Wireless Evacuation on m Rays with k Searchers

Sebastian Brandt[1], Klaus-Tycho Foerster[2], Benjamin Richner[1], and Roger Wattenhofer[1]

[1] ETH Zürich, Switzerland,
brandts@ethz.ch, benri@bluewin.ch, wattenhofer@ethz.ch,
[2] Aalborg University, Denmark,
ktfoerster@cs.aau.dk

Abstract. We study the online problem of evacuating k robots on m concurrent rays to a single unknown exit. All k robots start on the same point s, not necessarily on the junction j of the m rays, move at unit speed, and can communicate wirelessly. The goal is to minimize the competitive ratio, i.e., the ratio between the time it takes to evacuate all robots to the exit and the time it would take if the location of the exit was known in advance, on a worst-case instance.
When $k = m$, we show that a simple waiting strategy yields a competitive ratio of 4 and present a lower bound of $2 + \sqrt{7/3} \approx 3.52753$ for all $k = m \geq 3$. For $k = 3$ robots on $m = 3$ rays, we give a class of parametrized algorithms with a nearly matching competitive ratio of $2+\sqrt{3} \approx 3.73205$. We also present an algorithm for $1 < k < m$, achieving a competitive ratio of $1+2\cdot\frac{m-1}{k}\cdot\left(1+\frac{k}{m-1}\right)^{1+\frac{m-1}{k}}$, obtained by parameter optimization on a geometric search strategy. Interestingly, the robots can be initially oblivious to the value of $m > 2$.
Lastly, by using a simple but fundamental argument, we show that for $k < m$ robots, no algorithm can reach a competitive ratio better than $3 + 2\lfloor (m-1)/k \rfloor$, for every k, m with $k < m$.

1 Introduction

Searching for an unknown target is a fundamental problem in computer science and mathematics, especially in the area of robotics. The standard toolkit to analyze this class of problems is competitive analysis [32], i.e., our goal is to design online algorithms with a small competitive ratio, which compares the performance of the online algorithm to an optimal offline solution which knows the target location beforehand.

As pointed out by Hammar et al. [22], "*A problem with paradigmatic status in this framework is searching on m concurrent rays*," which is the focus of this paper. More precisely, we study the problem of evacuating $k \leq m$ robots on m concurrent rays (i.e., semi-infinite lines) to an unknown exit z [23,26], with the robots communicating wirelessly [14,18].

The seminal forefather of this problem is the linear search problem, also known as the cow path problem, first posed by Beck [6] and Bellman [8]: A searcher has to find an object of unknown location on the infinite line (i.e., 2 concurrent rays). The optimal online algorithm achieves a competitive ratio of 9, in each iteration doubling the search depth $1, 2, 4, \ldots$ on each side of the starting point s [7]. Gal [21] and Baeza-Yates et al. [4] then extended their results to the model of m concurrent rays, where the optimal strategy is to, instead of doubling the search depth, use a factor of $m/(m-1)$, yielding an optimal competitive ratio of $1 + 2m^m/(m-1)^{m-1}$ [29]. If k robots can search for the exit, and one robot finding it terminates the search, a competitive ratio of $1 + 2(m/k - 1)/(m/(m-k))^{m/k}$ is optimal [30].

The concepts of collaborative evacuation and wireless communication are more recent additions in this field. In the case of the (unit speed) robots only being able to communicate when they meet, for $k = m$ a competitive ratio of 9 is again optimal [23] if there is a minimum distance to the exit, else $1 + 2(p+1)^{p+1}/p^p$ for $p = \lceil \log m \rceil$ is optimal. In the special case of $m = 2$ and $k > m$, 9 is optimal as well [12]. Baeza-Yates and Schott studied wireless communication in this context: Even though most of their paper "Parallel searching in the plane" [5] is about searching the plane, they also considered the evacuation problem with two searchers on the line, pointing out that a competitive ratio of 3 is then optimal for $k \geq m = 2$. Further collaborative robot evacuation studies in geometric settings have been performed by Czyzowicz et al.: Evacuating the circle with $k = 2$ [13], the line with faulty robots [17], the disk [14,15,16] (see also [11]), and equilateral triangles and squares [18], with [15,18] also studying wireless communication.

Contributions In this paper, we extend the model of Baeza-Yates and Schott [5] beyond the infinite line (i.e., $m = 2$), by examining the problem of evacuating $1 < k \leq m$ robots on m rays with wireless communication, which has not been studied before to the best of our knowledge. We also study the case that the k robots do not start on the junction j of the m rays.

When starting on the junction with $k = m > 2$ robots, we show that a competitive ratio of 3 is still optimal, and starting away from the junction allows for a 4-competitive algorithm. For the special case of $k = m = 3$, we present a class of parametrized algorithms with a competitive ratio of $2 + \sqrt{3} \approx 3.73205$. We also give lower bounds of $2 + \sqrt{7/3} \approx 3.52753$, for every $k = m \geq 3$.

Furthermore, we consider the case of less robots than rays, i.e., collaborative wireless evacuation with $1 < k < m$ robots. Even though the k robots are oblivious to the number of $m > 2$ rays, our optimization of parametrized geometric search strategy yields a competitive ratio of at most $1 + 2 \cdot \frac{m-1}{k} \cdot \left(1 + \frac{k}{m-1}\right)^{1 + \frac{m-1}{k}}$. Moreover, as we show, even when starting on the junction, no algorithm can have a better competitive ratio than $3 + 2\lfloor (m-1)/k \rfloor$, for any k, m with $k < m$.

Paper Organization In the following paragraph we discuss further related work, before introducing the necessary formal preliminaries in Section 2. We then consider the case of m robots on m rays in Section 3, with an in-depth focus of 3 robots on 3 rays. Afterwards, we study the more general case of $1 < k < m$

robots on m rays in Section 4, also detailing a lower bound for $k < m$ with a simple but fundamental argument. Lastly, we conclude in Section 5.

Further Related Work Results for the search problem on m rays can be used for showing competitive bounds for search problems in various classes of simple polygons, cf. [23,29], with further applications in hybrid [26] and interruptible [1] algorithms. The classic linear search or cow path problem has moreover been studied in a multitude of models, e.g., adding turn costs [9,19] (also with multiple searchers on rays [2]), with a single [25] or multiple error prone robots [17], or a moving target [9]. Bose et al. [10] gave tight bounds on the competitive ratio with distance bounds to the target, showing that the optimal search strategy is then unique.

Searching on m rays has furthermore been considered with multiple targets [3], with only one robot being allowed to move at a time [26], regarding advice complexity [24], and randomized algorithms [27,31] – cf. the survey by Tate [33] for an overview of the latter.

On graphs, the problem of finding a specified node in an online fashion is also known as treasure hunt or as the node searching problem. [20,28].

2 Preliminaries

We consider the problem of collaboratively evacuating k robots $R_0, \ldots, R_{k-1}$ on m concurrent rays $a_0, \ldots, a_{m-1}$, joined at a common junction j. All robots start at the same point s, w.l.o.g. on ray a_0, where s does not have to be the junction j. All robots have to reach the single exit z on some ray a_z, the location and ray of z is unknown until one robot reaches the location of the exit z. We denote the distance of the junction j to the start s by $\overline{js}$. The robots have the same unit maximum speed and can communicate wirelessly, instantaneously sharing their information. As thus, we can assume that one central algorithm controls all robots. Unless otherwise noted, we assume that the robots travel at unit speed when moving.

The goal is to minimize the time needed for all robots to reach the exit, compared to the minimum time needed if all information about the environment would be revealed initially. Hence, we study this problem using competitive analysis: The competitive ratio of an online (evacuation) algorithm is measured as the supremum of the ratio of the time needed for all robots to reach the exit and the distance Z from s to z, for all start and exit locations.

If the distance between the start s and the exit z is allowed to be arbitrarily small, no online algorithm (without infinitesimal steps) can achieve a constant competitive ratio for $k < m$: As thus, we use the common assumption of at least unit distance between start s and exit z, cf. [1].

3 m Robots on m Rays

We start our study of robot evacuation by considering one robot for each ray. In Subsection 3.1 we gather some basic observations. Note that Observations 1

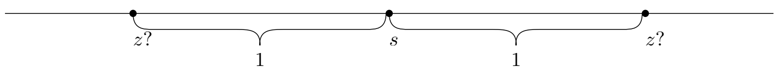

Fig. 1. As the robots starting on s are oblivious to the direction of the exit z, both points of distance 1 need to be explored by at least one robot, meaning that at least one robot takes a time of $t = 3$ to reach the exit (in this case all robots can also evacuate the graph at a time of $t = 3$).

and 2 can be found with similar arguments for $k = 2$ in [5]. We further examine the case of 3 robots on 3 rays in Subsection 3.2.

3.1 The General Case of m Robots on m Rays

If all m robots start *on* the junction j, then each robot R_i can explore ray a_i at unit speed, with some robot finding z at time $t = Z$. Then, all other robots are at distance $2Z$ from z, inducing a total evacuation time of $t = 3Z$ if they all directly travel to the exit. Trivially, in the case of $k = 1$, a single robot starting on the end of a single ray will find the exit in optimal time.

Observation 1 *Let $s = j$ and $k = m$, with $m > 1$. There exists an online algorithm evacuating the m robots with a competitive ratio of 3.*

For $m > 1$, no better ratio than 3 is possible (cf. also Figure 1): Assume all $2 \leq k \leq m$ robots start on the junction j and the exit is at distance $Z = 1$. In the worst case, the exit will be on the last ray explored until distance 1 (which could coincide with the first ray being explored until distance 1), so at least one robot will need a time of $t = 3Z$ to reach the exit z.

Observation 2 *For every $2 \leq k \leq m$: No online algorithm can achieve a better competitive ratio than 3 for evacuating the k robots.*

The situation is more difficult when the robots do not start on the junction j and $m > 2$.[3] If we knew the initial direction of the junction, we could send $m - 1$ robots there, again obtaining a competitive ratio of 3 as before.

The following algorithm yields an upper bound of 4 for the competitive ratio even when the direction of the junction is not known: Send two robots R_0, R_1 in opposing directions until either the exit z or the junction j is found, with the remaining $m - 2$ robots waiting at the start s. If the exit z is found first (or simultaneously), a competitive ratio of 3 can again be achieved by directly sending all robots to the exit z. If the junction is found first, we stop the robots R_0, R_1 for a duration of $\overline{js}$, while the other $m - 2$ robots travel to the junction. We then proceed as if s was the point from which all rays emanate and the section between s and j was actually comprised of the first parts of $m - 1$ rays that just

[3] If $m = 2$, then a competitive ratio of 3 can be reached again, as every point can be seen as the junction.

happened to be glued together. According to this equivalent consideration, at time $2\overline{js}$, all robots are on their rays at distance $\overline{js}$ from s and then continue to explore their assigned rays. When the exit z is found by one robot at time $\overline{js} + Z$, all other robots move to the exit z in time $2Z$, obtaining a competitive ratio of $(\overline{js} + 3Z)/Z < 4$.

Observation 3 *Let $k = m$, $m > 2$. There exists an online algorithm evacuating the m robots with a competitive ratio of at most 4.*

We will later show a lower bound of $2 + \sqrt{7/3} \approx 3.52753$ in Corollary 2, for all $k = m \geq 3$.

3.2 The Case of 3 Robots on 3 Rays

We start with a lower bound for the competitive ratio of evacuating 3 robots from 3 rays, before giving a nearly matching upper bound in Theorem 3.

Theorem 1 (Lower bound of $2+\sqrt{7/3}$ for 3 robots on 3 rays). *No online algorithm can achieve a better competitive ratio than $2 + \sqrt{7/3} \approx 3.52753$ for evacuating 3 robots on 3 rays.*

Proof. As evacuating 3 robots on 3 rays has a competitive ratio of 3 when $s = j$, we assume that $s \neq j$, $s \in a_0$, and $Z > \overline{js}$. Also, we can assume in a worst-case fashion that the junction j lies on the side of s that ensures that at time $\overline{js}$ at most one of the three robots is closer to j than in the beginning, i.e., closer to j than $\overline{js}$.

It follows that the earliest time when the 2 points of distance $3/2 \cdot \overline{js}$ from s on a_1, a_2 have been visited is at time $5/2 \cdot \overline{js}$: Only the robot that is (possibly) closer to j at time $\overline{js}$ than in the beginning can visit any of these 2 points before time $5/2 \cdot \overline{js}$; however, since it can visit the first of the two at time $3/2 \cdot \overline{js}$ at the earliest, it cannot visit the other one before time $5/2 \cdot \overline{js}$.

W.l.o.g., let R_2 be a robot who has (possibly previously) visited a point p farthest away from the junction on the starting ray a_0 at time $t = 5/2 \cdot \overline{js}$. We will now show Theorem 1 by case distinction for a point y, denoting where R_2 is at time $5/2 \cdot \overline{js}$. The case distinction will depend on a "border"-value b, later to be optimized. We refer to Figure 2 for an overview of the construction.

We start with the first case of $\overline{jy} \geq b + \overline{js}$ and y lies on a_0: Then, we place the exit z at one of the points of distance $3/2 \cdot \overline{js}$ from s on a_1, a_2 that is visited last by the strategy used by the three robots. As the exit cannot have been found before time $5/2 \cdot \overline{js}$, robot R_2 will need (in the best case) $5/2 \cdot \overline{js} + \overline{sy} + 3/2 \cdot \overline{js} = 4 \cdot \overline{js} + \overline{sy} \geq 4 \cdot \overline{js} + b$ total time to reach the exit z. Note that in this case, the optimal time is $Z = 3/2 \cdot \overline{js}$.

Next, we consider the second and remaining case of $\overline{jy} < b + \overline{js}$ or y not lying on a_0. To still reach y at time $5/2 \cdot \overline{js}$, R_2 could have moved at most to a p with $\overline{ps} \leq 5/4 \cdot \overline{js} + b/2$. We now place the exit z a distance of ε "behind" one of the three points of distance $5/4 \cdot \overline{js} + b/2$ to the start s which will be reached last. Note that, as shown before, the earliest time when both of these points on

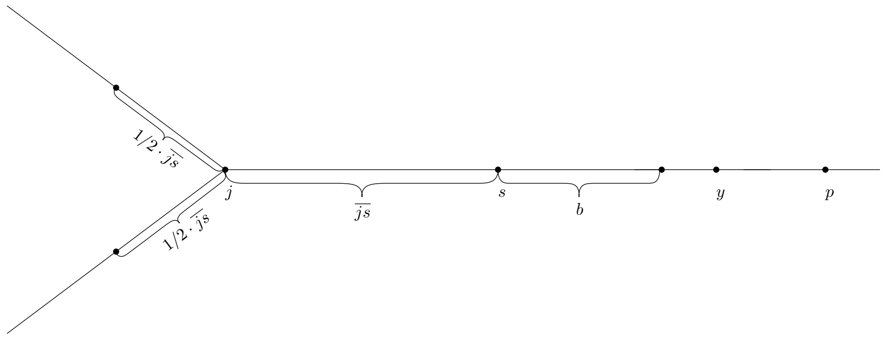

Fig. 2. The robots R_1, R_2, R_3 start on s and have to find the unknown exit z. Point p depicts the farthest any robot has been away from the junction on the starting ray a_0 until time $5/2 \cdot \overline{js}$, and y where a robot visiting p is at time $5/2 \cdot \overline{js}$. Depending on if y is at least $\overline{js} + b$ away from the junction j or not, we give two different arguments in the proof of Theorem 1, resulting in a (normalized to $\overline{js}$) value of b of $-1+\sqrt{21}/2 \approx 1.29129$ and a lower bound of $2 + \sqrt{7/3} \approx 3.52753$.

a_1, a_2 can be reached is at time $5/2 \cdot \overline{js} + b/2 + \varepsilon$, and the earliest time when the respective point on a_0 can be reached (assuming both points on a_1, a_2 were reached) is not before time $5/2 \cdot \overline{js} + 5/4 \cdot \overline{js} + b/2 + \varepsilon - b$. As thus, for all robots to evacuate to the exit, a time of at least $5/2 \cdot \overline{js} + (5/4 \cdot \overline{js} + b/2 + \varepsilon - 3/2) + 5/4 \cdot \overline{js} + b/2 + \varepsilon + 5/4 \cdot \overline{js} + b/2 + \varepsilon = 19/4 \cdot \overline{js} + 3/2 \cdot b + 3 \cdot \varepsilon$ is needed, with the optimal solution taking time $Z = 5/4 \cdot \overline{js} + b/2 + \varepsilon$.

To optimize the lower bound in respect to b, we solve $\frac{19/4 \cdot \overline{js} + 3/2 \cdot b + 3 \cdot \varepsilon}{5/4 \cdot \overline{js} + b/2 + \varepsilon} = \frac{4 \cdot \overline{js} + b}{3/2 \cdot \overline{js}}$ for b. By normalizing $\overline{js}$ to unit value and restricting $b > 0$, solving the above equation gives us the parameter $b = -1 - \epsilon + 1/2 \cdot \sqrt{21 + 12\varepsilon + 4\varepsilon^2}$, which is approximately 1.29129 for $\varepsilon \to 0$ for our proof, as the functions defined by the terms on the individual sides of the equation are monotonically decreasing and increasing, respectively.

Observe that $-1 - \epsilon + 1/2 \cdot \sqrt{21 + 12\varepsilon + 4\varepsilon^2}$ is monotonically decreasing when considered as a function of ε, i.e., for all values of $\varepsilon > 0$, we obtain the supremum at $-1 + \sqrt{21}/2 \approx 1.29129$.

Therefore, we achieve a lower bound of $\frac{4-1+\sqrt{21}/2}{3/2} = 2 + \sqrt{7/3} \approx 3.52753$. $\square$

We note that the construction from the above proof can be extended to $k = m > 3$ robots and rays, as at time $t = \overline{js}$, at most $\lfloor m/2 \rfloor$ robots can be guaranteed to be at the junction j.

Corollary 2. *For every $k = m \geq 3$ holds: No online algorithm can achieve a better competitive ratio than $2 + \sqrt{7/3} \approx 3.52753$ for evacuating $k = m$ robots on m rays.*

We now give an algorithm with a nearly matching competitive ratio for 3 robots:

Theorem 3. *There exists an online algorithm evacuating* 3 *robots on* 3 *rays with a competitive ratio of* $2+\sqrt{3}\approx 3.73205$.

Proof. We know from Observation 1 that there is an algorithm with a competitive ratio of 3 when starting on the junction j, so suppose that $j\neq s$. We prove Theorem 3 by giving a whole class of algorithms, all reaching the desired competitive ratio. To describe these strategies, we develop a parametrized approach by composing an algorithm that moves the robots according to certain parameters and then optimizing the competitive ratio over the parameter space. More specifically, the algorithm depends on two parameters α and β which are constrained by the inequalities $0\leq\beta\leq\alpha\leq\frac{1}{2}$ and $2\alpha\leq\beta+\frac{1}{2}$ and moves the three robots R_0,R_1,R_2 as described in the following. We note that if one robot finds the exit, all the other robots abandon their strategy and take the shortest path to the exit z.

Figure 3 serves as a visual aid to understand the parameters and the respective strategies. We send R_0 in one direction, R_1 in the other, and R_2 waits until the junction j (or the exit) is found. W.l.o.g. suppose R_0 reaches the junction j after $\overline{js}$ time passed, i.e., R_1 is at distance $\overline{js}$ to s on the other side of ray a_0, and R_2 is still on the start s. Then, R_0 moves for $\alpha\cdot\overline{js}$ time into one of the two branching rays a_1,a_2, returns back to the junction j, and moves into the other ray of a_1,a_2. Meanwhile, at time $\overline{js}$, R_1 starts to move deeper into the ray a_0 away from the junction j by $\beta\cdot\overline{js}$ before turning around and walking backwards until it reaches the same distance to the junction j as r_3, which starts moving towards the junction at time $\overline{js}$ and then moves into the ray R_0 explored first (and left by the time R_2 arrives at the junction). The three robots continue to move straight to a distance of $\overline{js}+\beta\cdot\overline{js}$ to s on their respective ray, and those that arrive early wait for the others. Then, they all move uniformly outwards at equal distance to the start s.

We will now start analyzing the competitive ratio of the above algorithm: Until the junction is found, any exit found will lead to a competitive ratio of 3. Observe that until all three robots move outwards from the start s on the three rays, the following three points, with additional ε distance to s, are worst case points regarding the competitive ratio of the algorithm (cf. Figure 3), i.e., the time when a robot visits them for the first time will determine the competitive ratio: p_2, the point where R_0 turns around to go back to the junction, p_1, the point where R_1 turns around to go back to the start, and p_0, the junction itself. After that, the competitive ratio of any exit placement can only be lower, as now any remaining distance of x to the exit will be covered in x time by one robot.

For ease of readability, we are going to omit the additional εs in the following calculations, as we are later going to consider the supremum of the competitive ratio anyway.

The three points will be reached at the following times: p_0 at time $\overline{js}+2\cdot\alpha\overline{js}$ by R_0, p_1 at time $2\cdot\overline{js}+\beta\overline{js}$ by R_1, and p_2 at time $2\cdot\overline{js}+\alpha\overline{js}$. If all other robots divert directly to the exit z when it is found, they will reach the exit with the following additional time: p_0 with time $2\cdot\overline{js}-2\cdot\alpha\overline{js}+2\cdot\beta\overline{js}$, p_1 with time $2\cdot\overline{js}+2\cdot\beta\overline{js}$, and p_2 with time $2\cdot\overline{js}+2\cdot\alpha\overline{js}$.

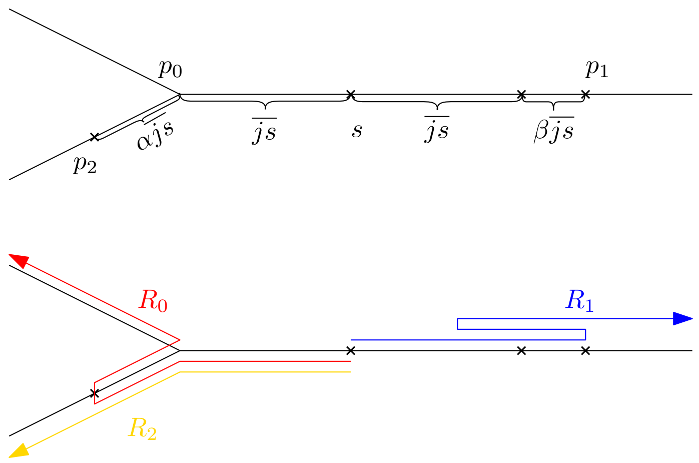

Fig. 3. A depiction of the parameters α and β on the 3-ray and the strategies of the 3 robots (waiting is not indicated). The three worst case points p_0, p_1, p_2 are also marked.

Hence, the competitive ratio induced by the three points is adding both times above, divided by the distance of the exit to the start, i.e.,: $3+2\cdot\beta$ for p_0, $3+\frac{1}{1+\beta}$ for p_1, and $3+\frac{1}{1+\alpha}$ for p_2. Note that $3+\frac{1}{1+\alpha} \leq 3+\frac{1}{1+\beta}$ due to initially choosing $\beta \leq \alpha$.

As $3+2\cdot\beta$ is strictly monotonically increasing and $3+\frac{1}{1+\beta}$ strictly monotonically decreasing, the desired solution can be obtained by equalizing both terms in the parameter range, with $\beta = \frac{\sqrt{3}-1}{2}$. As α can be chosen freely in the parameter space, we have generated a whole class of algorithms with identical competitive ratio of $\min_\beta \max\left(3+2\cdot\beta, 3+\frac{1}{1+\beta}\right) = 2+\sqrt{3}$. □

4 $1 < k < m$ Robots on m Rays

In this section, we continue our study of collaborative robot evacuation by considering the case of $1 < k < m$ robots on m rays. Since in this case the number of robots is not sufficient anymore to assign a ray to each robot, a more intricate scheme than before is required in order to achieve a good competitive ratio. In the literature, similar problems have been tackled by using geometric search. We show that this general idea can also be applied in our setting and present an upper bound for the competitive ratio where the factor that governs the exponential growth is chosen in a way that minimizes the bound. We complement this result with a lower bound for all wireless evacuation algorithms where $k < m$.

4.1 An Upper Bound on the Competitive Ratio

We start by developing GeomSearch(α, β), an algorithm for evacuating k robots from m rays where the robots start in the junction j. The algorithm depends on two parameters α and β which we will determine later. It proceeds as follows:

Each robot explores the m rays in so-called *exploration steps* where each exploration step consists of exploring some ray up to some depth and then returning to the junction. More specifically, robot R_i starts by exploring ray a_i up to depth $\alpha\beta^i$ upon which it returns to the junction. Then it explores ray $a_{i+k \pmod m}$ up to depth $\alpha\beta^{i+k}$, returns to the junction, explores ray $a_{i+2k \pmod m}$ up to depth $\alpha\beta^{i+2k}$, and so on. In other words, robot R_i performs its qth exploration step on ray $a_{i+(q-1)k \pmod m}$ with a depth of $\alpha\beta^{i+(q-1)k}$. Note that in each exploration step of robot R_i the explored depth increases by a factor of β^k and that it always chooses the ray to be explored next by increasing the ray index by k (modulo m). If a robot finds the exit z it immediately informs all other robots, upon which each robot immediately aborts its exploration and heads straight for z.

From the definition of the exploration steps it follows that for any two robots R_h and R_i with $h < i$ and any positive integer q, the qth exploration step of R_i takes strictly more time than the qth exploration step of R_h and the $(q+1)$th exploration step of R_h takes strictly more time than the qth exploration step of R_i. Thus, we obtain the following observation which sheds light on the order in which the robots take their exploration steps.

Observation 4 *Let h, i and q be integers satisfying $0 \le h < i \le k-1$ and $q \ge 1$. Then robot R_h finishes its qth exploration step before R_i finishes its qth exploration step, and R_i finishes its qth exploration step before R_h finishes its $(q+1)$th exploration step.*

In order to prove an upper bound on the competitive ratio of Algorithm GeomSearch(α, β) for suitably chosen α and β, we need a technical lemma, given in the following.

Lemma 4. *Let $\beta = \left(1 + \frac{k}{m-1}\right)^{1/k}$. Then $1 + 2\dfrac{\beta^{m+k-1}}{\beta^k - 1} \ge 3 + 2\dfrac{\beta^m}{\beta^k - 1}$.*

Proof. For a contradiction, assume that the statement is false. We obtain the following series of implications:

$$
\begin{aligned}
& 3 + 2\frac{\beta^m}{\beta^k - 1} > 1 + 2\frac{\beta^{m+k-1}}{\beta^k - 1} \\
\Longrightarrow \quad & \beta^k - 1 > \beta^m(\beta^{k-1} - 1) \\
\Longrightarrow \quad & \frac{k}{m-1} > \left(1 + \frac{k}{m-1}\right)^{m/k} \left(\left(1 + \frac{k}{m-1}\right)^{(k-1)/k} - 1\right) \\
\Longrightarrow \quad & \frac{k}{m-1} > \left(1 + \frac{m}{m-1}\right) \left(\left(1 + \frac{k}{m-1}\right)^{(k-1)/k} - 1\right)
\end{aligned}
$$

$$\begin{aligned}
\implies \quad & \frac{2m+k-1}{2m-1} > \left(\frac{m+k-1}{m-1}\right)^{(k-1)/k} \\
\implies \quad & \frac{2m-1}{2m+k-1} < \left(\frac{m-1}{m+k-1}\right)^{(k-1)/k} = \left(1+\frac{-k}{m+k-1}\right)^{(k-1)/k} \\
\implies \quad & \frac{2m-1}{2m+k-1} < 1+\frac{-(k-1)}{m+k-1} = \frac{m}{m+k-1} \\
\implies \quad & mk+1 < 2m+k
\end{aligned}$$

For the third and sixth implication we used the generalized version of Bernoulli's inequality which says that for any two real numbers $b > -1$ and $c \geq 0$ it holds that $(1+b)^c \geq 1+bc$ if $c \geq 1$, and $(1+b)^c \leq 1+bc$ if $0 \leq c \leq 1$. Since $m > k \geq 2$, the obtained statement implies $k = 2$. Going back to the result after the fourth implication and plugging in $k = 2$, we obtain the following new implications:

$$\frac{2m+1}{2m-1} > \left(\frac{m+1}{m-1}\right)^{1/2} \implies (2m+1)^2(m-1) > (2m-1)^2(m+1) \implies -1 > 1\,.$$

We obtain a contradication, which proves the lemma statement. □

Now we can finally prove the desired upper bound.

Theorem 5. *Let $\beta = \left(1+\frac{k}{m-1}\right)^{1/k}$ and let α be chosen such that $\alpha\beta^{m-1} < 1$. Then the competitive ratio of Algorithm GeomSearch(α,β) is at most*

$$1+2\cdot\frac{m-1}{k}\cdot\left(1+\frac{k}{m-1}\right)^{1+\frac{m-1}{k}}\quad.$$

Proof. Let R_h be the robot that finds the exit and assume that R_h finds the exit in its qth exploration step. It follows from the design of our algorithm that the exit lies on ray $a_{h+(q-1)k \pmod m}$. Note that since $\alpha\beta^{m-1} < 1$ and the exit has a distance of at least 1 from the junction, we have that $q \geq 2$. Let t_0 denote the point in time at which R_h reaches the point on $a_{h+(q-1)k \pmod m}$ with largest depth that has been explored before by some robot. Let Δt denote the time R_h travels on $a_{h+(q-1)k \pmod m}$ between t_0 and finding the exit, i.e., R_h finds the exit at time $t_0 + \Delta t$. Furthermore, for each R_i with $i \neq h$, let E_i denote the exploration step R_i is performing at the time when R_h starts its qth exploration step, and let $t_1(i)$ denote the point in time at which R_i finishes exploration step E_i. We note that at time t_0, the distance between R_h and the junction is the depth of the previous exploration step of a robot on ray $a_{h+(q-1)k \pmod m}$ which is $\alpha\beta^{h+(q-1)k-m}$, by the definition of the exploration steps.[4] Now, we consider two cases for each robot R_i:

[4] Here we implicitly use that $\alpha\beta^{m-1} < 1$ which ensures that the ray on which R_h finds the exit, has been previously explored by some robot.

First, consider the case that $t_0 \geq t_1(i)$. Then, at time t_0, R_i has finished exploration step E_i and has started with its next exploration step. This implies that the distance between R_i and the junction at time t_0 is smaller than the distance between R_h and the junction at time t_0, i.e., smaller than $\alpha\beta^{h+(q-1)k-m}$. Thus, at time $t_0 + \Delta t$, the distance between R_i and the junction is smaller than $\alpha\beta^{h+(q-1)k-m} + \Delta t$. We conclude that it takes R_i at most $2(\alpha\beta^{h+(q-1)k-m} + \Delta t)$ time to reach the exit after the exit has been found at time $t_0 + \Delta t$. Since R_h finishes its first $q-1$ exploration steps in time

$$\sum_{x=0}^{x=q-2} 2\alpha\beta^{h+xk} = 2\alpha\beta^h \sum_{x=0}^{x=q-2} \left(\beta^k\right)^x = 2\alpha\beta^h \frac{\beta^{(q-1)k} - 1}{\beta^k - 1}$$

and it takes R_h another $\alpha\beta^{h+(q-1)k-m} + \Delta t$ time to find the exit, we hence obtain an upper bound of

$$2\alpha\beta^h \frac{\beta^{(q-1)k} - 1}{\beta^k - 1} + 3(\alpha\beta^{h+(q-1)k-m} + \Delta t)$$

for the time it takes R_i to reach the exit.

In order to obtain an upper bound for the competitive ratio (for our first case), we divide by the length Z of the shortest path from s to z. Note that Δt appears with a factor of 3 in the numerator whereas it appears with a factor of 1 in the denominator. Since the competitive ratio we obtain is larger than 3, making Δt larger decreases the competitive ratio (towards 3). Hence, by setting $\Delta t = 0$, we obtain an upper bound of

$$\begin{aligned}\frac{2\alpha\beta^h \frac{\beta^{(q-1)k}-1}{\beta^k-1} + 3\alpha\beta^{h+(q-1)k-m}}{\alpha\beta^{h+(q-1)k-m}} &= 3 + 2\frac{\beta^{(q-1)k} - 1}{(\beta^k - 1)\,\beta^{(q-1)k-m}} \\ &= 3 + 2\frac{\beta^m}{\beta^k - 1} - \frac{2}{(\beta^k - 1)\,\beta^{(q-1)k-m}}\end{aligned}$$

for the competitive ratio, which implies an upper bound of $3 + 2\beta^m/(\beta^k - 1)$. Note that the last simplification does not increase the upper bound more than necessary: The term $2/((\beta^k - 1)\beta^{(q-1)k-m})$ can be made arbitrarily small by increasing q, i.e., by choosing the exit location accordingly.

Now consider the second case, namely, that $t_0 < t_1(i)$. Then, R_i is still performing exploration step E_i at time t_0. It follows that at the time the exit is found, R_i is still performing E_i or R_i has distance at most Δt from the junction. Thus, we can bound (from above) the total time it takes R_i to reach the exit by the sum of 1) the time it takes R_i to perform its exploration steps up to and including E_i, 2) two times Δt, which bounds the time between reaching the junction after E_i and reaching the junction possibly again after being told the location of the exit and 3) $\alpha\beta^{h+(q-1)k-m} + \Delta t$, the time it takes R_i to reach the exit from the junction. The first of the three summands in turn can be bounded by the time it takes $R_{h-1 \pmod m}$ to perform its exploration steps up to and including $E_{h-1 \pmod m}$, by the definition of E_i and Observation 4.[5] Hence, we obtain an upper bound of

$$\sum_{x=0}^{x=q-1} 2\alpha\beta^{h-1+xk} + 2\Delta t + \alpha\beta^{h+(q-1)k-m} + \Delta t$$
$$= 2\alpha\beta^{h-1}\frac{\beta^{qk}-1}{\beta^k-1} + \alpha\beta^{h+(q-1)k-m} + 3\Delta t$$

for the time it takes R_i to reach the exit. By an argumentation analogous to the one in the previous case, we obtain an upper bound of $1+2\beta^{m+k-1}/(\beta^k-1)$ for the competitive ratio. By Lemma 4, this upper bound is larger than the upper bound for the competitive ratio obtained in the first case. Now replacing β by $(1+k/(m-1))^{1/k}$ yields the lemma statement. □

We note that the choice of β in Theorem 5 is not arbitrary: The given β precisely minimizes the obtained upper bound of $1+2(\beta^{m+k-1})/(\beta^k-1)$ as can be shown by taking the derivative.

Interestingly, for $k=1$, our upper bound coincides with the competitive ratio of $1+2m^m/(m-1)^{m-1}$ from the optimal search strategy for a single robot, given in [4,21].

We now extend GeomSearch(α,β) to the setting where the robots are not required to start in the junction. As we will prove, even if the robots do not know the number of rays when they start, they can still achieve a competitive ratio of at most

$$1+2\cdot\frac{m-1}{k}\cdot\left(1+\frac{k}{m-1}\right)^{1+\frac{m-1}{k}} \quad . \tag{1}$$

Before describing the extension of GeomSearch(α,β), we present a lemma claiming that at a certain point in time during Algorithm GeomSearch(α,β), the distribution of the robots satisfies certain properties that will be of great use later on.

Lemma 6. *Let x be some positive real number. Consider GeomSearch(α,β) for*

$$\beta = \left(1+\frac{k}{m-1}\right)^{1/k} \qquad \textit{and} \qquad \alpha = \frac{x}{\left(1+\frac{k}{m-1}\right)^2} \quad .$$

Let t_0 denote the time at which R_0 is at the tip (i.e., exactly in the middle) of its third exploration step. Then, at time t_0, each robot has a distance of at most x from the junction and no robot R_i with $i \geq 1$ is on the same ray as R_0, except possibly in the junction.

[5] For the following calculation of the upper bound, we assume for simplicity that if $h=0$, then $R_{h-1 \pmod m}$ performs a 0th exploration step of length $\alpha\beta^{-1}$ before its 1st exploration step. Since this can only increase the upper bound, the given bound also holds if $h=0$.

Proof. By the definition of the exploration steps,

$$t_0 = 2\alpha\beta^0 + 2\alpha\beta^k + \alpha\beta^{2k} > \frac{2x}{\left(1+\frac{k}{m-1}\right)} + x \geq 2x \ .$$

Moreover, at time t_0, robot R_0 is exactly in distance x from the junction. By Observation 4, this implies that the distance of R_i from the junction is at most x, for any $1 \leq i \leq k-1$. The fact that at time t_0, R_0 is the only robot on the ray it currently occupies, follows directly from the definition of the exploration steps in conjunction with Observation 4. □

We call the distribution of the robots at time t_0 in Lemma 6 the *third distribution*. The general idea of our extended algorithm is that the robots simulate Algorithm GeomSearch(α, β) where they consider s as the junction and the path between s and j as $m-1$ separate paths (that just happen to be glued together). In order to be able to compute the appropriate β in Algorithm GeomSearch(α, β), they first have to determine the number of rays, which they do by exploring the ray they are on in both directions until they find the junction. At the point in time when the junction is found, the robots have already "wasted" some time; therefore they do not return to the junction and only then start the simulation of GeomSearch(α, β), but instead jump into a hypothetical execution of GeomSearch(α, β), i.e., they move to a configuration of points that will be reached by GeomSearch(α, β) (for some suitably chosen α) at some point in time. From there, they simply follow GeomSearch(α, β). For a formally correct description of the extended version of GeomSearch(α, β) we need some notation:

Let a_0 be the ray on which s is located and $a_1, \dots, a_{m-1}$ the remaining $m-1$ rays. We denote the path obtained by deleting $\overline{sj}$ from a_0 by a'_0 and the paths obtained by appending $a_1, \dots, a_{m-1}$ to $\overline{js}$ by $a'_1, \dots, a'_{m-1}$, respectively. We may now consider s as the junction of the m rays $a'_0, \dots, a'_{m-1}$. Therefore, provided we know m, any m-ray algorithm where the robots start in the junction can be simulated on our given input where s plays the role of the junction. In particular, the achieved competitive ratio of the simulation on our input is the same as the competitive ratio of the simulated m-ray algorithm. In the following, we describe the extension of GeomSearch(α, β) more formally:

Robots R_0 and R_1 start by exploring the ray they are located on in opposite directions until one of the two finds the exit or the junction (while everyone else simply stays in s). If the exit is found before or at the same time as the junction, then all robots immediately travel to the exit, which yields a competitive ratio of 3 (which is clearly smaller than the term given in (1), for any $2 \leq k < m$). Thus, in the following, assume that the junction is found before the exit. W.l.o.g. assume that R_1 finds the junction (which happens at time $\overline{sj}$). From here, the robots move as quickly as possible to a configuration of points that corresponds to the third distribution[6] in the (hypothetical) execution of GeomSearch(α, β)

[6] Here, a detail has to be mentioned: By changing the mapping of the m labels $a'_0, \dots, a'_{m-1}$ to the m actual rays, we can change which robot is on which ray.

on the m rays $a'_0, \dots, a'_{m-1}$ where

$$\beta = \left(1 + \frac{k}{m-1}\right)^{1/k} \qquad \text{and} \qquad \alpha = \frac{\overline{sj}}{\left(1 + \frac{k}{m-1}\right)^2} \quad .$$

By Lemma 6 moving to this configuration from the situation where the junction has just been found takes at most time $\overline{sj}$, i.e. the robots reach this configuration in a total time of at most $2\overline{sj}$. By the proof of Lemma 6 the (hypothetical) execution of GeomSearch(α, β) needs at least time $2\overline{sj}$ to reach the third distribution, i.e., this configuration. Thus, the robots can just wait in the reached configuration until time $2\overline{sj}$ (if they should have reached their respective points early) and then simulate the execution of GeomSearch(α, β) mentioned above, thereby reaching any point at least as fast as the (original) execution of GeomSearch(α, β) and hence achieving a smaller or equal competitive ratio as the one in Theorem 5.

Here two remarks are in order: First, since we assume that the junction is closer to s than the exit is to s, the exit can only be found after the robots moved to the third distribution. Hence, it is indeed enough to consider only the exits found (and therefore the competitive ratios achieved) during the simulation of GeomSearch(α, β). Second, so far, for the sake of the exposition, we ignored the detail that Theorem 5 actually requires $\alpha\beta^{m-1} < 1$. This can easily be remedied by dividing the current α repeatedly by β^k until $\alpha\beta^{m-1} < 1$ holds. Note that Lemma 6 then still holds with an analogous argumentation. Essentially, the only resulting change in the above considerations is that it takes GeomSearch(α, β) even longer to get to the configuration of points with which the above simulation starts.

By our above considerations, we obtain the following theorem:

Theorem 7. *There is an extension of Algorithm GeomSearch(α, β) for the case where the robots are not required to start in the junction that achieves a competitive ratio of at most*

$$1 + 2 \cdot \frac{m-1}{k} \cdot \left(1 + \frac{k}{m-1}\right)^{1 + \frac{m-1}{k}} \quad .$$

4.2 A Lower Bound on the Competitive Ratio

In this section, we use a simple but fundamental technique to bound the competitive ratio for the general case of k robots on $m > k$ rays from below.

Theorem 8. *There is no wireless evacuation algorithm for k robots on $m > k$ rays that achieves a competitive ratio of less than $3 + 2\lfloor (m-1)/k \rfloor$.*

We assume that the labels are changed in a way that ensures that R_0 is actually on ray a_0 in the third distribution.

Proof. Set $x = \lfloor (m-1)/k \rfloor$. Consider any wireless evacuation algorithm A for k robots on $m > k$ rays. We assume that all robots start in the junction (which we may choose to be the case as we are going to prove a lower bound). Since A solves the problem of wireless evacuation, there must be a point in time where all rays have been explored up to some depth that is strictly larger than 1, provided that the exit has not been found so far. Consider the last point in time where at least one ray has not been explored up to some depth > 1, and denote the point in time one time unit earlier by t_0. It follows that at time t_0 there must be some robot R_i at the junction or on a ray which at time $t_0 + 1$ has not been explored up to some depth > 1.

Let $\varepsilon > 0$. Let P be the set of points in distance $t_0/(2x)+\varepsilon$ from the junction and observe that $t_0/(2x) \geq 1$ since $t_0 \geq 2\lfloor (m-1)/k \rfloor$. We claim that at time $t_0 + t_0/(2x)$, robot R_i has explored at most $x - 1$ points in P: Since R_i starts in the junction and, at time t_0, is again in the junction or on a ray where the corresponding point from P will not be explored up to and including time t_0+1, it must travel a total distance of at least $2y(t_0/(2x) + \varepsilon)$ in order to explore y points from P up to time t_0. Thus, we obtain $2y(t_0/(2x)+\varepsilon) \leq t_0$ which implies $y < x$ and thereby proves the claim. Note that robot R_i cannot explore a point from P between t_0 and $t_0 + t_0/(2x)$ because of its location at time t_0.

Moreover, we claim that at time $t_0 + t_0/(2x)$, any robot R_h with $h \neq i$ has explored at most x points in P: Similarly to above, in order to explore y points from P starting in the junction, robot R_h has to travel a distance of at least $(2y-1)(t_0/(2x)+\varepsilon)$. We obtain $(2y-1)(t_0/(2x)+\varepsilon) \leq t_0+t_0/(2x)$ which implies $y < x + 1$ and thereby proves the claim. Hence, at time $t_0 + t_0/(2x)$, at most $kx - 1 \leq m - 2$ points from P have been explored in total. Thus, there exist two points $p_1, p_2 \in P$ that have not been explored at time $t_0 + t_0/(2x)$. Let t_1 and t_2 be the points in time when p_1 and p_2 are explored (for the first time), respectively. W.l.o.g. assume that $t_1 \leq t_2$.

Now consider the input instance where the exit is at point p_2. Since some robot is at p_1 at time $t_1 \geq t_0 + t_0/(2x)$, this robot cannot be at p_2 before time $t_1 + 2(t_0/(2x) + \varepsilon)$, by the definition of P. We obtain a lower bound of

$$\frac{t_0 + \frac{t_0}{2x} + 2(\frac{t_0}{2x} + \varepsilon)}{\frac{t_0}{2x} + \varepsilon} = 2 + \frac{2x+1}{1 + \frac{2\varepsilon x}{t_0}}$$

for the competitive ratio. By making ε arbitrarily small our lower bound gets arbitrarily close to $3+2x = 3+2\lfloor (m-1)/k \rfloor$ which proves the theorem statement. □

5 Concluding Remarks

We studied the problem of collaboratively evacuating k robots on m concurrent rays, using wireless communication. To the best of our knowledge, our work is the first that considers not starting on the junction j of the m rays, and also to consider $k < m$ robots for the specific problem of wireless collaborative evacuation on m rays.

For the case of $k = m$ robots, a simple waiting strategy gives a competitive ratio of 4, with a constructive lower bound of $2 + \sqrt{7/3} \approx 3.52753$ for every $k = m \geq 3$. For the specific case of $k = m = 3$, we develop a parametrized class of algorithms with a nearly matching competitive ratio of $2 + \sqrt{3} \approx 3.73205$, where the parameter choice decides on the first search depth beyond the junction j, once the junction is found.

Unlike prior work, not starting on the junction j allows to consider the scenario of the robots being initially oblivious to the number of rays. Our optimization over the parameter space of a geometric search strategy yields an algorithm with a competitive ratio of $1 + 2 \cdot \frac{m-1}{k} \cdot \left(1 + \frac{k}{m-1}\right)^{1+\frac{m-1}{k}}$. For a lower bound, we give a simple but fundamental argument, resulting in the fact that no algorithm can obtain a better competitive ratio than $3 + 2\lfloor (m-1)/k \rfloor$ for every combination of k, m with $k < m$ – even when starting on j.

Acknowledgments We would like to thank the anonymous reviewers for their helpful comments. Klaus-Tycho Foerster is supported by the Danish VILLUM FONDEN project ReNet.

References

1. Spyros Angelopoulos. Deterministic searching on the line. In *Encyclopedia of Algorithms*, pages 531–533. Springer New York, 2016. `doi:10.1007/978-1-4939-2864-4_106`.
2. Spyros Angelopoulos, Diogo Arsénio, Christoph Dürr, and Alejandro López-Ortiz. Multi-processor search and scheduling problems with setup cost. *Theory of Computing Systems*, pages 1–34, 2016. `doi:10.1007/s00224-016-9691-3`.
3. Spyros Angelopoulos, Alejandro López-Ortiz, and Konstantinos Panagiotou. Multi-target ray searching problems. *Theor. Comput. Sci.*, 540:2–12, 2014. `doi:10.1016/j.tcs.2014.03.028`.
4. Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Rawlins. Searching in the plane. *Inf. Comput.*, 106(2):234–252, 1993. `doi:10.1006/inco.1993.1054`.
5. Ricardo A. Baeza-Yates and René Schott. Parallel searching in the plane. *Comput. Geom.*, 5:143–154, 1995. `doi:10.1016/0925-7721(95)00003-R`.
6. Anatole Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964. `doi:10.1007/BF02759737`.
7. Anatole Beck and D. J. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429, 1970. `doi:10.1007/BF02798690`.
8. Richard Bellman. An optimal search. *SIAM Review*, 5(3):274–274, 1963. `doi:10.1137/1005070`.
9. Prosenjit Bose and Jean-Lou De Carufel. A general framework for searching on a line. In Mohammad Kaykobad and Rossella Petreschi, editors, *WALCOM: Algorithms and Computation - 10th International Workshop, WALCOM 2016, Kathmandu, Nepal, March 29-31, 2016, Proceedings*, volume 9627 of *Lecture Notes in Computer Science*, pages 143–153. Springer, 2016. `doi:10.1007/978-3-319-30139-6_12`.

10. Prosenjit Bose, Jean-Lou De Carufel, and Stephane Durocher. Searching on a line: A complete characterization of the optimal solution. *Theor. Comput. Sci.*, 569:24–42, 2015. doi:10.1016/j.tcs.2014.12.007.
11. Sebastian Brandt, Felix Laufenberg, Yuezhou Lv, David Stolz, and Roger Wattenhofer. Collaboration without communication: Evacuating two robots from a disk. In Dimitris Fotakis, Aris Pagourtzis, and Vangelis Th. Paschos, editors, *Algorithms and Complexity - 10th International Conference, CIAC 2017, Athens, Greece, May 24-26, 2017, Proceedings*, volume 10236 of *Lecture Notes in Computer Science*, pages 104–115, 2017. doi:10.1007/978-3-319-57586-5_10.
12. Marek Chrobak, Leszek Gasieniec, Thomas Gorry, and Russell Martin. Group search on the line. In Giuseppe F. Italiano, Tiziana Margaria-Steffen, Jaroslav Pokorný, Jean-Jacques Quisquater, and Roger Wattenhofer, editors, *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*, volume 8939 of *Lecture Notes in Computer Science*, pages 164–176. Springer, 2015. doi:10.1007/978-3-662-46078-8_14.
13. 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. ACM, 2016. doi:10.1145/2833312.2833318.
14. Jurek Czyzowicz, Leszek Gasieniec, Thomas Gorry, Evangelos Kranakis, Russell Martin, and Dominik Pajak. Evacuating robots via unknown exit in a disk. In Fabian Kuhn, editor, *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, volume 8784 of *Lecture Notes in Computer Science*, pages 122–136. Springer, 2014. doi:10.1007/978-3-662-45174-8_9.
15. 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 Vangelis Th. Paschos and Peter Widmayer, editors, *Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings*, volume 9079 of *Lecture Notes in Computer Science*, pages 140–152. Springer, 2015. doi:10.1007/978-3-319-18173-8_10.
16. Jurek Czyzowicz, Maxime Godon, Evangelos Kranakis, Maxime Godon, Danny Krizanc, Wojciech Rytter, and Michal Wlodarczyk. Evacuation from a disc in the presence of a faulty robot. In *Proc. SIROCCO*, 2017.
17. Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, and Jaroslav Opatrny. Search on a line with faulty robots. In George Giakkoupis, editor, *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 405–414. ACM, 2016. doi:10.1145/2933057.2933102.
18. Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, Lata Narayanan, Jaroslav Opatrny, and Sunil M. Shende. Wireless autonomous robot evacuation from equilateral triangles and squares. In Symeon Papavassiliou and Stefan Ruehrup, editors, *Ad-hoc, Mobile, and Wireless Networks - 14th International Conference, ADHOC-NOW 2015, Athens, Greece, June 29 - July 1, 2015, Proceedings*, volume 9143 of *Lecture Notes in Computer Science*, pages 181–194. Springer, 2015. doi:10.1007/978-3-319-19662-6_13.

19. Erik D. Demaine, Sándor P. Fekete, and Shmuel Gal. Online searching with turn cost. *Theor. Comput. Sci.*, 361(2-3):342–355, 2006. doi:10.1016/j.tcs.2006.05.018.
20. Klaus-Tycho Foerster and Roger Wattenhofer. Lower and upper competitive bounds for online directed graph exploration. *Theor. Comput. Sci.*, 655:15–29, 2016. doi:10.1016/j.tcs.2015.11.017.
21. Shmuel Gal. Minimax solutions for linear search problems. *SIAM Journal on Applied Mathematics*, 27(1):17–30, 1974. doi:10.1137/0127002.
22. Mikael Hammar, Bengt J. Nilsson, and Sven Schuierer. Parallel searching on m rays. In Christoph Meinel and Sophie Tison, editors, *STACS 99, 16th Annual Symposium on Theoretical Aspects of Computer Science, Trier, Germany, March 4-6, 1999, Proceedings*, volume 1563 of *Lecture Notes in Computer Science*, pages 132–142. Springer, 1999. doi:10.1007/3-540-49116-3_12.
23. Mikael Hammar, Bengt J. Nilsson, and Sven Schuierer. Parallel searching on m rays. *Comput. Geom.*, 18(3):125–139, 2001. doi:10.1016/S0925-7721(00)00028-6.
24. Patrick Jaillet and Matthew Stafford. Online searching. *Operations Research*, 49(4):501–515, 2001. doi:10.1287/opre.49.4.501.11227.
25. Thomas Kamphans and Elmar Langetepe. Optimal competitive online ray search with an error-prone robot. In Sotiris E. Nikoletseas, editor, *Experimental and Efficient Algorithms, 4th InternationalWorkshop, WEA 2005, Santorini Island, Greece, May 10-13, 2005, Proceedings*, volume 3503 of *Lecture Notes in Computer Science*, pages 593–596. Springer, 2005. doi:10.1007/11427186_51.
26. Ming-Yang Kao, Yuan Ma, Michael Sipser, and Yiqun Lisa Yin. Optimal constructions of hybrid algorithms. *J. Algorithms*, 29(1):142–164, 1998. doi:10.1006/jagm.1998.0959.
27. Ming-Yang Kao, John H. Reif, and Stephen R. Tate. Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem. *Inf. Comput.*, 131(1):63–79, 1996. doi:10.1006/inco.1996.0092.
28. Dennis Komm, Rastislav Královic, Richard Královic, and Jasmin Smula. Treasure hunt with advice. In Christian Scheideler, editor, *Structural Information and Communication Complexity - 22nd International Colloquium, SIROCCO 2015, Montserrat, Spain, July 14-16, 2015, Post-Proceedings*, volume 9439 of *Lecture Notes in Computer Science*, pages 328–341. Springer, 2015. doi:10.1007/978-3-319-25258-2_23.
29. Alejandro López-Ortiz and Sven Schuierer. The ultimate strategy to search on m rays? *Theor. Comput. Sci.*, 261(2):267–295, 2001. doi:10.1016/S0304-3975(00)00144-4.
30. Alejandro López-Ortiz and Sven Schuierer. On-line parallel heuristics, processor scheduling and robot searching under the competitive framework. *Theor. Comput. Sci.*, 310(1-3):527–537, 2004. doi:10.1016/j.tcs.2003.08.001.
31. Sven Schuierer. A lower bound for randomized searching on m rays. In Rolf Klein, Hans-Werner Six, and Lutz M. Wegner, editors, *Computer Science in Perspective, Essays Dedicated to Thomas Ottmann*, volume 2598 of *Lecture Notes in Computer Science*, pages 264–277. Springer, 2003. doi:10.1007/3-540-36477-3_20.
32. Daniel Dominic Sleator and Robert Endre Tarjan. Amortized efficiency of list update and paging rules. *Commun. ACM*, 28(2):202–208, 1985. doi:10.1145/2786.2793.
33. Stephen R. Tate. Randomized searching on rays or the line. In *Encyclopedia of Algorithms*, pages 1757–1759. Springer New York, 2016. doi:10.1007/978-1-4939-2864-4_328.