and the $(|L^*| - i)$-th stage of the robot with label L.

For $j = 1, 2$, let t_j be the moment when robot a_j enters its important stage and let t' be the first moment when both robots have finished the execution of the algorithm. Suppose by symmetry that $t_1 \leq t_2$, i.e., robot a_1 was the first to enter its important stage. Then a_2 must have entered its important stage during the first tour of the important stage of a_1. Otherwise, robot a_2 would have completed $2i+2$ tours between t_2 and t', while robot a_1 would have completed at most $2i+1$ tours. Hence, the two robots would have met in view of Lemma 3. Hence, from the time t_2, robot a_2 completes one tour in some sense of rotation, starting and ending at a vertex v, while robot a_1 either traverses v for the first time in the other sense of rotation or does not traverse it at all. Hence by Lemma 2, the two robots meet. □

The following result gives a lower bound matching the cost of Algorithm RVO.

Theorem 9. *There exist terrains for which the cost of any rendezvous algorithm is $\Omega(P + x|L|)$. This holds for arbitrarily small $D > 0$.*

4 Discussion of parameters

We presented rendezvous algorithms, analyzed their cost and proved matching lower bounds in all considered scenarios. However, it is important to note that the formulas describing the cost depend on the chosen parameters in each case. All our results have the following form. For a given scenario we choose some parameters (among D, P, x, l, L), show an algorithm whose cost in any terrain is $O(f)$, where f is some simple function of the chosen parameters, and then prove that for some class of terrains any rendezvous algorithm requires cost $\Omega(f)$, which shows that the complexity of our algorithm cannot be improved in general, for the chosen parameters.

This yields the question which parameters should be chosen. In the case of complexities D and $\Theta(P)$, this choice does not seem controversial, as here D and P are very natural parameters, and the only ones in these simple cases. However, for the two scenarios with incoherent compasses and with the presence of obstacles, there are several other possible parameters, and their choice may raise a doubt. As mentioned in the introduction, in these two scenarios, distinct labels of robots are necessary to break symmetry, since rendezvous is impossible for anonymous robots. Hence any rendezvous algorithm has to use labels l and L as inputs, and thus the choice of these labels as parameters seems natural. By contrast, the choice of parameter x may seem more controversial. Why do we want to express the cost of a rendezvous algorithm in terms of the largest perimeter of an obstacle? Are there other natural choices of parameter sets? What are their implications?

Let us start by pondering the second question. It is not hard to give examples of other natural choices of parameters for the two scenarios with incoherent compasses and with the presence of obstacles. For example, in the hardest scenario (without a map), we could drop parameter x and try to express the cost of the same Algorithm RVO only in terms of D, P, l, and L. Since $x \leq P$, we would get $O(P|L|)$ instead of $O(P + x|L|)$. Incidentally, as in our lower bound example of terrains we have $x = \Theta(P)$, this new complexity $O(P|L|)$ is optimal for the same reason as the former one.

Another possibility would be adding, instead of dropping a parameter. We could, for example, add the parameter P_e which is the length of the external perimeter of the terrain, i.e., the perimeter of polygon $\mathcal{P}_0$. Then it becomes natural to modify Algorithm RVO as follows. The first two phases are the same. In the third phase, the robot goes around obstacle $\mathcal{O}$ and compares its perimeter to P_e. If the perimeter of $\mathcal{O}$ is smaller (or equal), then the algorithm proceeds as before, and if it is larger, then the robot goes back to the boundary of $\mathcal{P}_0$ and executes Phase 3 on this boundary instead of the boundary of $\mathcal{O}$. The new algorithm has complexity $O(P+\min(x,P_e)|L|)$. Its complexity is again optimal because in our lower bound example we can choose the parameter $y=\min(x,P_e)$ and enlarge the largest of the two boundaries by lengthy but thin zigzags. Thus we can preserve the lower bound $\Omega(P+\min(x,P_e)|L|)$, even when x and P_e differ significantly.

The reason why we chose parameters D, P, l, L, and x instead of just D, P, l and L, is that complexity $O(P+x|L|)$ shows a certain continuity of the complexity of Algorithm RVO with respect to the sizes of obstacles: when the largest obstacle decreases, this complexity approaches $O(P)$ and it becomes $O(P)$ if there are no obstacles. In this case our algorithm coincides with Algorithm RV. This is not the case with complexity $O(P|L|)$. On the other hand, this choice coincides with $O(P+\ \min(x,P_e)|L|)$ in many important cases, for example for convex obstacles (as then we have $x<P_e$).

It is then natural to ask what happens if we add parameter x in the scenario with incoherent compasses and with the presence of obstacles but with the map. Obviously we could still use Algorithm RVO and get complexity $O(P+x|L|)$. However, our lower bound argument in this scenario gives in fact only $\Omega(D+\min(x,D)|l|)$. In our example we had $D=\Theta(x)$ but we only get $\Omega(D+x|l|)$ even if D is much larger than x. On the other hand, if D is much smaller than x, we can only get the lower bound $\Omega(D|l|)$ because it matches the complexity of $RVMO$ in this case. Hence it is natural to ask if there exists a rendezvous algorithm with cost $O(D+\min(x,D)|l|)$ for arbitrary terrains in this scenario. We leave this as an open question.

References

1. R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, and C. Rackoff, Random walks, universal traversal sequences, and the complexity of maze problems, Proc. Annual Symposium on Foundations of Computer Science FOCS'1979, 218-223.
2. S. Alpern, The rendezvous search problem, SIAM J. on Control and Optimization 33 (1995), 673-683.
3. S. Alpern, Rendezvous search on labelled networks, Naval Reaserch Logistics 49 (2002), 256-274.
4. S. Alpern and S. Gal, The theory of search games and rendezvous. Kluwer Academic Publ., 2002.
5. J. Alpern, V. Baston, and S. Essegaier, Rendezvous search on a graph, Journal of Applied Probability 36 (1999), 223-231.

6. E. Anderson and R. Weber, The rendezvous problem on discrete locations, Journal of Applied Probability 28 (1990), 839-851.
7. E. Anderson and S. Fekete, Asymmetric rendezvous on the plane, Proc. 14th Annual ACM Symp. on Computational Geometry, 1998.
8. E. Anderson and S. Fekete, Two-dimensional rendezvous search, Operations Res. 49 (2001), 107-118.
9. E. M. Arkin, J. S. B. Mitchell and C. D. Piatko, Bicriteria shortest path problems in the plane, In Proc. 3rd Canad. Conf. Comput. Geom, 153-156, 1991.
10. H. Blum, A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, Proc. Symp. Models for Perception of Speech and Visual Form, 362-380. MIT Press, 1967.
11. M. Cieliebak, P. Flocchini, G. Prencipe, N. Santoro, Solving the Robots Gathering Problem, Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP 2003), 1181-1196.
12. G. De Marco, L. Gargano, E. Kranakis, D. Krizanc, A. Pelc, U. Vaccaro, Asynchronous deterministic rendezvous in graphs, Theoretical Computer Science 355 (2006), 315-326.
13. F. Chin, J. Snoeyink, and C. A. Wang, Finding the Medial Axis of a Simple Polygon in Linear Time, Discrete Comput. Geom., 382–391, 1995.
14. A. Dessmark, P. Fraigniaud, D. Kowalski, and A. Pelc, Deterministic rendezvous in graphs, Algorithmica 46 (2006), 69-96.
15. P. Flocchini, G. Prencipe, N. Santoro, P. Widmayer, Gathering of asynchronous oblivious robots with limited visibility, Proc. 18th Annual Symposium on Theoretical Aspects of Computer Science STACS'2001, LNCS 2010, 247-258.
16. S. Gal, Rendezvous search on the line, Operations Research 47 (1999), 974-976.
17. J. Hershberger and S. Suri, An Optimal Algorithm for Euclidean Shortest Paths in the Plane, SIAM J. Comput. 28, 2215-2256, 1997.
18. R. Klasing, A. Kosowski, A. Navarra, Taking advantage of symmetries: gathering of asynchronous oblivious robots on a ring. Proc. 12th Int. Conf. on Principles of Distributed Systems, (OPODIS 2008), 446-462.
19. R. Klasing, E. Markou, A. Pelc, Gathering asynchronous oblivious mobile robots in a ring, Theoretical Computer Science 390 (2008), 27-39.
20. D. Kowalski, A. Malinowski, How to meet in anonymous network. Theoretical Computer Science 399 (2008), 141-156.
21. E. Kranakis, D. Krizanc, N. Santoro and C. Sawchuk, Mobile agent rendezvous in a ring, Proc. 23rd International Conference on Distributed Computing Systems (ICDCS'2003), 592-599.
22. G. Prencipe, Impossibility of gathering by a set of autonomous mobile robots, Theoretical Computer Science 384 (2007), 222-231.
23. F. P. Preperata, The medial axis of a simple polygon, Mathematical Foundations of Computer Science (LNCS), 53:443-450,1977.
24. T. Schelling, The strategy of conflict, Oxford University Press, Oxford, 1960.
25. G. Stachowiak, Asynchronous Deterministic Rendezvous on the Line, SOFSEM 2009: 497-508.
26. X. Yu and M. Yung, Agent rendezvous: a dynamic symmetry-breaking problem, Proc. International Colloquium on Automata, Languages, and Programming (ICALP'1996), LNCS 1099, 610-621.
27. A. Ta-Shma, U. Zwick, Deterministic rendezvous, treasure hunts and strongly universal exploration sequences., Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), 599-608.