

# Collaboration without Communication: Evacuating Two Robots from a Disk

Sebastian Brandt  
ETH Zürich, Switzerland  
brandts@ethz.ch

Felix Laufenberg  
ETH Zürich, Switzerland  
felix.laufenberg@gmx.de

Yuezhou Lv  
Tsinghua University, Beijing, China  
lvyezhou@gmail.com

David Stolz  
ETH Zürich, Switzerland  
stolzda@ethz.ch

Roger Wattenhofer  
ETH Zürich, Switzerland  
wattenhofer@ethz.ch

## Abstract

We study the problem of evacuating two robots from a bounded area, through an unknown exit located on the boundary. Initially, the robots are in the center of the area and throughout the evacuation process they can only communicate with each other when they are at the same point at the same time. Having a visibility range of 0, the robots can only identify the location of the exit if they are already at the exit position. The task is to minimize the time it takes until both robots reach the exit, for a worst-case placement of the exit.

For unit disks, an upper bound of 5.628 for the evacuation time is presented in [8]. Using the insight that, perhaps surprisingly, a forced meeting of the two robots as performed in the respective algorithm does not provide an exchange of any non-trivial information, we design a simpler algorithm that achieves an upper bound of 5.625. Our numerical simulations suggest that this bound is optimal for the considered natural class of algorithms. For dealing with the technical difficulties in analyzing the algorithm, we formulate a powerful new criterion that, for a given algorithm, reduces the number of possible worst-case exits radically. This criterion is of independent interest and can be applied to any area shape.

## 1 Introduction

Imagine that two robots are trapped in the middle of a room with a single door. Their goal is to evacuate both of them via the door in the shortest possible time. However, there is a problem: The position of the door is unknown to them in the beginning. Moreover, the robots have no sight and no wireless communication; they cannot see the door or the other robot except when they are right on top of it, and they can only communicate when they are at the same point at the same time. However, they do know the shape of the room and share all information

before they start searching for the door. How should they divide the work of scouring the boundary for the door among them? Which routes should they take? Does it make sense to meet at some predefined point in order to exchange information?

We consider the above problem for the most fundamental room shape possible, namely, the unit disk. In more formal terms, the goal is to minimize the time for evacuating both robots from the room where the exit is assumed to be worst-case placed and the robots move with unit speed. The state-of-the-art algorithm for this problem due to Czyzowicz et al. [8] proceeds roughly as follows: First, both robots move to the same point on the perimeter, then they search the perimeter in different directions until they meet at the opposite point. At some predefined point in time during their search along the perimeter they leave the perimeter on symmetric non-linear routes in order to meet inside the circle upon which they return to their search. If a robot finds the exit at any point in time, then it immediately calculates the shortest route for meeting the other robot and subsequently brings the other robot to the exit. The authors of [8] show that this algorithm achieves an evacuation time of 5.628 and remark that it is possible to improve upon this result by truncating the detour to the middle slightly.

We note that the forced meeting of the robots cannot be used to exchange any non-trivial information. Moreover, the requirement to meet introduces dependencies between the parameters of the detour such as position, length, shape and angle (with respect to the perimeter). We prove that removing this requirement indeed allows for an improved algorithm that utilizes the independence of the aforementioned parameters while preserving the simplicity of the algorithm. In fact, we present an algorithm that simplifies the algorithm described above by omitting the forced meeting and instead using one (symmetric) detour (per robot) that is a straight line with fixed depth. An important point in omitting the forced meeting is that there is actually an implicit exchange of information between the robots even before any meeting: When a robot finds the exit, the best it can do is to meet the other robot as quickly as possible in order to communicate the location of the exit. Conversely, from not being visited by the other robot up to some point a robot can deduce that the exit does not lie in a certain part of the perimeter.

Furthermore, we show that, surprisingly, the shape of the detour and its angle to the perimeter do not affect the evacuation time if they are chosen from some reasonable range. In particular, our linear detour is optimal for the parameters chosen for the depth and the position of the detour.

Our algorithm achieves an evacuation time of 5.625, thereby slightly improving upon the previously best known upper bound. For the class of algorithms as described above with exactly one symmetric detour per robot, our numerical simulations suggest that this bound is optimal (up to numerical precision, of course). A theoretical substantiation for this optimality claim is given by the fact that for our algorithm there are three different worst-case exit placements with the same evacuation time (again, up to numerical precision). These three exit positions are characteristic for the algorithms from the mentioned class—it stands to reason that, in an optimal algorithm, they have to exhibit the same evacuation time (since otherwise the parameters of the algorithm could be changed locally in a way that improves the evacuation time for the worst of the three points) and that there is only one algorithm that has the same worst-case evacuation time for these three exit positions.

A fundamental problem regarding evacuation from a disk is that the evacuation time for a fixed algorithm and a fixed exit placement is usually the solution to some equation of the kind  $x = \cos(x)$  (only more complex). For equations of this kind no closed-form solutions are known in general, which makes it difficult to find the worst-case exit placement for a fixed algorithm, not to mention to find an algorithm with an improved worst-case evacuation time. We take a substantial step in remedying this problem by proving that a very specific condition must be satisfied for an exit in order to be worst-case placed. In “reasonable” algorithms this condition is satisfied at only a few exit positions which makes it a powerful tool for determining that an exit is not worst-case placed. In fact, in order for an exit to be worst-case placed, it must satisfy one of the two following conditions: 1) The movement of one of the two robots at the exit point, resp. pick-up point<sup>1</sup>, is not differentiable, or 2) the angles  $\beta$  and  $\gamma$  between the line connecting exit and pick-up point and the directions of movement of the robots at the exit, resp. the pick-up point, satisfy  $2 \cos \beta + \cos \gamma = 1$ . Moreover, the presented tool is not restricted to the disk—it can be applied to any room shape. For the analysis of our aforementioned algorithm, we rely heavily on this tool. In fact, one might consider the development of this tool as the foremost contribution of this work, while its application to the disk scenario may serve as an example of its practicability.

## 1.1 Related Work

A thriving area in the context of problems involving mobile agents are search problems in all its diversity. Such problems include ants searching for food (cf., e.g., [11, 12, 13]), rendezvous problems (cf., e.g., [1, 10, 17]), pursuit-evasion games (cf., e.g., [14, 20, 21]) and graph exploration problems (cf., e.g., [15, 16, 19]), just to name a few. Another example from the class of search problems are evacuation problems, where one or multiple robot(s) search for one or multiple exit(s) through which usually all of the robots have to evacuate. Evacuation problems have been studied in a centralized setting in which the robots know the terrain they search and where the other robots are, and in a distributed setting where the knowledge of the robots is restricted to the area they have already explored.<sup>2</sup> Very recent results concerning optimal strategies on graphs in both settings can be found in [4]. In the following, we assume that the area is known to the robots. Moreover, we assume that the exit is worst-case placed and that the robots move with unit speed.

Evacuation problems can be grouped into two main categories, namely, evacuation problems on graphs and geometric evacuation problems. Since our paper deals with a problem from the latter class, we will focus on the related work in this domain. Another distinction is given by the model of communication between the robots: Here, we distinguish between instantaneous wireless communication and non-wireless communication where explicit communication can only take place when the communicating entities are at the same point.

In the geometric setting, research has considered different areas from which the robots have to escape. The famous cow-path problem asks how long it takes

---

<sup>1</sup>Recall that upon finding the exit, a robot immediately takes the shortest possible tour to meet the other robot and communicate the location of the exit. We call the point where this meeting happens the *pick-up point*.

<sup>2</sup>Of course, all kinds of intermediate models are possible.

a single robot (or cow) to evacuate through a worst-case placed exit on a line, in terms of the distance  $d$  between initial position and exit. The correct answer of  $9d$  (up to lower order terms) was given by Beck and Newman [3] already in 1970, and later rediscovered by Baeza-Yates et al. [2]. In [5], Chrobak et al. show that, somewhat surprisingly, the same is true for the evacuation time of multiple robots on the line in the non-wireless communication model.

For the case of two robots in equilateral triangles and squares, Czyzowicz et al. [9] present optimal evacuation trajectories, given wireless communication.

Study of the unit disk as the confining environment was initiated by Czyzowicz et al. in [7]. The authors present upper bounds of  $3 + 2\pi/k$  and  $3 + \pi/k + O(k^{-4/3})$  and lower bounds of  $3 + 2\pi/k - O(k^{-2})$  and  $3 + \pi/k$  for the non-wireless and the wireless communication model, respectively, where  $k$  is the number of robots. Moreover, they give better upper and lower bounds for the case of 2 and 3 robots, amongst them a lower bound of approximately 5.199 and an upper bound of approximately 5.74 for the case of two robots in the non-wireless model. In [8], Czyzowicz et al. improve the latter two bounds to a lower bound of approximately 5.255 and an upper bound of 5.628. Lamprou et al. [18] present (partly matching) upper and lower bounds for two robots in the wireless model where one robot, deviating from the above, has speed of larger than 1. Finally, in [6], Czyzowicz et al. consider variations of the problem of evacuating from a disk where the two robots do not know their own initial locations.

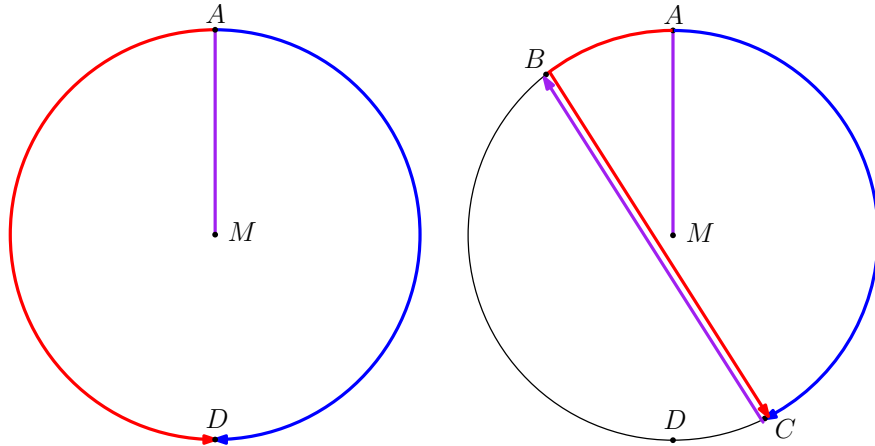
For our paper, the algorithms from [7] and [8] that achieve the upper bounds for two robots in the non-wireless communication model are of particular interest. The algorithm from [7] proceeds as follows: Starting in the center  $M$  of the disk, both robots move to the same point  $A$  on the perimeter and start searching for the exit in opposite directions. When one of the robots finds the exit, it picks the other robot up as fast as possible and returns with it to the exit. This results in an upper bound of approximately 5.74. For an illustration of the algorithm we refer to Figure 1.

The authors of [8] improve on the previously suggested algorithm by incorporating a *forced meeting* of the two robots before all of the perimeter is searched. For this, the robots leave the perimeter in a straight line, symmetric to each other, until they meet, and then return to their search of the perimeter if the exit has not been found yet. The authors were able to improve on this algorithm even more by moving towards the meeting point in a triangular fashion. The reasons for this further improvement are more subtle and explained in Section 3.1. For an illustration of the two algorithms with a forced meeting, we refer to Figure 2.

## 1.2 Model

The specifics of the model for our robot evacuation problem, developed in [7], are as follows: The area from which the robots have to escape is a disk of radius 1. Somewhere on its perimeter, there is a point, called *exit*, which two robots, initially placed in the center of the disk, have to find and evacuate through. The task of the robots is to minimize the time until both robots have reached the exit, which we call the *evacuation time*. We assume that the location of the exit on the perimeter is worst-case for the algorithm the two robots perform, i.e., the exit position maximizes the evacuation time.

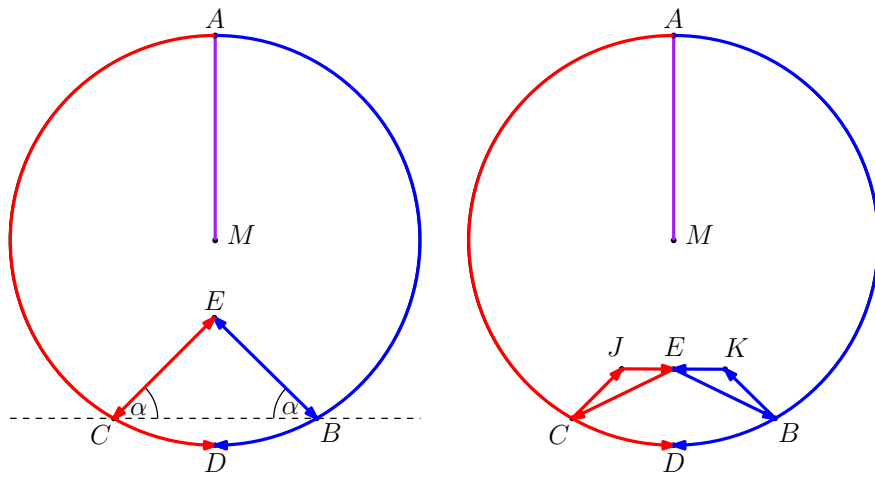
The robots itself are point-shaped and move at unit speed. Changing direction



(a) Trajectories of the two robots if the exit is at point  $D$ . Purple parts of the trajectories indicate that both the red and the blue robot travel along the respective segment.

(b) Trajectories of the two robots if the exit is at point  $B$ . The red robot collects the blue robot at the pick-up point at  $C$  and then both robots travel together to the exit.

Figure 1: Illustration of the evacuation algorithm described in [7].



(a) First suggestion from [8], achieving a runtime of approximately 5.644 for  $\alpha = 0$ . The blue robot first moves from  $M$  to  $A$ , then along the perimeter to  $B$ , then to  $E$ , back to  $B$  and finally to  $D$  (if the exit is not found before).

(b) Improved algorithm from [8], achieving a runtime of 5.628. Instead of the previously used linear cut, the robots move along a triangular cut. Again, the robots follow the indicated trajectories only to the end if the exit is at  $D$ .

Figure 2: Illustration of the evacuation algorithms described in [8].

takes no time and communication is also instantaneous, but only possible if both robots are at the same point.<sup>3</sup> Since this is the case in the beginning, the robots can exchange all information about each others algorithm before they start searching for the exit. The robots have no vision, and therefore can only identify the exit position when they are at the exact location of the exit. Computation also takes no time and we assume that the robots are able to actually perform all necessary computations. The robots know the shape of the area and they have the same sense of direction, i.e., we may assume that they have the same underlying coordinate system.

### 1.3 Notation

In the following, we give an overview of the notation and the most important terms we use.

**$R_1$  and  $R_2$**  References for the two robots. We will call the robot that finds the exit first  $R_1$ , and the other robot  $R_2$ .

$\widehat{AB}$  For two points  $A$  and  $B$  on the perimeter,  $\widehat{AB}$  denotes the shorter arc from  $A$  to  $B$  along the perimeter.

$AB$  Denotes the straight line between  $A$  and  $B$ .

$|\widehat{AB}|$  or  $|AB|$  Denote the lengths of  $\widehat{AB}$ , resp.  $AB$ .

**Cut** A movement of a robot from the perimeter onto the disk and back to the point where the perimeter was left. Note that a cut can take any shape in general. However, in our algorithm, the term *cut* describes a linear cut, i.e., a movement from the perimeter onto the disk and back on a straight line.

**Cut length** The distance traveled when moving along a cut.

**Cut depth** Only used if the cut is linear, in which case the cut depth is defined as half of the cut length.

**Cut position** Point where a robot leaves the perimeter to perform a cut.

**Meeting protocol** A term coined in [8]. When  $R_1$  finds the exit, the best it can do to minimize the evacuation time is to compute (and take) the shortest route to meet  $R_2$ . Since  $R_1$  knows the algorithm  $R_2$  follows (and therefore also that  $R_2$  has not found the exit so far),  $R_1$  can actually determine this shortest route.<sup>4</sup> Note that this route is always a straight line since otherwise there would be a shorter route, by the triangle inequality. After meeting each other, both robots travel straight to the exit. This process of

---

<sup>3</sup>Note that a robot can also infer information from the fact that the other robot is not at the same point as it is. For instance, it may conclude that the other robot has not already found the exit in some specific segment of the perimeter, since otherwise the other robot would have picked him up at the latest at the current position. This indirect information transfer plays an important role in our arguments that the robots cannot infer any non-trivial information from a forced meeting.

<sup>4</sup>We emphasize that  $R_1$  does not calculate a shortest route to the point where  $R_2$  is when  $R_1$  finds the exit, but rather the shortest route for picking  $R_2$  up, knowing that and how  $R_2$  will move until being picked up.

picking the other robot up after finding the exit and traveling to the exit together is called the *meeting protocol*. If  $R_1$  finds the exit at time  $t$  and the aforementioned shortest route has length  $x$ , then the evacuation time is  $t + 2x$ .

**Pick-up point** The point where  $R_2$  is picked up by  $R_1$  according to the meeting protocol.

## 2 Determining the Worst-case Placement of the Exit

If an algorithm for the two robots is fixed, it is still a challenging task to determine the worst-case placement of the exit and thereby the evacuation time. One reason is that already determining the pick-up point for a fixed exit placement often involves solving equations where polynomial and trigonometric functions in some variable  $x$  occur side by side. For equations of this kind no closed-form solutions are known in general.

In this section, we develop a new technique to determine possible candidates for the worst-case placement of the exit. More precisely, we give a criterion that determines for a pair (exit, pick-up point) whether the exit can be excluded from the list of candidates of worst-case placed exits, by only looking at the behaviour of the algorithm in  $\varepsilon$ -neighborhoods of the exit and the pick-up point. The criterion is quite strong in the following sense: Let  $\beta$  denote the angle between the straight line from exit to pick-up point and the direction of the movement of  $R_1$  at the exit. Let  $\gamma$  denote the angle between the straight line from exit to pick-up point and the direction of the movement of  $R_2$  at the pick-up point. Then, a very specific relation between  $\beta$  and  $\gamma$  has to be satisfied in order that the exit from our pair (exit, pick-up point) is not excluded from the list of possible worst-case placed exits.

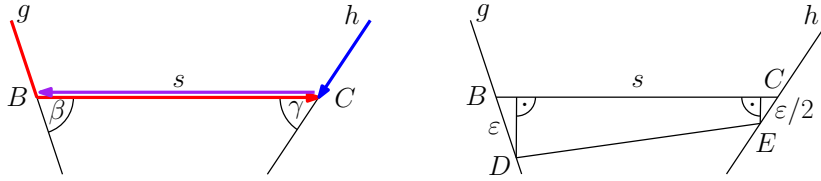
In order to be able to exclude an exit, the movement of the two robots at the exit and the corresponding pick-up point have to be differentiable (a property that holds for almost all possible exit points in reasonable algorithms).

We make a natural distinction between two cases regarding the direction of the movement of the two robots at the exit and the pick-up point: If according to the algorithm, the two robots would move to the same side of the infinite line through exit and pick-up point if they did not find the exit, resp. were not picked up, as described above, then we say that the movement of the two robots (at exit and pick-up point) is *conform*. If the two robots would move to different sides of the infinite line, we say that their movement is *converse*. The cases where one or both robots would move *on* the infinite line can be arbitrarily considered to belong to one of the two cases.

Since our considerations in this section are independent of the shape of the area, we phrase them as generally as possible, i.e., for an arbitrarily shaped area.

### 2.1 The Case of Conform Movement

Consider two lines  $g$  and  $h$  and two points  $B$  on  $g$  and  $C$  on  $h$ . Denote the straight line from  $B$  to  $C$  by  $s$ . For simplicity, we assume that  $s$  has length 1. The presented results can easily be scaled to any length of  $s$ . Denote the angle



(a) The basic setting for the case of conform movement: Robot  $R_1$  finds the exit at  $B$  and picks  $R_2$  up at  $C$ . A very similar picture applies if we do not require straight-line movements of the two robots, but only differentiable ones.

(b) If  $2 \cos \beta + \cos \gamma < 1$ , then traveling from  $B$  to  $E$  via  $D$  takes more time than traveling from  $B$  to  $E$  via  $C$ . Presuming  $R_1$  picks  $R_2$  up at  $C$  if the exit is at  $B$ , then  $R_1$  is too late to pick  $R_2$  up at  $E$  if the exit is at  $D$ .

Figure 3: Zooming in at a potential worst-case exit and the corresponding pick-up point.

between  $g$  and  $s$  by  $\beta$  and the angle between  $s$  and  $h$  by  $\gamma$ . In particular, we have  $\beta, \gamma \leq \pi$ . For an illustration, we refer to Figure 3(a).

In the following, we consider an arbitrary, but fixed evacuation algorithm for two robots in some arbitrarily shaped bounded area. We assume that  $B$  is the location of the exit and  $C$  the pick-up point where  $R_1$  picks  $R_2$  up via  $s$ . Furthermore, we assume that *locally* (i.e., in some  $\varepsilon$ -neighborhood of the exit),  $R_1$  arrives at  $B$  via  $g$  and would continue on  $g$  if it didn't find the exit at  $B$ . More precisely, the part of  $g$  where  $R_1$  would continue forms the angle  $\beta$  with  $s$  (to specify the direction of  $R_1$ ). Analogously, we assume that locally,  $R_2$  arrives at  $C$  via  $h$  and would continue on  $h$  if it wasn't picked up by  $R_1$ . Here the part of  $h$  where  $R_2$  would continue forms the angle  $\gamma$  with  $s$ .

Now, we consider two cases, namely,  $2 \cos \beta + \cos \gamma < 1$  and  $2 \cos \beta + \cos \gamma > 1$ . We will show for both cases that  $B$  is not a worst-case placed exit even if the trajectories of the robots at  $B$  and  $C$  are only differentiable and not necessarily straight lines in an  $\varepsilon$ -neighborhood.

### 2.1.1 The Case $2 \cos \beta + \cos \gamma < 1$

Let  $0 < \varepsilon \leq 1/2$ . We assume that in the  $\varepsilon$ -neighborhood of  $B$ , resp.  $C$ ,  $R_1$  and  $R_2$  behave as described above. Let  $D$  be the point that  $R_1$  would reach in distance  $\varepsilon$  from  $B$  if it continued on  $g$ . Let  $E$  be the point that  $R_2$  would reach in distance  $\varepsilon/2$  from  $C$  if it continued on  $h$ . For the moment, let  $\beta \leq \pi/2$ . For an illustration, we refer to Figure 3(b).

Set  $z_{\beta, \gamma} := 1 - 2 \cos \beta - \cos \gamma$  which is positive due to the constraint on  $\beta$  and  $\gamma$ . Note that  $z_{\beta, \gamma} \leq 2$  since  $\beta \leq \pi/2$ . For our proof that placing the exit at  $B$  is not worst-case, we need the following two technical lemmas.

**Lemma 2.1.** *If  $2 \cos \beta + \cos \gamma < 1$  and  $\beta \leq \pi/2$ , then  $|DE| \geq 1 - \varepsilon/2 + z_{\beta, \gamma} \cdot \varepsilon/2$ .*

*Proof.* By adding orthogonal lines to  $BC$  through  $D$  and  $E$  to the picture (cf.



Figure 3(b)), we see that we can express  $|DE|$  as follows<sup>5</sup>:

$$\begin{aligned} |DE| &= \sqrt{(1 - \varepsilon \cos \beta - \varepsilon/2 \cdot \cos \gamma)^2 + (\varepsilon \sin \beta - \varepsilon/2 \cdot \sin \gamma)^2} \\ &= \sqrt{1 + \varepsilon^2(\sin^2 \beta + \cos^2 \beta) + 1/4 \cdot \varepsilon^2(\sin^2 \gamma + \cos^2 \gamma) + \varepsilon^2(\cos \beta \cos \gamma - \sin \beta \sin \gamma) - \varepsilon(2 \cos \beta + \cos \gamma)} \end{aligned}$$

Using the trigonometric identities  $\sin^2 \beta + \cos^2 \beta = 1$ ,  $\sin^2 \gamma + \cos^2 \gamma = 1$  and  $\cos(\beta + \gamma) = \cos \beta \cos \gamma - \sin \beta \sin \gamma$ , the fact that the minimum value of the cosine function is  $-1$ , and the definition of  $z_{\beta, \gamma}$ , we obtain:

$$\begin{aligned} |DE| &\geq \sqrt{1 + \varepsilon^2 + 1/4 \cdot \varepsilon^2 - \varepsilon^2 + \varepsilon(z_{\beta, \gamma} - 1)} \\ &\geq \sqrt{1 - \varepsilon + 1/4 \cdot \varepsilon^2 + z_{\beta, \gamma} \cdot \varepsilon + 1/4 \cdot z_{\beta, \gamma}^2 \cdot \varepsilon^2 - 1/2 \cdot z_{\beta, \gamma} \cdot \varepsilon^2} \\ &= \sqrt{(1 - \varepsilon/2 + z_{\beta, \gamma} \cdot \varepsilon/2)^2} \\ &= 1 - \varepsilon/2 + z_{\beta, \gamma} \cdot \varepsilon/2 \end{aligned}$$

The second inequality follows since  $\beta \leq \pi/2$  implies  $z_{\beta, \gamma} \leq 2$  which in turn implies  $1/4 \cdot z_{\beta, \gamma}^2 \cdot \varepsilon^2 - 1/2 \cdot z_{\beta, \gamma} \cdot \varepsilon^2 \leq 0$ .  $\square$

**Lemma 2.2.** *If  $2 \cos \beta + \cos \gamma < 1$  and  $\beta \leq \pi/2$ , then  $\varepsilon + |DF| \geq 1 + |CF| + z_{\beta, \gamma} \cdot \varepsilon/2$  for all  $F$  on  $CE$ .*

*Proof.* Let  $F$  be some point on  $CE$ . Then, by the triangle inequality,  $|DF| \geq |DE| - |EF|$ . Using Lemma 2.1, we obtain

$$\begin{aligned} \varepsilon + |DF| &\geq \varepsilon + |DE| - |EF| \\ &\geq 1 + \varepsilon/2 + z_{\beta, \gamma} \cdot \varepsilon/2 - |EF| \\ &= 1 + |CE| + z_{\beta, \gamma} \cdot \varepsilon/2 - |EF| \\ &= 1 + |CF| + z_{\beta, \gamma} \cdot \varepsilon/2 . \end{aligned}$$

$\square$

By comparing the evacuation times of the case where the exit is located at  $B$  and the case where the exit is located at  $D$  instead, we obtain the following theorem that states that placing the exit at  $B$  is not worst-case:

**Theorem 2.3.** *If  $2 \cos \beta + \cos \gamma < 1$  and  $\beta \leq \pi/2$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

*Proof.* Let  $t$  be the time it took  $R_1$  to get to point  $B$ . Then at time  $t + 1$ ,  $R_2$  is at point  $C$ . Moreover, the evacuation time for the exit position at  $B$  is  $t + 2$ .

Consider the case that the exit is located at  $D$  instead. In this case,  $R_1$  finds the exit at time  $t + \varepsilon$ . We claim that  $R_2$  is picked up at a point in time later than  $t + 1 + \varepsilon/2$ . Suppose otherwise. Then,  $R_2$  is picked up at some point  $F$  on  $CE$  at time  $t + 1 + |CF|$ . But, even if  $R_1$  heads straight for point  $F$  after finding the exit at  $D$ , he arrives there only at time  $t + \varepsilon + |DF|$  which, by Lemma 2.2, is strictly larger than  $t + 1 + |CF|$ , thereby yielding a contradiction and proving the claim.

<sup>5</sup>Note that the expression for  $|DE|$  also holds if  $\gamma > \pi/2$ .

Robot  $R_2$  is at point  $E$  at time  $t + 1 + \varepsilon/2$  and, by Lemma 2.1, the distance between  $D$  and  $E$  is strictly larger than  $1 - \varepsilon/2$ . Thus,  $R_2$  reaches the exit at  $D$  strictly later than at time  $t + 2$ . It follows that the evacuation time for the exit position at  $D$  is strictly larger than the evacuation time for the exit at  $B$ . Hence,  $B$  cannot be a worst-case placed exit position.  $\square$

Up to now, in our quest for showing that exits can only be worst-case placed under very specific conditions, we only considered the case that the robots move locally in a straight line around the exit and the pick-up point. Now we extend our approach to the case of curves that are differentiable in  $\varepsilon$ -neighborhoods around the exit and the pick-up point. Thereby, we can finally tackle fundamental shapes like circles.

Let  $g, h, B, C, s, \beta, \gamma, \varepsilon, D$  and  $E$  be defined as above with the slight difference that  $g$  and  $h$  now indicate the directions of the movement of  $R_1$  and  $R_2$  at point  $B$ , resp.  $C$ , linearly extended (which not necessarily coincides with the routes the robots take, for any point except  $B$ , resp.  $C$ ). Again, we start with the assumption that  $R_1$  finds the exit at  $B$  and picks  $R_2$  up at  $C$ . Let  $g'$  and  $h'$  denote the curves along which  $R_1$  and  $R_2$  would move if they did not find the exit at  $B$ , resp. were not picked up at  $C$ . From our definitions it follows that  $B$  lies on  $g'$ ,  $C$  lies on  $h'$ , and  $g$  and  $h$  indicate the derivative of  $g'$ , resp.  $h'$ , in  $B$ , resp.  $C$ .

The direction of the robots on  $g'$ , resp.  $h'$ , is analogous to the direction of the movement on  $g$ , resp.  $h$ , in the previous case (without curves), i.e., when moving along  $g'$ , resp.  $h'$ , the robots are always on the same “side” with respect to  $s$  and they move towards the side which encloses  $\beta$ , resp.  $\gamma$ , with  $s$ . Let  $D'$  be the point on  $g'$  that  $R_1$  reaches by traveling a distance of  $\varepsilon$  from  $B$  on  $g'$  and  $E'$  the point on  $h'$  that  $R_2$  reaches by traveling a distance of  $\varepsilon/2$  from  $C$  on  $h'$ . Furthermore, for all  $0 \leq w \leq \varepsilon/2$ , let  $F_w$  be the point on  $h$  that  $R_2$  would reach by traveling a distance of  $w$  from  $C$  on  $h$  and  $F'_w$  the point on  $h'$  that  $R_2$  reaches by traveling a distance of  $w$  from  $C$  on  $h'$ . Again, we assume that  $2 \cos \beta + \cos \gamma < 1$  and set  $z_{\beta, \gamma} := 1 - 2 \cos \beta - \cos \gamma$ .

In our new setting, we obtain an analogous statement to Theorem 2.3.

**Theorem 2.4.** *If the trajectories of the two robots are differentiable around  $B$  and  $C$ , the following statement (still) holds: If  $2 \cos \beta + \cos \gamma < 1$  and  $\beta \leq \pi/2$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

*Proof.* Let  $\varepsilon$  be chosen small enough such that  $|DD'| \leq z_{\beta, \gamma} \cdot \varepsilon/6$  and  $|F_w F'_w| \leq z_{\beta, \gamma} \cdot \varepsilon/6$  for all  $0 \leq w \leq \varepsilon/2$ . This is possible since  $z_{\beta, \gamma}$  depends only on  $\beta$  and  $\gamma$  and (the directions of)  $g$  and  $h$  indicate the derivative of  $g'$  and  $h'$  in  $B$ , resp.  $C$ .

Now, using Lemma 2.2, we obtain the following statement for any point  $F'_w$  on  $h'$  between  $C$  and  $E'$  (which is quite similar to the statement of Lemma 2.2):

$$\begin{aligned} \varepsilon + |D'F'_w| &\geq \varepsilon + |D'F_w| - z_{\beta, \gamma} \cdot \varepsilon/6 \\ &\geq \varepsilon + |DF_w| - z_{\beta, \gamma} \cdot \varepsilon/6 - z_{\beta, \gamma} \cdot \varepsilon/6 \\ &\geq 1 + |CF_w| + z_{\beta, \gamma} \cdot \varepsilon/2 - z_{\beta, \gamma} \cdot \varepsilon/6 - z_{\beta, \gamma} \cdot \varepsilon/6 \\ &= 1 + |CF_w| + z_{\beta, \gamma} \cdot \varepsilon/6 . \end{aligned}$$

Now the theorem statement follows by a argumentation completely analogous to the proof of Theorem 2.3.  $\square$

So far, the above results are achieved under the condition that  $\beta \leq \pi/2$ . What if  $\beta > \pi/2$ ? In this case, independent of the trajectory of the two robots around  $B$ , resp.  $C$ , provided the trajectories are differentiable at  $B$ , resp.  $C$ , there exists an  $\varepsilon' > 0$  such that at time  $t + \varepsilon'$ ,  $R_1$  is at some point  $G$  with  $|GC| > |BC| = 1$ . Then, if the exit is placed at point  $G$ , the evacuation time is strictly larger than the evacuation time  $t + 2$  for the exit placement at  $B$  since  $R_2$  is at point  $C$  at time  $t + 1$  and  $|GC| > 1$ . Thus, if  $\beta > \pi/2$ , placing the exit at  $B$  is also not worst-case. We condense our observations into the following corollary.

**Corollary 2.5.** *If the trajectories of the two robots are differentiable around  $B$  and  $C$ , the following statement holds: If  $2 \cos \beta + \cos \gamma < 1$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

### 2.1.2 The Case $2 \cos \beta + \cos \gamma > 1$

Again, let  $0 < \varepsilon \leq 1/2$  and again we assume that in the  $\varepsilon$ -neighborhood of  $B$ , resp.  $C$ ,  $R_1$  and  $R_2$  behave as described above. Deviating from the previous notation, let  $D$  be the point on  $g$  that  $R_1$  reached in distance  $\varepsilon$  from  $B$  before arriving at  $B$ . Similarly, let  $E$  be the point that  $R_2$  reached in distance  $\varepsilon/2$  from  $C$  before arriving at  $C$ .

Set  $z_{\beta,\gamma} := 2 \cos \beta + \cos \gamma - 1$ . It follows that  $0 < z_{\beta,\gamma} \leq 2$  due to the constraint on  $\beta$  and  $\gamma$ . Note that in the current case, we do not have to specifically require that  $\beta \leq \pi/2$  in order to have  $z_{\beta,\gamma} \leq 2$ . Moreover, we observe that the constraint on  $\beta$  and  $\gamma$  already implies  $\beta \leq \pi/2$ .

The following lemma is the analogous version of Lemma 2.1 for the case of  $2 \cos \beta + \cos \gamma > 1$ . Note that, for technical reasons, the last summand on the right-hand side of the inequality is  $z_{\beta,\gamma} \cdot \varepsilon/4$ , and not  $z_{\beta,\gamma} \cdot \varepsilon/2$ . Apart from having to carry the additional factor of  $1/2$  through the proofs of the following theorems, this change does not affect our approach at all.

**Lemma 2.6.** *If  $2 \cos \beta + \cos \gamma > 1$ , then  $|DE| \geq 1 + \varepsilon/2 + z_{\beta,\gamma} \cdot \varepsilon/4$ .*

*Proof.* Similarly to the proof of Lemma 2.1, we obtain:

$$\begin{aligned}
|DE| &= \sqrt{(1 + \varepsilon \cos \beta + \varepsilon/2 \cdot \cos \gamma)^2 + (\varepsilon \sin \beta - \varepsilon/2 \cdot \sin \gamma)^2} \\
&= \sqrt{1 + \varepsilon^2(\sin^2 \beta + \cos^2 \beta) + 1/4 \cdot \varepsilon^2(\sin^2 \gamma + \cos^2 \gamma) + \varepsilon^2(\cos \beta \cos \gamma - \sin \beta \sin \gamma)} \\
&\quad + \varepsilon(2 \cos \beta + \cos \gamma) \\
&\geq \sqrt{1 + \varepsilon^2 + 1/4 \cdot \varepsilon^2 - \varepsilon^2 + \varepsilon(z_{\beta,\gamma} + 1)} \\
&\geq \sqrt{1 + \varepsilon + 1/4 \cdot \varepsilon^2 + 1/2 \cdot z_{\beta,\gamma} \cdot \varepsilon + 1/16 \cdot z_{\beta,\gamma}^2 \cdot \varepsilon^2 + 1/4 \cdot z_{\beta,\gamma} \cdot \varepsilon^2} \\
&= \sqrt{(1 + \varepsilon/2 + z_{\beta,\gamma} \cdot \varepsilon/4)^2} \\
&= 1 + \varepsilon/2 + z_{\beta,\gamma} \cdot \varepsilon/4
\end{aligned}$$

The second inequality follows since  $\varepsilon \leq 1/2$  and  $z_{\beta,\gamma} \leq 2$  together imply

$$1/2 \cdot z_{\beta,\gamma} \cdot \varepsilon + 1/16 \cdot z_{\beta,\gamma}^2 \cdot \varepsilon^2 + 1/4 \cdot z_{\beta,\gamma} \cdot \varepsilon^2 \leq (1/2 + 1/16 + 1/8) \cdot z_{\beta,\gamma} \cdot \varepsilon < z_{\beta,\gamma} \cdot \varepsilon .$$

□

Using Lemma 2.6, we can prove an analogous version of Theorem 2.3 for the case  $2 \cos \beta + \cos \gamma > 1$ .

**Theorem 2.7.** *If  $2 \cos \beta + \cos \gamma > 1$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

*Proof.* Let  $t$  be the time it took  $R_1$  to get to point  $D$ . Then at time  $t + \varepsilon$ ,  $R_1$  is at point  $B$  and in case that the exit is placed at  $B$ ,  $R_2$  is picked up at  $C$  at time  $t + 1 + \varepsilon$ . This implies that  $R_2$  is at point  $E$  at time  $t + 1 + \varepsilon/2$  unless it is picked up before. Moreover, the evacuation time for the exit position at  $B$  is  $t + 2 + \varepsilon$ .

Consider the case that the exit is located at  $D$  instead. Robot  $R_1$  finds the exit at time  $t$ . We claim that  $R_2$  is picked up at a point in time later than  $t + 1 + \varepsilon/2$ . Suppose otherwise, say,  $R_2$  is picked up at some point  $G$  at time  $t + t'$  with  $t' \leq 1 + \varepsilon/2$ . Since  $R_2$  would be at  $E$  at time  $t + 1 + \varepsilon/2$  if it was not picked up before, it follows that  $|EG| \leq 1 + \varepsilon/2 - t'$ . Moreover, since  $R_1$  reaches  $G$  at time  $t + t'$ , we obtain  $|DG| \leq t'$ . By the triangle inequality, this implies  $|DE| \leq 1 + \varepsilon/2$  which yields a contradiction to Lemma 2.6 and thereby proves the claim.

Thus, at time  $t + 1 + \varepsilon/2$ ,  $R_2$  is actually at point  $E$ , and since, by Lemma 2.6,  $|DE|$  is strictly larger than  $1 + \varepsilon/2$ ,  $R_2$  reaches the exit at  $D$  strictly later than at time  $t + 2 + \varepsilon$ . It follows that the evacuation time for the exit position at  $D$  is strictly larger than the evacuation time for the exit position at  $B$ . Hence, again,  $B$  cannot be a worst-case placed exit position.  $\square$

Analogously to the course of action in Section 2.1.1, we can adapt Theorem 2.7 for the more general case of differentiable curves. The proof follows by an argumentation similar to the proof of Theorem 2.4.

**Theorem 2.8.** *If the trajectories of the two robots are differentiable around  $B$  and  $C$ , the following statement (still) holds: If  $2 \cos \beta + \cos \gamma > 1$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

## 2.2 The Case of Converse Movement

We adopt the definitions from the case of conform movement with the exception that now  $\gamma$  denotes the angle between  $h$  and  $s$  (not  $s$  and  $h$ ). This change ensures that the two robots continue in different directions with respect to  $s$  provided that they do not find the exit, resp. that they are not picked up. Now we can apply the arguments from the conform case and thereby obtain completely analogous lemmas and theorems, in particular:

**Theorem 2.9.** *If the trajectories of the two robots are differentiable around  $B$  and  $C$  and  $2 \cos \beta + \cos \gamma < 1$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

**Theorem 2.10.** *If the trajectories of the two robots are differentiable around  $B$  and  $C$  and  $2 \cos \beta + \cos \gamma > 1$ , then there is an exit position that yields a larger evacuation time than placing the exit at  $B$ .*

As in the conform case, it makes sense to distinguish between these two theorems since the details in the respective proofs provide additional information:

On which side of 1 the term  $2 \cos \beta + \cos \gamma$  lies, determines to which direction one has to move on  $g'$  from  $B$  in order to obtain an exit with a larger evacuation time. If  $2 \cos \beta + \cos \gamma < 1$ , then shifting the exit at  $B$  in the direction of the movement of  $R_1$  (if it did not find the exit at  $B$ ) will provide an exit position with larger evacuation time (than the exit position at  $B$ ). If  $2 \cos \beta + \cos \gamma > 1$ , then shifting the exit at  $B$  in the reverse direction will provide an exit position with larger evacuation time. The last two statements hold for both the conform and the converse case.

### 3 Evacuating from a Disk

In this section we use the criterion  $2 \cos \beta + \cos \gamma \neq 1$ , which we developed in Section 2, in order to improve the upper bound for evacuating two robots from a unit disk to 5.625. Like the algorithm presented in [8], which achieves the previously best known upper bound of 5.628, our algorithm consists of each robot exploring its assigned half of the perimeter, only interrupted by exactly one detour each, called cut, to the inside of the circle (and symmetric to the other's cut). In contrast, the algorithm from [8] additionally contains a forced meeting of the two robots at the far end of the cuts (provided that none of the robots finds the exit beforehand in which case the meeting protocol is invoked).

We will show that, perhaps counterintuitively, the robots cannot infer any non-trivial information from this meeting that they could not have inferred from the previous course of events. Thus, such a meeting can be omitted. We will see in more detail that all the advantages of the meeting come from the actual movement of cutting to the middle and not from an explicit exchange of information.

Without the condition that the two robots actually have to meet at the endpoint of their (symmetric) cuts, many cuts are possible candidates for improving the runtime of the evacuation algorithm. A cut is determined by four properties: the position on the perimeter where the cut starts and ends, the shape of the cut, the angle at which the cut protrudes from the perimeter and the size of the cut, which corresponds to the cut depth in the case of a linear cut. As we will show, somewhat surprisingly, the shape of the cut is optimal if it is linear, for the choices of the other three parameters made in our algorithm. Similarly, we will show that the angle does not influence the performance of the algorithm if it is chosen in a reasonable range.

We will provide a choice of the remaining two parameters for linear cuts that achieves the stated bound of 5.625. Moreover, we give a rigorous proof for the evacuation time.

#### 3.1 The Algorithm $A(y, \alpha, d)$

In this section, we describe a parameterized evacuation algorithm (or a family of evacuation algorithms if you wish) and provide a partitioning of  $R_1$ 's half of the perimeter into segments that will be useful in the analysis of the evacuation time. Furthermore, we show that the forced meeting in the previously best algorithm from [8] does not help in exchanging non-trivial information between the two robots. In Section 3.2, we will prove that the parameters can actually be chosen in a way that improves the previously best known upper bound.

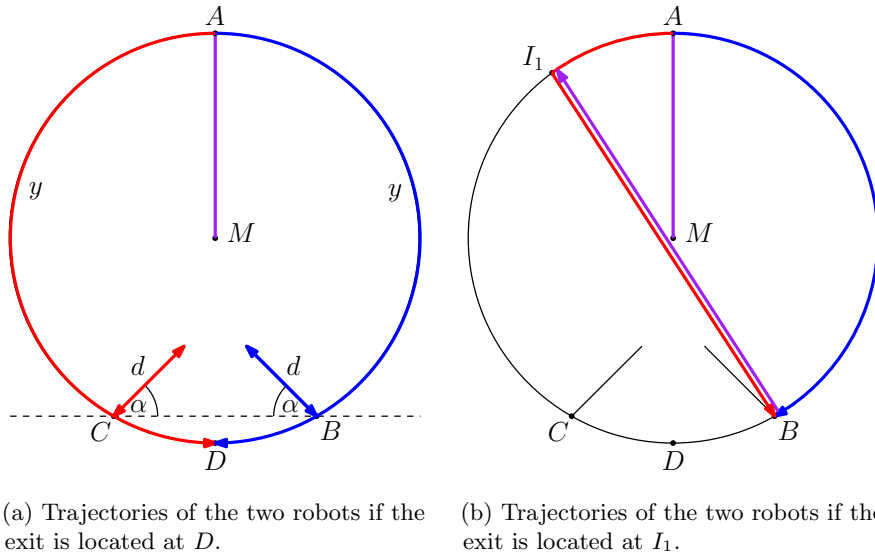
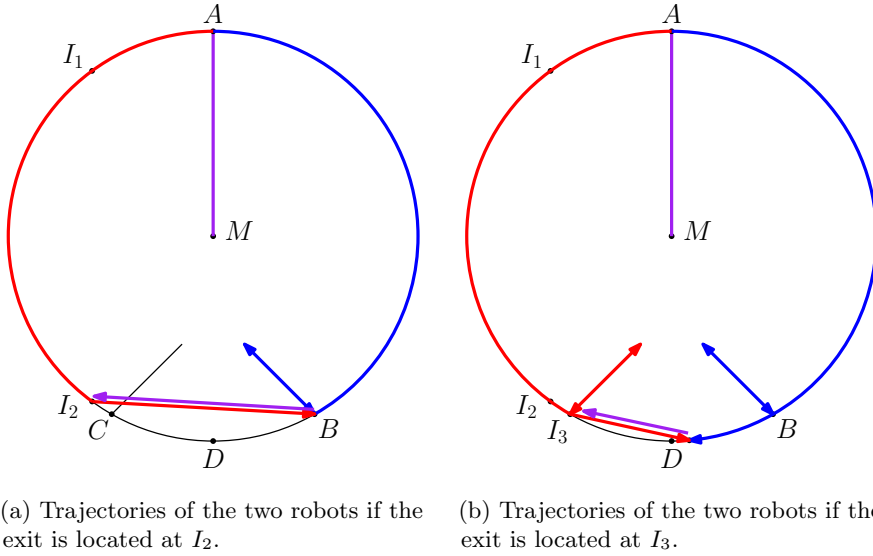


Figure 4: Illustration of Algorithm  $A(y, \alpha, d)$ .

Our parameterized algorithm  $A(y, \alpha, d)$  is similar to the algorithm proposed in [8]: From the center  $M$  of the disk, both robots move to the same point  $A$  of the perimeter and continue on the perimeter in opposite directions. At some point  $C$ , resp.  $B$ , where  $\widehat{AC} = \widehat{AB} = y$ , the robots leave the perimeter at angle  $\alpha$  in a straight line, until they reach depth  $d$  and then return straight to point  $C$ , resp.  $B$ . Then both robots continue to search for the exit along the perimeter until they meet at point  $D$ . If a robot finds the exit at any point in time, it immediately performs the meeting protocol to pick the other robot up and evacuate through the exit. Note that, for convenience,  $\alpha$  denotes the angle between the cuts and  $BC$ . For an illustration, we refer to Figure 4(a).

Now we examine Algorithm  $A(y, \alpha, d)$  in more detail, laying the foundation for the analysis of the evacuation time for a specific choice of the parameters  $y$ ,  $\alpha$  and  $d$  in Section 3.2. Since the described algorithm is symmetric, it is sufficient to analyze possible exit positions on one side of the symmetry axis, i.e., for one of the two robots. Without loss of generality, we assume that the exit lies on the arc from  $A$  to  $D$  that contains  $C$ , which implies that the robot that explores this arc is called  $R_1$  and the other one  $R_2$ . We partition this arc into four segments by specifying the points on the arc where one segment ends and the next one begins. Note that, for simplicity, we include any of these dividing points in both its adjacent segments if not explicitly specified otherwise. The choice of the segments depends on the parameters of our algorithm. For an illustration of the trajectories of the two robots for the exit placements at three important points defined in the following, we refer to Figure 4(b) and Figure 5.

1. **Segment  $\widehat{AI_1}$ :** Here,  $I_1$  is the point with the following property: If the exit is at  $I_1$ , then  $R_1$  will pick  $R_2$  up at point  $B$  before  $R_2$  performs its cut. In other words,  $I_1$  satisfies  $|\widehat{AI_1}| + |I_1B| = |AB|$ . This segment contains exactly those exit positions for which the evacuation time is not influenced



(a) Trajectories of the two robots if the exit is located at  $I_2$ . (b) Trajectories of the two robots if the exit is located at  $I_3$ .

Figure 5: Illustration of Algorithm  $A(y, \alpha, d)$  for important exit positions.

by the cut.

2. **Segment  $\widehat{I_1 I_2}$ :** Here,  $I_2$  is the point with the following property: If the exit is at  $I_2$ , then  $R_1$  will pick  $R_2$  up at point  $B$  after  $R_2$  performs its cut. In other words,  $I_2$  satisfies  $|\widehat{AI_2}| + |I_2 B| = |AB| + 2d$ . This segment contains exactly those exit positions for which the pick-up point lies on the cut.
3. **Segment  $\widehat{I_2 I_3}$ :** Here,  $I_3 = C$ . The exit positions in this segment are those for which  $R_1$  finds the exit before performing its cut, but  $R_2$  is picked up after performing its cut.
4. **Segment  $\widehat{I_3 D}$ :** For this segment, we explicitly specify that  $I_3$  itself does not belong to the segment. This segment contains exactly those exit positions that  $R_1$  reaches after performing its cut.

One main difference of our general algorithm to the one suggested in [8] is that in the latter the robots always cut far enough to meet each other. In the following, we argue that this meeting is not necessary since no real information can be shared. Consider the following three cases for the algorithm from [8]:

**Case 1:** The exit is located in one of the first three segments, excluding  $I_3$ . If  $R_1$  went now immediately to the meeting point on a straight line, then it would arrive there earlier than if it had not found the exit before performing its cut, because of the triangle inequality. Thus, because of symmetry reasons, it would also arrive earlier than  $R_2$  at the meeting point. Hence, by using the meeting protocol upon finding the exit,  $R_1$  picks  $R_2$  up before  $R_2$  actually reaches the meeting point. It follows that in Case 1,  $R_2$  never reaches the meeting point and therefore no information can be shared.

**Case 2:** The exit is located at  $I_3$ . In this case, there actually is an exchange of information at the meeting point, but the reason is that the predefined meeting point happens to be the pick-up point for the exit position at  $I_3$ . In other words, if  $R_1$  (but not  $R_2$ ) followed a completely different algorithm without a forced meeting but with the property that it finds the exit at  $I_3$  at the same time as in Algorithm  $A(y, \alpha, d)$ , then it would still pick  $R_2$  up at the same point at the same time and the resulting evacuation time would not change. Thus, even in this specific case, the benefit in the algorithm from [8] does not come from the forced meeting, but from the fact that  $R_2$  cuts far enough in the direction of the exit to be picked up at the tip of its cut.

**Case 3:** The exit lies behind  $R_1$ 's cut, in the fourth segment. At the meeting point, the only relevant information that can be shared is that neither robot has found the exit yet. However, both robots can deduce this information from the fact that they have not been picked up yet (see Case 1).

Note that in [8], two algorithms were presented, as described in Section 1.1: One with a linear cut and an improved one where the robots cut to the meeting point in a triangular fashion. In the latter algorithm, the exit position at  $I_3$  is also dealt with by the explanations in the above Case 1, while Case 2 is not needed at all.

We can conclude that the meeting itself does not contribute to a better runtime of the algorithm. But it *does* limit the algorithm by introducing a dependency between cut position and cut length. At first sight it might seem as if the improvement between the two algorithms presented in [8] simply comes from the shape of the cut and therefore a shortening of the pick-up distance. However, the possibility to find parameters for the algorithm with the triangular cut that give an improved evacuation time essentially comes from a decoupling of cut position and cut length. Yet there is still some correlation between cut position and cut length which is completely nullified in our algorithm  $A(y, \alpha, d)$ .

### 3.2 The Evacuation Time for $y = 2.62843$ , $\alpha = \pi/4$ and $d = 0.48793$

In this section, we show an evacuation time of 5.625 for Algorithm  $A(y, \alpha, d)$  for the parameters  $y = 2.62843$ ,  $\alpha = \pi/4$  and  $d = 0.48793$ .<sup>6</sup> To do so we determine, for each of the four segments defined in Section 3.1, the potential candidates for the worst-case exit position and then take the maximum over the evacuation times for those exits positions. For determining these candidates we use our findings from Section 2. To this end, for any pair (exit position, pick-up point), let  $\beta$  and  $\gamma$  denote the same angles as in Section 2, i.e.,  $\beta$  is the angle between the direction of movement of robot  $R_1$  at the (potential) exit position and the line connecting exit position and pick-up point and  $\gamma$  the angle between this line and the direction of movement of  $R_2$  at the (potential) pick-up point. Note that when both robots move along the perimeter, we have  $\beta = \gamma$  for reasons of symmetry.

---

<sup>6</sup>These parameters are chosen in a way that for the (only) three possible global worst-case exit positions (which we determine in the following), the evacuation times are the same up to numerical precision. While the values of the parameters were determined numerically, we give a rigorous proof for the correctness of the claimed evacuation time for the presented algorithm.



For calculating the distances of two points along the perimeter or along a straight line, observe that for any two points on the perimeter with a distance of  $w$  along the perimeter, their euclidian distance is  $2 \sin(w/2)$ . Note that the correctness of the latter does not depend on the choice of the way along the perimeter for determining  $w$ —going clockwise or counterclockwise yields the same value for  $2 \sin(w/2)$ . For instance, the value of  $|\widehat{AI_1}|$  is equal to the solution of the equation  $x + 2 \sin((x + y)/2) = y$ , where in our case  $y = 2.62843$ . For the given parameters, we obtain  $|\widehat{AI_1}| \approx 0.63196$ ,  $|\widehat{AI_2}| \approx 2.5837$  and  $|\widehat{AI_3}| = 2.62843$ . Now, we examine the four segments one by one.

**Lemma 3.1.** *If there is a (global) worst-case exit position in the first segment, then this exit position is at  $I_1$ .*

*Proof.* Recall that for each possible exit position in the first segment,  $R_2$  is picked up before it starts its cut. For the exit position at  $I_1$ , we obtain  $\gamma = \beta \approx 1.5114$ . It follows that  $2 \cos \beta + \cos \gamma < 1$  and because of the monotonicity of the cosine function between  $0$  and  $\pi$ , the same holds for any potential exit position that is an inner point on  $\widehat{AI_1}$ . Since we are in the conform case, we can apply Corollary 2.5 and obtain that, in the first segment, the only possible candidates for the worst-case exit position are  $A$  and  $I_1$ . Note that we cannot apply Corollary 2.5 to these two points since the movement of  $R_1$  is not differentiable at  $A$ , resp. the movement of  $R_2$  at the pick-up point corresponding to  $I_1$  is not differentiable. Also,  $A$  is obviously not a worst-case placement for the exit since both robots would find the exit together after only 1 time unit. Thus, the only candidate for a worst-case exit position in the first segment is  $I_1$ .  $\square$

**Lemma 3.2.** *If there is a (global) worst-case exit position in the second segment, then this exit position is at  $I_1$  or  $I_2$ .*

*Proof.* Recall that for each possible exit position in the second segment,  $R_2$  is picked up on its cut. Let  $P$  be the point at the tip of the cut and  $Q$  the intersection of the perimeter and the line through  $P$  and  $B$  that is not  $B$  itself. Hence,  $P$  lies on  $BQ$ . Moreover, let  $S$  be the exit position on  $R_1$ 's half of the perimeter for which  $R_2$  is picked up at  $P$ . For an illustration, we refer to Figure 6.

In order to simplify the analysis, we choose an underlying coordinate system for the disk such that  $M = (0, 0)$ ,  $A = (0, 1)$  and  $R_1$  explores the half of the perimeter with negative first coordinate. We obtain  $B \approx (0.49094, -0.8712)$ ,  $P \approx (0.14592, -0.52168)$ ,  $I_1 \approx (-0.59073, 0.80687)$ ,  $Q \approx (-0.8712, 0.49094)$ ,  $S \approx (-0.85568, -0.5175)$  and  $I_2 \approx (-0.5294, -0.84837)$ .

We consider first the case where the exit lies on  $\widehat{I_1Q}$ . In this case,  $R_2$  is picked up while it is on its way from  $B$  to  $P$  since  $S$  has a smaller second coordinate than  $Q$ , i.e., since  $Q$  is explored by  $R_1$  before  $S$ . For any possible exit on  $\widehat{I_1Q}$ , the respective  $\beta$  is at most as large as the  $\beta$  for the exit position at  $I_1$ , which is approximately 1.5114. Thus, for any possible exit on  $\widehat{I_1Q}$ , we obtain  $2 \cos \beta > 0.11872$ . On the other hand, again for any possible exit on  $\widehat{I_1Q}$ , the  $\gamma$  at the pick-up point is smaller than  $\angle I_1PQ \approx 0.28056$ . We obtain (for every possible  $\gamma$  in this case)  $\cos \gamma > 0.96089$  which yields  $2 \cos \beta + \cos \gamma > 1$ .

Now, we consider the case where the exit lies on  $\widehat{QS}$ . Similarly to the last case, we can bound any possible  $\beta$  from above by the  $\beta$  for the exit position at

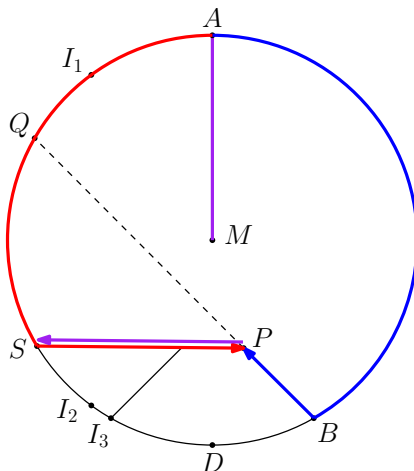


Figure 6: Illustration of Algorithm  $A(y, \alpha, d)$  with our chosen parameters, for the exit position at  $S$ . Note that  $SP$  and  $I_3B$  only appear to be parallel.

$Q$ , which is approximately 1.29856. Thus,  $2 \cos \beta > 0.53777$ . On the other hand, for any possible  $\gamma$ , we have  $\gamma \leq \angle QPS \approx 0.77673$  which yields  $\cos \gamma > 0.7132$ . We obtain  $2 \cos \beta + \cos \gamma > 1$ , for any exit position on  $\widehat{QS}$ .

Lastly, consider the case where the exit lies on  $\widehat{SI_2}$ . In this case, we want to show that  $2 \cos \beta + \cos \gamma < 1$ . Let  $X$  be an arbitrary point on  $\widehat{SI_2}$  and  $Y$  an arbitrary point on  $PB$ . Let  $\beta'$  be the angle between the direction of movement of  $R_1$  at  $X$  and  $XY$  and let  $\gamma'$  be the angle between  $XY$  and the direction of movement of  $R_2$  on the way from  $P$  to  $B$ .<sup>7</sup>

Assume for the moment that  $Y \neq B$ . We claim that moving  $Y$  a small  $\varepsilon$  towards  $B$  along  $PB$  increases the value of  $2 \cos \beta' + \cos \gamma'$ . In order to prove this claim, we start by observing that moving  $Y$  in the aforementioned way decreases  $\beta'$  by the same amount by which  $\gamma'$  is increased. By making  $\varepsilon$  arbitrarily small, the decrease in  $\beta'$  gets arbitrarily close to a value that is proportional to the derivative of the function  $-2 \cos$  at  $\beta'$  and the increase in  $\gamma'$  similarly gets arbitrarily close to a value that is proportional to the derivative of the function  $\cos$  at  $\gamma'$ . In other words,  $2 \cos \beta' + \cos \gamma'$  gets larger by the aforementioned movement of  $Y$  if  $2 \sin \beta' - \sin \gamma' > 0$ .

It is straightforward to verify that the choice for  $X$  and  $Y$  for which  $2 \sin \beta'$  is minimized is  $X = I_2$  and  $Y = B$ .<sup>8</sup> Similarly, the choice for  $X$  and  $Y$  that maximizes  $\sin \gamma'$  is  $X = I_2$ ,  $Y = P$ . Calculating the corresponding values for  $\beta'$  and  $\gamma'$ , we obtain  $2 \sin \beta' - \sin \gamma' > 1.02058 - 0.94268 = 0.0779$ , for any choice of  $X$  and  $Y$ . This proves the claim.

Thus, in order to show that  $2 \cos \beta + \cos \gamma < 1$  for any exit position on  $\widehat{SI_2}$ , it is sufficient to show that for any choice of  $X$  and  $Y = B$ , it holds that  $2 \cos \beta' + \cos \gamma' < 1$ . Hence, let  $Y = B$ . Similarly to before, it is straightforward

<sup>7</sup>Definitionwise, the angles  $\beta'$  and  $\gamma'$  correspond to  $\beta$ , resp.  $\gamma$ , for the (hypothetical) case where we consider  $X$  as the exit position and  $Y$  as the pick-up point.

<sup>8</sup>For this purpose, we allow  $Y$  to be  $B$  which we disallowed before. As any other choice for  $Y$  on  $PB$  results in a larger value for  $2 \sin \beta'$ , choosing  $X = I_2$  and  $Y = B$  gives a lower bound for  $2 \sin \beta'$ .

to show that both  $\beta'$  and  $\gamma'$  are minimized for  $Y = I_2$ . Thus  $2 \cos \beta' + \cos \gamma'$  is maximized<sup>9</sup> for  $Y = I_2$ . By calculating the values for  $\beta'$  and  $\gamma'$  for  $X = I_2$  and  $Y = B$ , we obtain that  $2 \cos \beta' + \cos \gamma' < 1.72000 - 0.72274 = 0.99726$ , for all choices of  $X$  on  $\widehat{SI_2}$ . This concludes the case where the exit lies on  $\widehat{SI_2}$ .

In all three considered subcases for the second segment, we therefore obtain either  $2 \cos \beta + \cos \gamma > 1$  or  $2 \cos \beta + \cos \gamma < 1$ . By applying Corollary 2.5, Theorem 2.8 and Theorem 2.10, it follows that a (global) worst-case exit can only be located at the points where the movement of at least one robot at the point itself or the corresponding pick-up point is not differentiable. Thus, the only three candidates are  $I_1$ ,  $S$  and  $I_2$ .

For the candidate  $S$ , recall the remark after Theorem 2.10. The problem with the non-differentiability with respect to determining that an exit is not worst-case placed is the following: In the proofs of the results in Section 2, we show that an exit is not worst-case placed by finding a close-by exit position for which it holds that the exit position and the corresponding pick-up point are visited by robots immediately before or after this would happen for the initial exit placement. Now, if the movement of one robot is not differentiable at the initial exit or the corresponding pick-up point, then this visiting condition is not necessarily satisfied. However, figuratively speaking, if the direction in which the exit has to be moved in order to get worse is not affected by the non-differentiability, then the results from Section 2 can still be applied. This is the case for  $S$ : From  $P$ ,  $R_2$  moves towards  $B$  on  $PB$  and this is exactly the direction in which the pick-up point moves when moving the exit at  $S$  along  $\widehat{SI_2}$  to a worse position. Note that moving the exit at  $S$  along  $\widehat{SI_2}$ , i.e., in the direction of the movement of  $R_1$ , indeed makes the exit worse according to the remarks after Theorem 2.10 since we are in the case  $2 \cos \beta + \cos \gamma < 1$ . It follows that in the second segment, the only candidates for worst-case exit positions are  $I_1$  and  $I_2$ .  $\square$

**Lemma 3.3.** *If there is a (global) worst-case exit position in the third segment, then this exit position is at  $I_2$  or  $I_3$ .*

*Proof.* Recall that for each possible exit position in the third segment,  $R_2$  is picked up after its cut. Thus, as in the first segment, we have  $\gamma = \beta$  for any exit position on  $\widehat{I_2I_3}$ . For the exit position at  $I_2$ , we have  $\gamma = \beta \approx 0.53553$ .<sup>10</sup> Since  $\beta$  and  $\gamma$  are decreasing if we move the exit from  $I_2$  along  $\widehat{I_2I_3}$ ,  $2 \cos \beta + \cos \gamma$  takes on its minimum if the exit is placed at  $I_2$ . Using the values for  $\beta$  and  $\gamma$  for this case determined above, we obtain that  $2 \cos \beta + \cos \gamma > 2.57999 > 1$ , for all exit positions on  $\widehat{I_2I_3}$ . Therefore, again, the only candidates for a worst-case placed exit are the non-differentiable ones, i.e.,  $I_2$  and  $I_3$ .  $\square$

For our examination of the fourth segment, we add a virtual point  $I'_3$  to the fourth segment that coincides with  $I_3$ , but has the additional property that if the exit is at  $I'_3$ , then  $R_1$  will only find the exit after performing its cut. The reason for this is that without the addition of  $I'_3$  the fourth segment is half-open which makes it possible that there is a sequence of exit positions with increasing

<sup>9</sup>Note that the cosine function is monotonically decreasing on the interval  $[0, \pi]$ .

<sup>10</sup>To be precise, the movement of  $R_2$  at the pick-up point  $B$  is not differentiable. Thus, for the definition of  $\gamma$ , we choose for the direction of movement of  $R_2$  at  $B$  the direction along the perimeter, not along the cut.

evacuation times that converges towards  $I_3$  and for which there is no exit position that has a larger evacuation time than all exit positions in the sequence. Thus, we add  $I'_3$  in order to obtain a closed segment.

**Lemma 3.4.** *If there is a (global) worst-case exit position in the fourth segment, then this exit position is at  $I'_3$ .*

*Proof.* With an argumentation analogous to the one for the third segment, we obtain  $2 \cos \beta + \cos \gamma > 1$  for all possible exit positions on  $I'_3 D$ . Using our differentiability argument again, we see that the only possible worst-case exit positions are  $I'_3$  and  $D$ . A simple calculation shows that the evacuation time of  $1 + \pi$  for the exit placement at  $D$  is much smaller than the evacuation time for the exit placement at, e.g.,  $I_1$ . Therefore, the only candidate for a worst-case placed exit is  $I'_3$ .  $\square$

Observe that the evacuation time for the exit at  $I'_3$  cannot be smaller than the evacuation time for the exit at  $I_3$  since in the latter case  $R_1$  could just simulate the former case which is worse than activating the meeting protocol right away. Combining this observation with Lemmas 3.1–3.4, we obtain the following theorem:

**Theorem 3.5.** *For  $y = 2.62843$ ,  $\alpha = \pi/4$  and  $d = 0.48793$ , the worst-case exit placement for Algorithm  $A(y, \alpha, d)$  is at  $I_1$ ,  $I_2$  or  $I'_3$ .*

In order to determine the evacuation time for the worst-case exit, we simply take the maximum of the evacuation times for the exit placements at these three candidate locations. We obtain evacuation times of approximately 5.6249, 5.62488 and 5.62491 for  $I_1$ ,  $I_2$  and  $I'_3$ , respectively. Hence, the evacuation time for the worst-case exit is approximately 5.62491. Thus, we obtain the following corollary:

**Corollary 3.6.** *For  $y = 2.62843$ ,  $\alpha = \pi/4$  and  $d = 0.48793$ , the evacuation time of Algorithm  $A(y, \alpha, d)$  is at most 5.625.*

Observe that, if the length of the cut is not changed, then altering the shape or angle of the cut does not affect the evacuation times for the exit positions at  $I_1$ ,  $I_2$  and  $I'_3$ . Thus, by Theorem 3.5, in such a case the overall evacuation time cannot decrease. We cast this insight into the following corollary:

**Corollary 3.7.** *For  $y = 2.62843$ ,  $\alpha = \pi/4$  and  $d = 0.48793$ , the evacuation time of Algorithm  $A(y, \alpha, d)$  cannot be improved by altering the shape or angle of the cut.*

Since all the inequalities in Lemmas 3.1–3.4 are not sharp, we can choose  $\alpha$  in some reasonable range without compromising our evacuation time.<sup>11</sup> As long as the chosen  $\alpha$  ensures that there is no worst-case exit placement such that  $R_2$  is picked up on the cut (without the start and end points of the cut), Corollary 3.6 holds. The value of exactly  $\pi/4$  for  $\alpha$  is chosen for the reason of convenience.

<sup>11</sup>The same holds for the shape of the cut, by the same reason.

## 4 Conclusion

In this paper, we studied the evacuation of two robots from a disk using non-wireless communication. We presented a new tool for the analysis of evacuation algorithms for any area shape by showing that a strong local condition has to be satisfied in order for an exit to be worst-case placed. Using this tool and further insights, e.g., about the nature of forced meetings and the irrelevance of the chosen shape and angle in some range, we improved the state-of-the-art algorithm and gave indicators for where to look (and where not) for further improvement.

However, we believe that our improved upper bound on the evacuation time is already very close to the tight bound that is the correct answer. We do not believe that our upper bound is optimal (up to numerical precision) because of the following reason: Imagine an additional second cut of very small depth (“ $\varepsilon$ -cut”) close to the point opposite of the point on the perimeter where the robots start their search. If we choose the position (and the angle and depth) of this  $\varepsilon$ -cut appropriately, then the evacuation time for the exit at  $I'_3$  will be improved since  $R_1$  will pick  $R_2$  up at around the tip of the  $\varepsilon$ -cut which is somewhat closer to  $I'_3$  than if there was no such  $\varepsilon$ -cut. Now we can make small changes to position and depth of the first cut that result in improving the evacuation times for the exit positions at  $I_1$  and  $I_2$  while increasing the previously decreased evacuation time for the exit position at  $I'_3$ . By finding the parameters that again lead to equal evacuation times for these three exits, the overall evacuation time is improved. If one is careful not to let other points become worse exit positions, this approach can even be applied iteratively. However, the improvement in the evacuation time achieved by the collection of these very small cuts is negligibly small, even compared to the improvement given by our algorithm.

While the lower bound is still a long way from our upper bound, it is hard to imagine how an improvement to our algorithm apart from the  $\varepsilon$ -cuts might look like. In fact, we conjecture that, apart from these  $\varepsilon$ -cuts and numerical precision, the algorithm we presented is indeed optimal.

## References

- [1] Steve Alpern. The rendezvous search problem. *SIAM Journal on Control and Optimization*, 33(3):673–683, 1995.
- [2] Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Rawlins. Searching with uncertainty (extended abstract). In *SWAT 88, 1st Scandinavian Workshop on Algorithm Theory, Halmstad, Sweden, July 5-8, 1988, Proceedings*, pages 176–189, 1988. URL: [http://dx.doi.org/10.1007/3-540-19487-8\\_20](http://dx.doi.org/10.1007/3-540-19487-8_20), doi:10.1007/3-540-19487-8\_20.
- [3] Anatole Beck and DJ Newman. Yet more on the linear search problem. *Israel journal of mathematics*, 8(4):419–429, 1970.
- [4] Piotr Borowiecki, Shantanu Das, Dariusz Dereniowski, and Lukasz Kuszner. Distributed evacuation in graphs with multiple exits. *SIROCCO 2016*.
- [5] Marek Chrobak, Leszek Gasieniec, Thomas Gorry, and Russell Martin. Group search on the line. In *SOFSEM 2015: Theory and*

- Practice of Computer Science - 41st International Conference on Current Trends in Theory and Practice of Computer Science, Pec pod Sněžkou, Czech Republic, January 24-29, 2015. Proceedings*, pages 164–176, 2015. URL: [http://dx.doi.org/10.1007/978-3-662-46078-8\\_14](http://dx.doi.org/10.1007/978-3-662-46078-8_14), doi:10.1007/978-3-662-46078-8\_14.
- [6] Jurek Czyzowicz, Stefan Dobrev, Konstantinos Georgiou, Evangelos Kranakis, and Fraser MacQuarrie. Evacuating two robots from multiple unknown exits in a circle. In *Proceedings of the 17th International Conference on Distributed Computing and Networking, Singapore, January 4-7, 2016*, pages 28:1–28:8, 2016. URL: <http://doi.acm.org/10.1145/2833312.2833318>, doi:10.1145/2833312.2833318.
- [7] Jurek Czyzowicz, Leszek Gasieniec, Thomas Gorry, Evangelos Kranakis, Russell Martin, and Dominik Pajak. Evacuating robots via unknown exit in a disk. In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, pages 122–136, 2014. URL: [http://dx.doi.org/10.1007/978-3-662-45174-8\\_9](http://dx.doi.org/10.1007/978-3-662-45174-8_9), doi:10.1007/978-3-662-45174-8\_9.
- [8] Jurek Czyzowicz, Konstantinos Georgiou, Evangelos Kranakis, Lata Narayanan, Jaroslav Opatrny, and Birgit Vogtenhuber. Evacuating robots from a disk using face-to-face communication (extended abstract). In *Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings*, pages 140–152, 2015. URL: [http://dx.doi.org/10.1007/978-3-319-18173-8\\_10](http://dx.doi.org/10.1007/978-3-319-18173-8_10), doi:10.1007/978-3-319-18173-8\_10.
- [9] Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil M. Shende. Wireless autonomous robot evacuation from equilateral triangles and squares. In *Ad-hoc, Mobile, and Wireless Networks - 14th International Conference, ADHOC-NOW 2015, Athens, Greece, June 29 - July 1, 2015, Proceedings*, pages 181–194, 2015. URL: [http://dx.doi.org/10.1007/978-3-319-19662-6\\_13](http://dx.doi.org/10.1007/978-3-319-19662-6_13), doi:10.1007/978-3-319-19662-6\_13.
- [10] Anders Dessmark, Pierre Fraigniaud, and Andrzej Pelc. Deterministic rendezvous in graphs. In *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, pages 184–195, 2003. URL: [http://dx.doi.org/10.1007/978-3-540-39658-1\\_19](http://dx.doi.org/10.1007/978-3-540-39658-1_19), doi:10.1007/978-3-540-39658-1\_19.
- [11] Yuval Emek, Tobias Langner, David Stolz, Jara Uitto, and Roger Wattenhofer. How many ants does it take to find the food? In *Structural Information and Communication Complexity - 21st International Colloquium, SIROCCO 2014, Takayama, Japan, July 23-25, 2014. Proceedings*, pages 263–278, 2014. URL: [http://dx.doi.org/10.1007/978-3-319-09620-9\\_21](http://dx.doi.org/10.1007/978-3-319-09620-9_21), doi:10.1007/978-3-319-09620-9\_21.
- [12] Ofer Feinerman and Amos Korman. Memory lower bounds for randomized collaborative search and implications for biology. In *Distributed Computing - 26th International Symposium, DISC 2012, Salvador, Brazil, October 16-18,*

2012. *Proceedings*, pages 61–75, 2012. URL: [http://dx.doi.org/10.1007/978-3-642-33651-5\\_5](http://dx.doi.org/10.1007/978-3-642-33651-5_5), doi:10.1007/978-3-642-33651-5\_5.
- [13] Ofer Feinerman, Amos Korman, Zvi Lotker, and Jean-Sébastien Sereni. Collaborative search on the plane without communication. In *ACM Symposium on Principles of Distributed Computing, PODC '12, Funchal, Madeira, Portugal, July 16-18, 2012*, pages 77–86, 2012. URL: <http://doi.acm.org/10.1145/2332432.2332444>, doi:10.1145/2332432.2332444.
- [14] Klaus-Tycho Förster, Rijad Nuridini, Jara Uitto, and Roger Wattenhofer. Lower bounds for the capture time: Linear, quadratic, and beyond. In *Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings*, pages 342–356, 2015. URL: [http://dx.doi.org/10.1007/978-3-319-25258-2\\_24](http://dx.doi.org/10.1007/978-3-319-25258-2_24), doi:10.1007/978-3-319-25258-2\_24.
- [15] Klaus-Tycho Förster and Roger Wattenhofer. Directed graph exploration. In *Principles of Distributed Systems, 16th International Conference, OPODIS 2012, Rome, Italy, December 18-20, 2012. Proceedings*, pages 151–165, 2012. URL: [http://dx.doi.org/10.1007/978-3-642-35476-2\\_11](http://dx.doi.org/10.1007/978-3-642-35476-2_11), doi:10.1007/978-3-642-35476-2\_11.
- [16] Pierre Fraigniaud, David Ilcinkas, Guy Peer, Andrzej Pelc, and David Peleg. Graph exploration by a finite automaton. In *Mathematical Foundations of Computer Science 2004, 29th International Symposium, MFCS 2004, Prague, Czech Republic, August 22-27, 2004, Proceedings*, pages 451–462, 2004. URL: [http://dx.doi.org/10.1007/978-3-540-28629-5\\_34](http://dx.doi.org/10.1007/978-3-540-28629-5_34), doi:10.1007/978-3-540-28629-5\_34.
- [17] Evangelos Kranakis, Danny Krizanc, and Sergio Rajsbaum. Mobile agent rendezvous: A survey. In *Structural Information and Communication Complexity, 13th International Colloquium, SIROCCO 2006, Chester, UK, July 2-5, 2006, Proceedings*, pages 1–9, 2006. URL: [http://dx.doi.org/10.1007/11780823\\_1](http://dx.doi.org/10.1007/11780823_1), doi:10.1007/11780823\_1.
- [18] Ioannis Lamprou, Russell Martin, and Sven Schewe. Fast two-robot disk evacuation with wireless communication. In *Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27-29, 2016. Proceedings*, pages 1–15, 2016. URL: [http://dx.doi.org/10.1007/978-3-662-53426-7\\_1](http://dx.doi.org/10.1007/978-3-662-53426-7_1), doi:10.1007/978-3-662-53426-7\_1.
- [19] Nicole Megow, Kurt Mehlhorn, and Pascal Schweitzer. Online graph exploration: New results on old and new algorithms. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, pages 478–489, 2011. URL: [http://dx.doi.org/10.1007/978-3-642-22012-8\\_38](http://dx.doi.org/10.1007/978-3-642-22012-8_38), doi:10.1007/978-3-642-22012-8\_38.
- [20] Richard J. Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43(2-3):235–239, 1983. URL: [http://dx.doi.org/10.1016/0012-365X\(83\)90160-7](http://dx.doi.org/10.1016/0012-365X(83)90160-7), doi:10.1016/0012-365X(83)90160-7.

- [21] Torrence D Parsons. Pursuit-evasion in a graph. In *Theory and applications of graphs*, pages 426–441. Springer, 1978.