Search on a Line with Faulty Robots

Jurek Czyzowicz*
Dept. d'informatique
Universite d'Outaouais
Gatineau, Canada
Jurek.Czyzowicz@uqo.ca

Evangelos Kranakis†
School of Computer Science
Carleton University, Ottawa
Ottawa, Canada
kranakis@scs.carleton.ca

Danny Krizanc
Dept. Math. and Comp. Sci.
Wesleyan University
Middletown CT, USA
dkrizanc@wesleyan.edu

Lata Narayanan‡
Dept. Comp. Sc. and
Software Eng.
Concordia University
Montreal, Canada
lata@cs.concordia.ca

Jaroslav Opatrny
Dept. Comp. Sc. and
Software Eng.
Concordia University
Montreal, Canada
opatrny@cs.concordia.ca

ABSTRACT

We consider the problem of searching on a line using n mobile robots, of which at most f are faulty, and the remaining are reliable. The robots start at the same location and move in parallel along the line with the same speed. There is a *target* placed on the line at a location unknown to the robots. Reliable robots can find the target when they reach its location, but faulty robots cannot detect the target. Our goal is to design a parallel algorithm minimizing the competitive ratio, represented by the worst case ratio between the time of arrival of the first reliable robot at the target, and the distance from the source to the target.

If $n \geq 2f + 2$, there is a simple algorithm with competitive ratio 1. For $f < n < 2f + 2$ we develop a new class of algorithms, called *proportional schedule algorithms.* For any given (n, f), we give a proportional schedule algorithm $A(n, f)$, whose competitive ratio is

$$\left(\frac{4f+4}{n}\right)^{\frac{2f+2}{n}} \left(\frac{4f+4}{n} - 2\right)^{1-\frac{2f+2}{n}} + 1.$$

Setting $a = n/f$ as a constant, the asymptotic competitive ratio is $(4/a)^{2/a}\,(4/a-2)^{1-2/a} + 1$.

Our search algorithm is easily seen to be optimal for the case $n = f + 1$. We also show that as n tends to ∞ the competitive ratio of our algorithm for the case $n = 2f + 1$ approaches 3 and this is optimal. More precisely, we show that asymptotically, the competitive ratio of our proportional schedule algorithm $A(2f + 1, f)$ is at most $3 + \frac{4 \ln n}{n}$, while any search algorithm has a lower bound $3 + \frac{2 \ln n}{n}$ on its competitive ratio.

CCS Concepts

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

Keywords

Search on a line; Faulty robots; Cow-path problem; Competitive ratio.

1. INTRODUCTION

Exploring an environment to find a target placed at an unknown location is a well studied problem in computer science, robotics, operations research, and mathematics. Starting with the work of Bellman [9] and Beck [7], many researchers have studied the optimal trajectory to be followed by a single robot searching for a target placed at an unknown location on a line. The goal of the robot is to minimize the competitive ratio, that is, the supremum, over all possible target locations, of the ratio between the distance travelled by the robot until it finds the target, and the distance of the target from the robot's starting position. It is well-known that, for the case of a single robot on a line, the optimal trajectory uses a doubling strategy: roughly, the robot travels some distance x to the left, then turns around and moves $2x$ to the right, and so on, until it finds the target. This trajectory has a competitive ratio of 9 (e.g. see Baeza-Yates et al. [6]).

In this paper, we study the problem of *parallel search* on a line, by n robots, of which some robots are *faulty*, and the remaining robots are *reliable*. All robots are assumed to start at a shared initial location, and to move at maximum speed 1 along the line. A robot can change the direction of its movement at some points defined by its algorithm. An algorithm for parallel search specifies a trajectory, e.g. the turning points, for each robot; the identities of robots are different, and they may have different trajectories. We assume that at most f of the robots are faulty. A faulty

*Supported in part by NSERC of Canada.
†Supported in part by NSERC of Canada.
‡Supported in part by NSERC of Canada.

PODC'16, July 25–28, 2016, Chicago, IL, USA.
© 2016 ACM. ISBN 978-1-4503-3964-3/16/07. . . $15.00
DOI: http://dx.doi.org/10.1145/2933057.2933102

robot follows its assigned trajectory and is indistinguishable from a reliable robot, except that *a faulty robot does not detect the target while visiting its location*. Thus, the target is detected only when its location is visited by a reliable robot. Therefore, an algorithm for a parallel search with f faulty robots must make sure that any point on the line can be visited by at least $f+1$ robots. Accordingly, we define the search time of a parallel search algorithm with faulty robots to be the time when the first reliable robot reaches the target, which could be, in the worst case, the visit of the $f+1$st robot. The competitive ratio of a search algorithm is defined to be the worst case ratio between the search time to the distance of the target from the starting position.

Since a faulty robot behaves the same as a reliable robot except that it does not detect the target, it follows that a robot can be deduced to be faulty only when the target is found by a reliable robot at a position previously visited by the faulty robot. It is therefore irrelevant if the robots were faulty at the beginning or later during the search. We note that the assumption of f faulty robots also models the situation when the target is hard to detect, and therefore we require visits by $f+1$ robots to the location of the target in order to assure the detection of the target. In our algorithms, we do not assume any communication among robots, but our lower bounds apply even in the presence of communication between the robots.

Observe that if $n \geq 2f+2$, we can simply partition the robots into two groups of $f+1$ robots each, and send the two groups in opposite directions from the starting position. Since each group has at least one reliable robot, a reliable robot is sure to discover the target in time d where d is the distance of the target from the starting position of the robots. This algorithm has competitive ratio one and is clearly optimal.

Thus, in the sequel we study the case when $f < n < 2f+2$. The trajectory of each robot can be defined by a possibly infinite sequence $x_0, x_1, x_2, \ldots$ of reals, called *turning points*, which are the points where the trajectory changes direction. Starting at the origin s the robot travels in one direction to reach x_0. At x_0 it turns around until x_1, where the direction is changed again, etc. In the sequel, a trajectory with an infinite number of turning points will be called a *zig-zag* strategy. It is natural to consider zig-zag strategies in which $x_{i+1} = \kappa x_i$ for every $i \geq c$ for some constant c; we call κ the *expansion factor* of the strategy. Indeed, the optimal doubling strategy for a single robot given in [7, 9] has expansion factor 2. In the context of parallel search, all robots could have different expansion factors, or have the same expansion factor, but start at different times or move at different speeds.

1.1 Our results

As mentioned earlier, when $n \geq 2f+2$, there is a trivial optimal solution. For $f < n < 2f+2$, we develop *proportional schedule algorithms*, a new class of algorithms for parallel search on a line with faulty robots. In Section 3, we prove the existence of this class of algorithms and show that for each pair (n, f), there exists a proportional schedule algorithm $A(n, f)$ with competitive ratio at most

$$\left(\frac{4f+4}{n}\right)^{\frac{2f+2}{n}} \left(\frac{4f+4}{n}-2\right)^{1-\frac{2f+2}{n}} + 1.$$

Setting $a = n/f$ as constant (i.e. when a fixed proportion of robots may be faulty), this gives us the asymptotic expression

$$(4/a)^{2/a} (4/a-2)^{1-2/a} + 1$$

for the competitive ratio.

When $n = f+1$, the proportional schedule algorithm $A(f+1, f)$ has a competitive ratio of 9. It is easy to see that this is optimal: if there is an algorithm $\mathcal{A}$ with a competitive ratio better than 9, then following the trajectory of the first robot in $\mathcal{A}$ would give a better competitive ratio than 9 for the case of a single robot, since it could be that in the execution of $\mathcal{A}$, the first robot is the only reliable robot. We remark that a competitive ratio of 9 is also achieved by all robots starting at the same time, and moving together while following a doubling strategy.

Finally, for $n = 2f+1$, the proportional schedule algorithm $A(2f+1, f)$ has competitive ratio at most $3 + \frac{4 \ln n}{n}$ (after excluding low order terms). In terms of lower bounds, in Section 4, we prove that any parallel search algorithm with $n < 2f+2$ robots has competitive ratio at least α where $\alpha > 3$ and satisfies $(\alpha-1)^n(\alpha-3) = 2^{n+1}$. Asymptotically, this yields a lower bound of $3 + \frac{2 \ln n}{n}$ (after excluding low order terms) on the competitive ratio. Thus, our proportional schedule algorithm $A(2f+1, f)$ is asymptotically optimal.

Table 1 gives our results for some specific values of n and f. Observe that the expansion factor for $A(n, f)$ changes when changing values of n or f. For $n = 2f+1$, it turns out that the expansion factor for $A(2f+1, f)$ is always $n+1$, and for $n = f+1$ the expansion factor is 2.

n	f	comp. ratio of $A(n,f)$	lower bound on comp. ratio	expansion factor of $A(n,f)$
2	1	9	9	2
3	1	5.24	3.76	4
3	2	9	9	2
4	1	1	1	
4	2	6.2	3.649	3
4	3	9	9	2
5	1	1	1	
5	2	4.43	3.57	6
5	3	6.76	3.57	2.67
5	4	9	9	2
11	5	3.73	3.345	12
41	20	3.24	3.12	42

Table 1: Upper and lower bounds for specific values of n and f

1.2 Related work

In general, search is concerned with minimizing the time to find a hidden target under various conditions on the environment and capabilities of the searcher(s). When the mobile robots do not know the environment in advance then search involves exploring [2, 3, 20, 25], and usually exploration is combined with mapping of the environment and positioning of the robots within it [29, 34]. Coordinating exploration by a team of robots is one of the core research themes in robotics [12, 23, 37, 38] and distributed computing [22]. Related search questions include the lost at sea problem [24, 27], the cow-path problem [7, 9], and the plane searching problem [6], to name a few.

The study of search on an infinite line, sometimes called linear search, started with the independent works of Bellman [9] and Beck [7]. It consists of finding a motionless target on an infinite line in the smallest possible worst-case time expressed as a function of the distance of the target to the starting position. The competitive ratio of 9 was first shown to be optimal in [8] and numerous variations of this question have been studied. For example, when an upper bound on the distance to the target is known in advance, slightly better bounds were shown [10], while [19] studied the question when a cost is charged for changing the search direction. Recently, [14] showed that having many searchers, communicating upon meeting, does not improve the competitive ratio, when the search time is determined by the time of arrival of the last searcher at the target. Several papers considered the problem of searching in the two-dimensional plane by one or more searchers [5, 6]. The book [4] reviews most search games studied earlier.

Fault tolerance in distributed computing problems has been extensively studied in the past (see, e.g., [26, 31, 32]). Failures were most often related to static elements of the environment (network nodes and links) rather than mobile components. However such malfunctions are sometimes modelled by dynamic alteration of the network [13, 30]. The problems of having some faulty mobile robots were investigated in the context of the problems of gathering [1, 18, 21, 36], convergence [11, 15], flocking [39], and patrolling [17]. Some papers investigated the case of unreliable or inaccurate robot sensing devices, e.g., [16, 28, 36]. Perhaps the most related to our study is [17], where a collection of robots, some of which are unreliable, perform efficient patrolling of a fence.

2. PRELIMINARIES AND NOTATION

We assume that the n robots are named $a_0, a_1, \ldots a_{n-1}$. We first define a space/time 2D-representation of the movement of a robot on the line that will prove useful in arguing about the performance of the search algorithms proposed in this paper.

Consider the movement of a robot on an infinite line L. The trajectory of such a robot can be represented in the half-plane by a curve consisting of points (x, t) where x is the position of the robot on L at time t. As robots always use maximal available (unit) speed, each trajectory is composed of segments using slopes -1 and 1, see Figure 1.

Moreover, in our algorithms, we will require that the turning points of all the zig-zag strategies, are located on two lines forming a cone, as in Figure 2. This implies that all robots have the same expansion factor, though they may have different turning points.

For a fixed real number $\beta > 1$ we denote by C_β the cone delimited by a pair of lines $t = x\beta$ for $x \geq 0$, and $t = -x\beta$ for $x \leq 0$.

Definition 1. *Consider a point* $(x_0, |x_0\beta|)$ *on the boundary of the cone* C_β*, By a* zig-zag *movement of a robot defined by cone* C_β *and point* $(x_0, |x_0\beta|)$ *we mean the motion of the robot with unit speed inside the cone* C_β *that starts at* $(x_0, |x_0\beta|)$ *and that reverses its direction whenever it arrives at the boundary of* C_β*.*

The movement of a robot defined by a cone and a point on it boundary is depicted in Figure 2. Notice that this gives rise to a sequence $x_0, x_1, x_2, \ldots$ of *turning points* on the line at which the robot changes the direction of its movement.

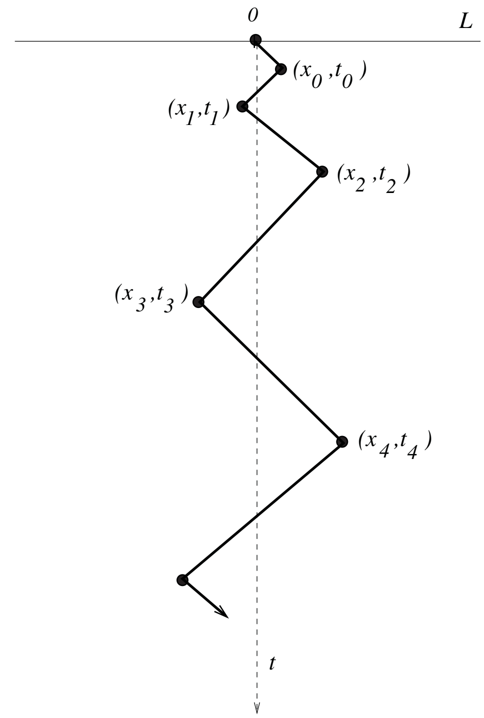


Figure 1: A general zig-zag strategy with turning points (x_i, t_i), $1 \leq i \leq 4$.

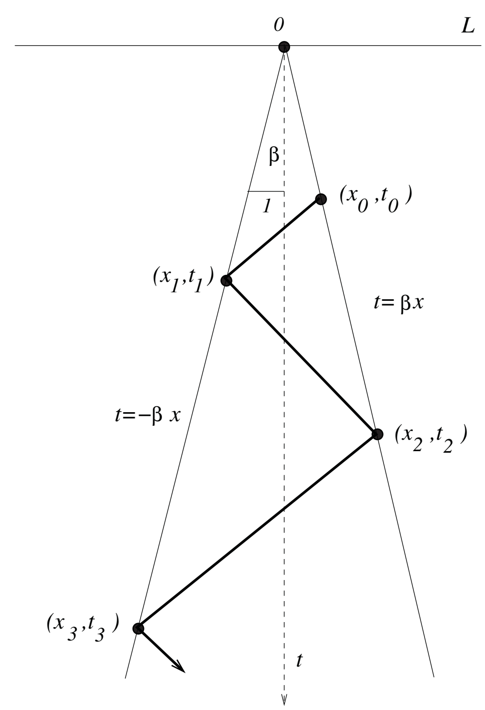


Figure 2: A zig-zag strategy defined by cone C_β, $\beta > 1$ (delimited by lines $t = \beta x$ for $x \geq 0$ and $t = -\beta x$ for $x \leq 0$) and point (x_0, t_0) on the cone.

Using the basic properties of triangles, it is easy to verify the following lemma which gives the expansion factor determined by C_β.

LEMMA 1. *Let x_0 be the initial position of a robot on the line at time t_0 which corresponds to point (x_0, t_0) on the boundary of the cone C_β, where $\beta > 1$. Then the turning points of the robot are given by the formula*

$$x_i = x_0 \left(\frac{\beta+1}{\beta-1}\right)^i (-1)^i \tag{1}$$

with corresponding expansion factor $(\beta+1)/(\beta-1)$.

3. UPPER BOUND

In this section we give a set of ziz-zag movements for a collection of n robots searching the line. We define set movements for all robots according to the same cone C_β. We also assume that two different robots will never share the same turning point. The collections of zigzag movements that we define for a set of robots will have a specific regularity that we define below. We call such collections of movements *proportional schedules*. The sets of turning points (taken by all robots) of proportional schedules will have the property that the distances between two consecutive turning points (corresponding to different robots) will behave according to some geometric sequence.

Let C_β be a cone for some fixed value of $\beta > 1$. Consider two robots a_0, a_1 and their zig-zag movements defined by C_β. Let $0 < \tau' < \tau$" be two consecutive turning points of a_0 on the same side of the cone (by assumption above neither of them is a turning point of a_1). It is easy to see that there is exactly one point τ_1, which is a turning point of a_1 such that $\tau' < \tau_1 < \tau$". By a similar argument, this is true for any pair of robots, hence, without loss of generality we can number the collection of n robots in such a way that for a sequence of turning points $0 < \tau_0 = \tau' < \tau_1 < \cdots < \tau_{n-1} < \tau_n = \tau''$, such that τ' and τ" are two consecutive turning points of a_0, we have that τ_i is a turning point of robot a_i for $i = 1, \ldots, n-1$. Observe that choosing a sequence of turning points $\tau_0, \tau_1, \cdots, \tau_n$, according to the condition above, entirely defines the schedule of complete ziz-zag movements of all robots. We call the schedule *proportional* if the ratio of consecutive differences of turning points on the same side of the cone remains the same. More precisely, we give the following definition.

DEFINITION 2. *Suppose that $0 < \tau_0 < \tau_1 < \cdots$ is the infinite sequence of positive turning points of all the robots in their respective zig-zag movements. We say that the schedule is* proportional *if for some real value r, the ratio $\frac{\tau_{i+1}-\tau_i}{\tau_i-\tau_{i-1}} = r$, for $i = 1, 2, \ldots$. We call r the proportionality ratio of the schedule.*

We denote by $S_\beta(n)$ the proportional schedule using n robots zig-zagging within the cone C_β. Given n and a value of $\beta > 1$, a turning point of one of the robots completely defines the proportional schedule $S_\beta(n)$. In the following technical lemma we determine the ratio of the proportional schedule $S_\beta(n)$ and establish some of its properties.

LEMMA 2. *Consider any $\beta > 1$ and n robots moving on the line according to the proportional schedule $S_\beta(n)$. The proportionality ratio of schedule $S_\beta(n)$ equals*

$$r = \left(\frac{\beta+1}{\beta-1}\right)^{2/n}. \tag{2}$$

Moreover, if $\tau_i > 0$ is a turning point of some robot at time t_i and $\tau_{i+1} > \tau_i$ is the next larger turning point by any robot visited at time t_{i+1} then we have $t_{i+1} = t_i + \tau_i\beta(r-1)$ and $\tau_{i+1} = r\tau_i$. [1]

PROOF. Let $A_i = (\tau_i, t_i)$, for $i = 0, 1, \ldots n-1$, be such that $\tau_0, \tau_1, \ldots, \tau_{n-1}$ be the consecutive turning points of robots $a_0, a_1, \ldots, a_{n-1}$ resulting from a proportional schedule, and $t_0, t_1, \ldots, t_{n-1}$ be the times of their respective visits. Let $d = |A_0A_1|$ denote the length of the segment A_0A_1 (see Figure 3). By the proportionality condition $\frac{\tau_2-\tau_1}{\tau_1-\tau_0} = r$

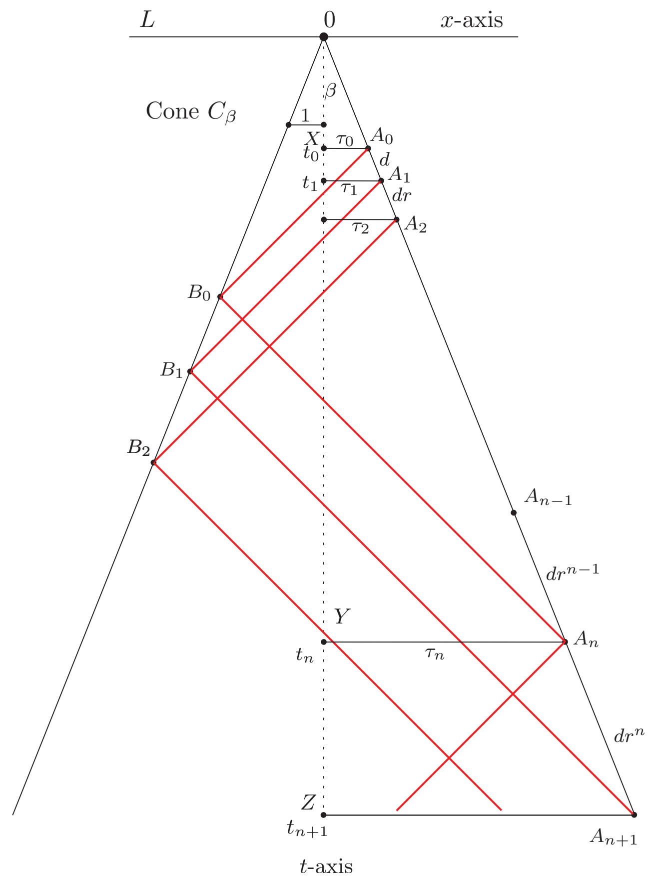


Figure 3: Proportional schedule for n robots $a_0, a_1, \ldots, a_{n-1}$, in the cone C_β .

and using similar triangles we conclude that $|A_1A_2| = dr$, and consequently by induction $|A_iA_{i+1}| = dr^i$, for $i = 0, 1, \ldots, n-1$. Moreover, $|A_{n-1}A_n| = dr^{n-1}$ and $|A_nA_{n+1}| = dr^n$, where $A_n := (\tau_n, t_n)$, $A_{n+1} := (\tau_{n+1}, t_{n+1})$, and τ_n, τ_{n+1} are the next positive turning points after A_0, A_1 of robots a_0, a_1, respectively.

Consequently, the length of segment A_0A_n is given by the formula

[1]By symmetry the same result applies for negative turning points.

$$|A_0A_n| = d + dr + \cdots + dr^{n-1} = d\frac{r^n - 1}{r - 1}. \tag{3}$$

Denote by $B_0 = (\tau', t')$ the point on the cone C_β corresponding to the negative turning point τ' of robot a_0 at time t' such that $t_0 < t' < t_n$ (see Figure 3). By Lemma 1,

$$\begin{aligned} t_n - t_0 &= (t_n - t') + (t' - t_0) \\ &= \left(\tau_0 + \tau_0\frac{\beta+1}{\beta-1}\right) + \left(\tau_0\frac{\beta+1}{\beta-1} + \tau_0\left(\frac{\beta+1}{\beta-1}\right)^2\right) \\ &= \tau_0\left(1 + \frac{\beta+1}{\beta-1}\right)^2 = \tau_0\frac{4\beta^2}{(\beta-1)^2}. \end{aligned}$$

Consider points $X := (0, t_0), Y := (0, t_n)$. By similarity of triangles we have

$$\frac{|A_nA_0|}{|XY|} = \frac{|A_0O|}{|XO|}. \tag{4}$$

and since $|XO| = \tau_0\beta$ and $|A_0O| = \tau_0\sqrt{1+\beta^2}$ we get

$$|A_nA_0| = \frac{\tau_0\frac{4\beta^2}{(\beta-1)^2}\tau_0\sqrt{1+\beta^2}}{\tau_0\beta} = \frac{4\tau_0\beta\sqrt{\beta^2+1}}{(\beta-1)^2}. \tag{5}$$

Therefore

$$d\frac{r^n-1}{r-1} = \frac{4\tau_0\beta\sqrt{\beta^2+1}}{(\beta-1)^2}. \tag{6}$$

Robot a_1 visited its turning point τ_1 at time t_1 (depicted by point A_1 in Figure 3). Let t_{n+1} be its next turning point τ_{n+1} visited at t_{n+1} and let $A_{n+1} = (\tau_{n+1}, t_{n+1})$ and $Z = (0, t_{n+1})$. Note that

$$\begin{aligned} \frac{|OA_{n+1}|}{|OA_n|} &= \frac{\tau_0\sqrt{\beta^2+1} + d + dr + \cdots + dr^n}{\tau_0\sqrt{\beta^2+1} + d + dr + \cdots + dr^{n-1}} \\ &= 1 + \frac{dr^n}{\tau_0\sqrt{\beta^2+1} + d\frac{r^n-1}{r-1}} \end{aligned} \tag{7}$$

On the other hand we have

$$\begin{aligned} \frac{|OA_{n+1}|}{|OA_n|} &= \frac{|A_{n+1}B_1|}{|A_nB_0|} \\ &\quad \text{(by similarity of } \triangle OB_1A_{n+1} \text{ and } \triangle OB_0A_n\text{)}, \\ &= \frac{|A_{n+1}A_1|}{|A_nA_0|} \\ &\quad \text{(by similarity of } \triangle A_1B_1A_{n+1} \text{ and } \triangle A_0B_0A_n\text{)}, \\ &= \frac{\sum_{i=1}^{n} dr^i}{\sum_{i=0}^{n-1} dr^i} = r \end{aligned} \tag{8}$$

Combining Equations (7) and (8) we conclude that

$$r - 1 = \frac{dr^n}{\tau_0\sqrt{\beta^2+1} + d\frac{r^n-1}{r-1}} \tag{9}$$

To solve the system of Equations (6) and (9) for the unknowns r and d, we argue as follows. Substituting d from Equation (6) into Equation (9) we get

$$r - 1 = \frac{\frac{4\tau_0\beta\sqrt{\beta^2+1}}{(\beta-1)^2}\frac{r-1}{r^n-1}r^n}{\tau_0\sqrt{\beta^2+1} + \frac{4\tau_0\beta\sqrt{\beta^2+1}}{(\beta-1)^2}} \tag{10}$$

Simplifying Equation (10) we derive

$$1 = \frac{\frac{4\beta}{(\beta-1)^2}\frac{r^n}{r^n-1}}{1 + \frac{4\beta}{\beta^2-1}} \tag{11}$$

Multiplying out terms in Equation (11) and simplifying the expression we derive Equation (2). This proves the first part of Lemma 2.

We now prove the second part of Lemma 2. From Equation (6) we have

$$d = \frac{4\tau_0\beta\sqrt{\beta^2+1}}{(\beta-1)^2}\frac{r-1}{r^n-1} = \tau_0\sqrt{\beta^2+1}(r-1). \tag{12}$$

By similarity of triangles, observe that

$$\frac{t_1 - t_0}{d} = \frac{\beta}{\sqrt{\beta^2+1}}.$$

Hence, $t_1 = t_0 + \tau_0\beta(r-1)$. By similarity of triangles (see Figure 3) we have

$$\frac{\tau_1}{\tau_0} = \frac{|A_1O|}{|A_0O|} = \frac{d + \tau_0\sqrt{\beta^2+1}}{\tau_0\sqrt{\beta^2+1}}$$

After substituting in the last Equation the value of d as obtained from Equation (12) we get $\tau_1 = r\tau_0$. This concludes the proof of Lemma 2. $\square$

Consider the proportional schedule for 3 robots when one of them may be faulty. The target is known to be found if at least two robots visited its position. Note the "tower-like" shape represented with bold lines on Figure 4. For any point (x, t) inside this shape we have that at time t point x was seen by at least two robots. Also if point (x, t) is outside this shape it means that at time t point x was seen by one robot or none. A similar tower-like shape may be constructed for a collection of n robots with up to f faulty robots for any n, f such that $f < n < 2f + 2$.

Definition 3. *Denote by $T_{f+1}(x)$ the time of the visit of point x by the $(f+1)$-st robot according to the schedule $S_\beta(n)$. Define the function*

$$K_\beta^{n,f}(x) := \frac{T_{f+1}(x)}{|x|},$$

and let

$$CR_\beta^{n,f} = \sup_{x \in \mathcal{R}} K_\beta^{n,f}(x)$$

denote the competitive ratio of the proportional schedule $S_\beta(n)$ when there are f faulty robots.

In the next two lemmas we establish properties of $K_\beta^{n,f}(x)$ and $T_{f+1}(x)$ needed to derive the competitive ratio of our algorithms.

Lemma 3. *The function $K_\beta^{n,f}(x)$ has discontinuities at turning points and it is continuous and decreasing in any interval not containing a turning point.*

Proof. Consider a vertical line, denoted by V, intersecting point x of the (infinite) line L. The line V intersects first the boundary of the cone C_β and subsequently the trajectories of the robots (see Figure 4.)

Clearly, the distance along line V from L to the intersection of the trajectory of the $(f+1)$st robot equals $T_{f+1}(x)$.

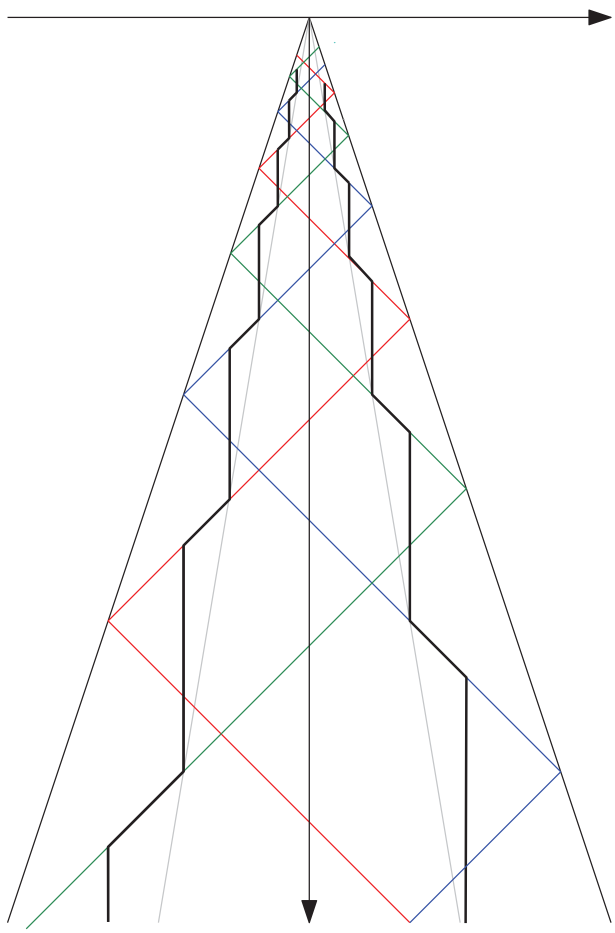


Figure 4: Searching by three robots one of which is faulty.

Observe that by moving line V continuously to the right, when it crosses a turning point of a robot, say a_0 without loss of generality, it ceases to intersect its trajectory. Consequently, the value of $T_{f+1}(x)$ increases by the vertical distance between the trajectories of the $(f+1)$st and $(f+2)$nd robots following a_0 at the moment of their visitation of point x, here denoted by a_f and a_{f+1}. This creates a discontinuity.

Consider now the values of $K_\beta^{n,f}(x)$ and $K_\beta^{n,f}(x+\epsilon)$ such that no turning point is in the interval $[x, x+\epsilon]$. As the $(f+1)$st robot at points x and $x+\epsilon$ remains the same (along V) we must have $T_{f+1}(x+\epsilon) = T_{f+1}(x)+\epsilon$. We claim that $K_\beta^{n,f}(x+\epsilon) < K_\beta^{n,f}(x)$. Indeed, observe that $x < T_{f+1}(x)$ since the point $(x, T_{f+1}(x))$ is inside the cone C_β. In turn this implies

$$\begin{aligned} K_\beta^{n,f}(x+\epsilon) &= \frac{T_{f+1}(x+\epsilon)}{x+\epsilon} = \frac{T_{f+1}(x)+\epsilon}{x+\epsilon} \\ &< \frac{T_{f+1}(x)}{x} = K_\beta^{n,f}(x), \end{aligned}$$

which proves the claim. □

LEMMA 4. *Suppose that robot a_0 visits its turning point τ_0 at time t_0. The time of the first visit of point τ_0 by robot a_{f+1} is*

$$T_{f+1} = \tau_0 \left((\beta+1)^{\frac{2f+2}{n}} (\beta-1)^{1-\frac{2f+2}{n}} + 1 \right). \tag{13}$$

PROOF. Consider robot a_i arriving at time t_i at its first positive turning point τ_i for which $\tau_i > \tau_0$, $i = 1, 2, \ldots, f+1$. By Lemma 2 applied to robot a_i, we have that

$$\begin{aligned} t_i &= t_{i-1} + \tau_{i-1}\beta(r-1) \\ &= t_{i-2} + \tau_{i-2}\beta(r-1) + \tau_{i-1}\beta(r-1) \\ &\vdots \\ &= t_0 + (\tau_0 + \tau_1 + \cdots \tau_{i-1})\beta(r-1) \end{aligned}$$

By Lemma 2 (applied inductively) we have that $\tau_i = r^i\tau_0$ and

$$\begin{aligned} t_i &= t_0 + \tau_0(1 + r + \cdots + r^{i-1})\beta(r-1) \\ &= t_0 + \tau_0\beta(r^i - 1) \\ &= \tau_0\beta r^i \text{ (as } t_0 = \beta\tau_0) \end{aligned}$$

Note that before arriving at its turning point τ_{f+1} at time $t_{f+1} = \tau_0\beta r^{f+1}$, robot a_{f+1} visits point τ_0 at time $T_{f+1} = t_{f+1} - (\tau_{f+1} - \tau_0)$. Consequently, we have

$$T_{f+1} = \tau_0\beta r^{f+1} - \tau_0 r^{f+1} + \tau_0.$$

By Equation (2), we have that $r = \left(\frac{\beta+1}{\beta-1}\right)^{2/n}$. It follows that

$$\begin{aligned} T_{f+1} &= \tau_0\beta r^{f+1} - \tau_0 r^{f+1} + \tau_0 \\ &= \tau_0 \left(r^{f+1}(\beta-1) + 1 \right) \\ &= \tau_0 \left(\left(\frac{\beta+1}{\beta-1} \right)^{\frac{2f+2}{n}} (\beta-1) + 1 \right) \\ &= \tau_0 \left((\beta+1)^{\frac{2f+2}{n}} (\beta-1)^{1-\frac{2f+2}{n}} + 1 \right). \end{aligned}$$

This proves Lemma 4. □

Using the two preceding lemmas we can derive the following upper bound on the competitive ratio of the proportional schedule $S_\beta(n)$.

LEMMA 5. *Consider n robots executing the proportional schedule $S_\beta(n)$. If up to f robots may be faulty where $f < n < 2f+2$ then*

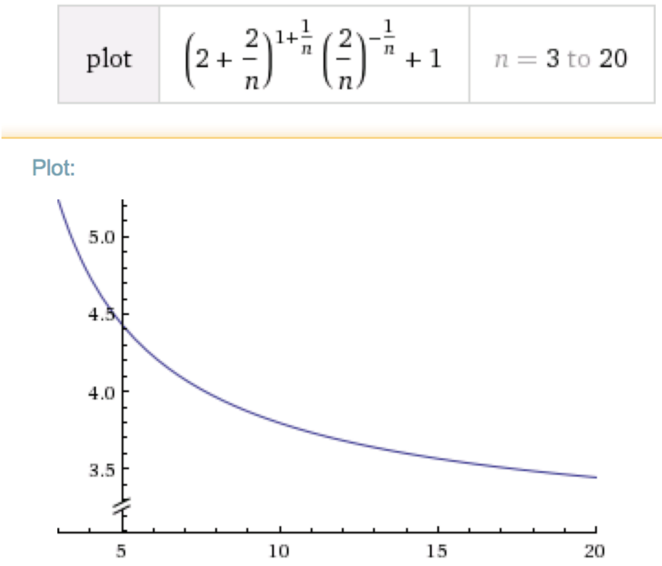
$$CR_\beta^{n,f} = (\beta+1)^{\frac{2f+2}{n}} (\beta-1)^{1-\frac{2f+2}{n}} + 1. \tag{14}$$

PROOF. By Lemma 3, we can subdivide line L into an infinite collection of intervals such that in each such interval the function $K_\beta^{n,f}(x)$ is continuous and decreasing and by Lemma 4, the suprema of the function $K_\beta^{n,f}(x)$ on these intervals are identical. Hence,

$$\begin{aligned} CR_\beta^{n,f} &= \sup_x K_\beta^{n,f}(x) \\ &= (\beta+1)^{\frac{2f+2}{n}} (\beta-1)^{1-\frac{2f+2}{n}} + 1. \end{aligned}$$

This proves Lemma 5. □

We now choose the value of β as a function of f and n so as to minimize the function $F(\beta) := (\beta+1)^{\frac{2f+2}{n}} (\beta-1)^{1-\frac{2f+2}{n}} + 1$, where $\beta > 1$. In turn, this can be optimized



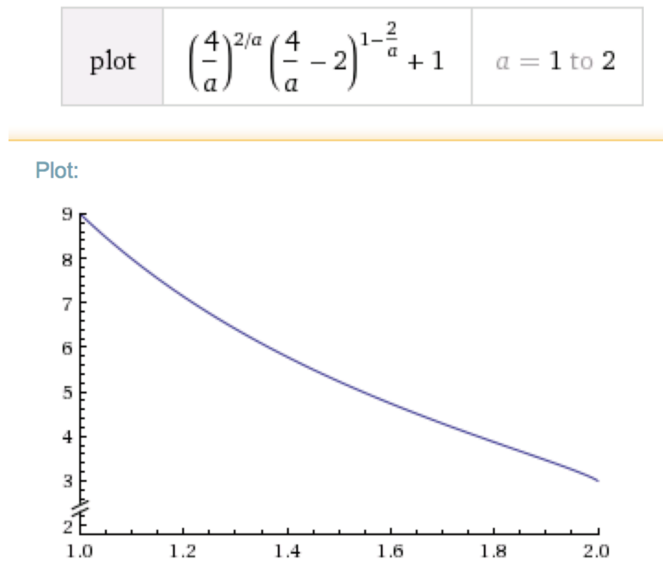


Figure 5: At left: plot of the competitive ratio $\left(2+\frac{2}{n}\right)^{1+\frac{1}{n}}\left(\frac{2}{n}\right)^{-\frac{1}{n}}+1$ of the proportional schedule algorithm for $n = 2f + 1$ robots, f of which are faulty, At right: plot of $\left(\frac{4}{a}\right)^{\frac{2}{a}}\left(\frac{4}{a}-2\right)^{1-\frac{2}{a}}+1$, of the competitive ratio of our proportional schedule algorithm for $n = af$ robots f of which are faulty for $1 < a < 2$.

by taking the derivative of F with respect to β and setting it equal to 0. Indeed, we have that equation $F'(\beta) = 0$ is equivalent to the equation $0 = (\beta+1)^{\frac{2f+2}{n}-1}(\beta-1)^{-\frac{2f+2}{n}}\left(\frac{2f+2}{n}(\beta-1)+\left(1-\frac{2f+2}{n}\right)(\beta+1)\right)$. By solving this last equation for β we get the unique solution $\beta = \frac{4f+4}{n} - 1$.

Next, we show how to convert a proportional schedule $S_\beta(n)$ into an algorithm for the above optimal value of β. Observe that the zig-zag movement of a robot having a turning point x_0 at time $t_0 > 0$ is infinite. It may also be extended backwards in the time interval $(0, t_0)$ by any number of steps. To make it into an algorithm, we need to determine the first part of the trajectory of each robot from the origin to the very first turning point on the cone C_β. The subsequent turning points are then fully determined by the cone. This requires the assumption that either an additive constant should be added to the competitive ratio, or that the minimal distance from the source to the target is known in advance (cf., [35][page 39]). We choose the second option and thus assume that the target is at distance at least one unit from the starting point 0.

Definition 4. *Fix $\beta = \frac{4f+4}{n} - 1$. Set the turning point τ_0 of a_0 to 1, and let $\tau_1, \tau_2, \ldots, \tau_{n-1}$ be the turning points of $a_1, a_2, \ldots, a_{n-1}$ as determined by the proportional schedule $S_\beta(n)$. For any i, $1 \le i \le n-1$ let τ_i' be the first turning point of a_i less than 1 obtained by extending the trajectory of a_i backward inside the cone. Then in the* proportional schedule algorithm *$A(n, f)$, the trajectories of robots are defined as follows:*
Robot a_0 moves from 0 so that it reaches 1 at time β. Then it moves at speed 1 in the cone C_β. For $1 \le i \le n-1$, robot a_i moves from 0 so that it reaches τ_i' at time $\beta\tau_i'$. Then it moves at speed 1 in the cone C_β.

It is easy to observe that according to algorithm $A(n, f)$, at any time $t \ge \beta$, all robots are moving according to the proportional schedule $S_\beta(n)$ for the optimal value of β. We now give the main result of this section.

Theorem 1. *Consider a collection of n robots up to f of which are faulty. Then the competitive ratio of algorithm $A(n, f)$ is at most*

$$\left(\frac{4f+4}{n}\right)^{\frac{2f+2}{n}}\left(\frac{4f+4}{n}-2\right)^{1-\frac{2f+2}{n}}+1. \tag{15}$$

Proof. Our algorithm $A(n, f)$ is based on a proportional schedule $S_\beta(n)$ whose competitive ratio was shown to be at most

$$(\beta+1)^{\frac{2f+2}{n}}(\beta-1)^{1-\frac{2f+2}{n}}+1.$$

Substituting the value of $\beta = \frac{4f+4}{n} - 1$ gives the result. ☐

In the sequel we compute some estimates of the resulting competitive ratio of our proportional schedule algorithms for various values of n and f, where $f < n < 2f + 2$.

For $n = f + 1$, the competitive ratio as given by Formula (15) is

$$\left(\frac{4f+4}{n}\right)^{\frac{2f+2}{n}}\left(\frac{4f+4}{n}-2\right)^{1-\frac{2f+2}{n}}+1 = 4^2(4-2)^{-1}+1 = 9,$$

i.e., the same as for $n = 1$ in [8] and as was observed earlier is optimal.

For $n = 3$ and $f = 1$ we have that $\beta = 8/3 - 1 = 5/3$ and the competitive ratio is $(\beta+1)^{1+\frac{1}{n}}(\beta-1)^{1-\frac{1}{n}}+1 = (8/3)4^{1/3} \sim 5.233$. More generally, for $n = 2f + 1$, the resulting competitive ratio as given by Formula (15) is

$$\left(2+\frac{2}{n}\right)^{1+\frac{1}{n}}\left(\frac{2}{n}\right)^{-\frac{1}{n}}+1,$$

which tends to 3 as $n \to \infty$. The plot of this last function is depicted in Figure 5 (left). However, we can state and prove an even more precise upper bound as follows.

COROLLARY 1. *The optimal competitive ratio of a proportional schedule for $n = 2f + 1$ robots, exactly f of which are faulty, is at most $3 + \frac{4\ln n}{n} + \frac{O(1)}{n}$.*

PROOF. Let $u_n = (n+1)^{1/n}$. Observe, that

$$u_n < \left(1 + \frac{\ln(n+1)}{n}\right)^2 .$$

Indeed,

$$(u_n)^n = n + 1 = e^{\ln(n+1)} < \left(1 + \frac{\ln(n+1)}{n}\right)^{n+\ln(n+1)/2} .$$

The last inequality is based on $e^t < (1 + t/n)^{n+t/2}$ and can be found in [33][page 435]. Therefore,

$$u_n < \left(1 + \frac{\ln(n+1)}{n}\right)^{1+\ln(n+1)/2n} < \left(1 + \frac{\ln(n+1)}{n}\right)^2 .$$

It follows that

$$\begin{aligned}
&\left(2 + \frac{2}{n}\right)^{1+\frac{1}{n}} (2/n)^{-1/n} + 1 \\
&= \left(2 + \frac{2}{n}\right) u_n + 1 \\
&< \left(2 + \frac{2}{n}\right) \left(1 + \frac{\ln(n+1)}{n}\right)^2 + 1 \\
&< \left(2 + \frac{2}{n}\right) \left(1 + 2\frac{\ln(n+1)}{n} + \text{low order terms}\right) + 1 \\
&< 3 + \frac{4\ln n}{n} + \text{low order terms},
\end{aligned}$$

where the low order terms are in $O(1)/n$. This completes the proof of the corollary. □

Now assume that $1 < a := n/f < 2$. The resulting competitive ratio as given by Formula (15) is

$$\left(\frac{4}{a} + \frac{4}{n}\right)^{\frac{2}{a}+\frac{2}{n}} \left(\frac{4}{a} + \frac{4}{n} - 2\right)^{1-\frac{2}{a}-\frac{2}{n}} + 1,$$

which tends to $(4/a)^{2/a}\,(4/a - 2)^{1-2/a} + 1$ as $n \to \infty$. The plot of this last function is depicted in Figure 5 (right).

4. LOWER BOUND

In this section, we give a lower bound for the competitive ratio of any algorithm for search by $n < 2f + 2$ robots, of which f may be faulty. Let $x > 1$. We say that a robot has a *positive trajectory for x* if its *first* visits to the points in $\{-x, -1, 1, x\}$ are in the order $1, x, -1, -x$. Similarly, we say it has a *negative trajectory for x* if its first visits to the points in $\{-x, -1, 1, x\}$ are in the order $-1, -x, 1, x$. See Figure 6 for an illustration of positive and negative trajectories.

We start with a couple of simple observations about positive and negative trajectories, which we will use repeatedly in the proof of our lower bound.

LEMMA 6. *Let $x > 1$ and suppose a single robot visits both points $x, -x$ strictly before time $3x + 2$. Then it has to follow either a positive or a negative trajectory for x.*

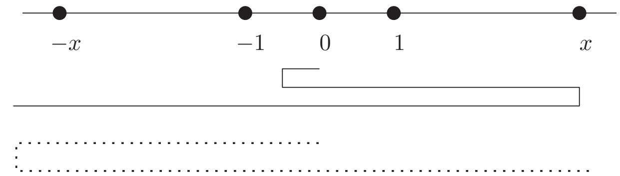


Figure 6: A positive trajectory for x is shown with a solid line: Robot visits $1, x, -1, -x$ in that order. A negative trajectory for x is shown with a dotted line: Robot visits $-1, -x, 1, x$ in that order.

PROOF. The robot has to visit either 1 or -1 first. Suppose it visits 1 before -1. If it visits $-x$ before x, then it takes time at least $3x + 2$ to reach x. Therefore it must first visit x (having visited 1 on the way), then -1 then $-x$, that is, it follows a positive trajectory for x. Similarly, if it visits -1 before 1, it must follow a negative trajectory for $-x$. □

LEMMA 7. *Let $x, y \geq 1$. If a robot follows a positive or negative trajectory for x, then it cannot reach both y and $-y$ before time $2x + y$.*

PROOF. If the robot follows a positive trajectory for x, it must visit x before it visits -1, and therefore needs time at least $2x + y$ to reach $-y$. Similarly if it follows a negative trajectory for x, it needs at least $2x + y$ time to reach y. □

We now present our main result. The key idea is as follows. First, we choose an $x > 1$ such that a single robot is forced to visit both $\pm x$ before a certain time in order for the algorithm to be competitive. Then we use Lemma 6 to show that the robot is forced to take a positive or negative trajectory for $\pm x$. Next we use Lemma 7 to show that the robot cannot possibly visit both $\pm y$ for some $y < x$. We repeat this argument for the other robots to derive a contradiction that the algorithm has the claimed competitive ratio.

THEOREM 2. *The competitive ratio of any algorithm for search by $n < 2f + 2$ robots, of which f are faulty, is at least α for any $\alpha > 3$ satisfying*

$$(\alpha - 1)^n(\alpha - 3) \leq 2^{n+1}$$

PROOF. Fix an α satisfying the conditions of the lemma. Assume the robots start at location 0. Depending on the movement of the robots, the adversary will place the target at one of the points $\pm 1, \pm x_{n-1}, \pm x_{n-2}, \ldots, \pm x_0$, where $x_i = \frac{2^{i+1}}{(\alpha-1)^i(\alpha-3)}$ for every i with $0 \leq i \leq n-1$, Observe that for all i such that $0 \leq i \leq n-2$,

$$x_i = \frac{\alpha - 1}{2} x_{i+1} \tag{16}$$

Furthermore,

$$\begin{aligned}
x_{n-1} &= \frac{2^n}{(\alpha-1)^{n-1}(\alpha-3)} && (17)\\
&= \frac{\alpha-1}{2}\,\frac{2^{n+1}}{(\alpha-1)^n(\alpha-3)} && (18)\\
&> \frac{\alpha-1}{2} && (19)
\end{aligned}$$

where the last inequality follows from the assumption on α.

Since $\alpha > 3$, it follows from Eqns 16 and 19 that:

$$x_0 > x_1 > \dots x_{n-2} > x_{n-1} > 1 \tag{20}$$

See Figure 7 for an illustration. Suppose now, for the pur-

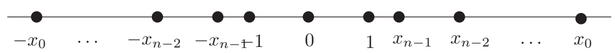


Figure 7: The adversary places the target at a location in $\{\pm 1, \pm x_{n-1}, \dots, \pm x_0\}$, $x_0 > x_1 > x_2 \dots > x_{n-1} > 1$.

pose of contradiction, that there is an algorithm $\mathcal{A}$ that has a competitive ratio $\rho < \alpha$.

First, we claim that for every i such that $0 \le i \le n-1$, at least $f+1$ of the robots must visit both $\pm x_i$ strictly before time αx_i. Suppose instead that at most f robots visit one of these points, say x_i, before time αx_i. The adversary makes them all faulty, and places the target at x_i. Therefore the earliest time a non-faulty robot visits x_i is at least αx_i, which implies that $\rho \ge \alpha$, a contradiction.

Furthermore, since there are at most $2f+1$ robots in all, by the pigeonhole principle, we conclude that there must be a robot that visits *both* $\pm x_i$ before time αx_i. Next we give a proof by induction that for every i with $0 \le i \le n-1$, it must be a *unique* robot that visits both $\pm x_i$ before time αx_i, and furthermore, it must use either a positive or a negative trajectory for x_i.

Inductively, we assume that for some i with $0 \le i < n-1$, and for each j such that $0 \le j \le i$, there is a unique robot a_j that visits both $\pm x_j$ before time αx_j, following either a positive or a negative trajectory for x_j. Now consider the points $\pm x_{i+1}$. By Lemma 7, such a robot a_j cannot visit both $\pm x_{i+1}$ before time $2x_j + x_{i+1}$. However, for all $1 \le j \le i$, since $x_j \ge x_i$, we have

$$\begin{aligned} 2x_j + x_{i+1} &\ge 2x_i + x_{i+1} \\ &= 2\frac{(\alpha-1)x_{i+1}}{2} + x_{i+1} \\ &= \alpha x_{i+1} \end{aligned}$$

where the first equality follows from (16). Therefore, none of the previous robots can visit both $\pm x_{i+1}$ before time αx_{i+1} and it must be a new robot, say a_{i+1}, that visits both $\pm x_{i+1}$ before time αx_{i+1}. Furthermore, since $x_{i+1} < x_0 = 2/(\alpha-3)$, we conclude that $\alpha x_{i+1} < 3x_{i+1}+2$, and so by Lemma 6, a_{i+1} must use a positive or a negative trajectory for x_{i+1}. This completes the proof by induction that there must be a unique robot visiting the points $\pm x_i$ for $0 \le i \le n-1$ before time αx_i, using either a positive or a negative trajectory for x_i.

Finally, consider the points ± 1. Following the same logic as for the points $\pm x_i$, there must be a single robot that visits both ± 1 before time α, otherwise $\rho \ge \alpha$. However, since for each j such that $0 \le j \le n-1$, the robot a_j follows either a positive or a negative trajectory for x_j, it cannot visit both ± 1 before time $2x_j + 1$. Since $x_j \ge x_{n-1}$, we have

$$\begin{aligned} 2x_j + 1 \ge 2x_{n-1} + 1 &= \frac{2^{n+1}}{(\alpha-1)^{n-1}(\alpha-3)} + 1 \\ &= (\alpha-1)\frac{2^{n+1}}{(\alpha-1)^n(\alpha-3)} + 1 \\ &\ge \alpha \end{aligned}$$

since $\frac{2^{n+1}}{(\alpha-1)^n(\alpha-3)} \ge 1$ by assumption. Thus, none of the n robots can visit ± 1 before time α, contradicting the assumption that the competitive ratio of $\mathcal{A}$ is less than α. $\square$

Theorem 2 gives a lower bound of ≈ 3.76 on the competitive ratio of any algorithm for search by 3 robots, of which one may be faulty.

COROLLARY 2. *As n approaches ∞, the competitive ratio of an algorithm for search by $n < 2f+2$ robots, of which f may be faulty, is bounded from below by*

$$3 + \frac{2\ln n}{n} - \frac{2\ln\ln n}{n}.$$

PROOF. Take $\alpha = 3 + 2\frac{\ln n - \ln\ln n}{n}$ and observe that the following inequality holds $\frac{(\alpha-1)^n(\alpha-3)}{2^{n+1}} < 1$. $\square$

5. CONCLUSION

Search on a line has been a source of numerous beautiful problems in robotics and our setting is no exception. In this paper, we analyzed the problem of search on a line by n robots. The novelty of our model is in that it allows $f < n$ robots to be faulty. We presented a new class of parallel search algorithms, called proportional schedule algorithms, and analyzed their competitive ratio. For every (n, f), we give a proportional schedule algorithm $A(n, f)$ whose expansion factor depends on the ratio n/f. For $n = f+1$ our algorithm is optimal. Finally, we showed a lower bound on the competitive ratio of any parallel search algorithm which shows our proportional schedule algorithm is asymptotically optimal when $n = 2f+1$.

6. REFERENCES

[1] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM Journal on Computing*, 36(1):56–82, 2006.

[2] S. Albers and M. R. Henzinger. Exploring unknown environments. *SIAM Journal on Computing*, 29(4):1164–1188, 2000.

[3] S. Albers, K. Kursawe, and S. Schuierer. Exploring unknown environments with obstacles. *Algorithmica*, 32(1):123–143, 2002.

[4] S. Alpern and S. Gal. *The theory of search games and rendezvous*, volume 55. Kluwer Academic Publishers, 2002.

[5] R. Baeza Yates, J. Culberson, and G. Rawlins. Searching in the plane. *Information and Computation*, 106(2):234–252, 1993.

[6] R. Baeza-Yates and R. Schott. Parallel searching in the plane. *Computational Geometry*, 5(3):143–154, 1995.

[7] A. Beck. On the linear search problem. *Israel Journal of Mathematics*, 2(4):221–228, 1964.

[8] A. Beck and D. Newman. Yet more on the linear search problem. *Israel Journal of Mathematics*, 8(4):419–429, 1970.

[9] R. Bellman. An optimal search. *SIAM Review*, 5(3):274–274, 1963.

[10] P. Bose, J.-L. De Carufel, and S. Durocher. Revisiting the problem of searching on a line. In *Algorithms–ESA 2013, LNCS*, volume 8125, pages 205–216. Springer, 2013.

[11] Z. Bouzid, M.G. Potop-Butucaru, and S. Tixeuil. Optimal byzantine-resilient convergence in uni-dimensional robot network. *Theoretical Computer Science*, 411(34-36):3154–3168, 2010.

[12] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386, 2005.

[13] A. Casteigts, P. Flocchini, W. Quattrociocchi, and N. Santoro. Time-varying graphs and dynamic networks. In *Ad-hoc, mobile, and wireless networks, LNCS*, volume 6811, pages 346–359. Springer, 2011.

[14] M. Chrobak, L. Gasieniec, Gorry T., and R. Martin. Group search on the line. In *Proceedings of SOFSEM 2015, LNCS 8939*, pages 164–176. Springer, 2015.

[15] R. Cohen and D. Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM Journal of Computing*, 41(1):1516–1528, 2005.

[16] R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. *SIAM Journal on Computing*, 38(1):276–302, 2008.

[17] J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, and N. Taleb. When patrolmen become corrupted: Monitoring a graph using faulty mobile robots. In *Algorithms and Computation - Proceedings of 26th International Symposium, ISAAC 2015*, pages 343–354, 2015.

[18] X. Défago, M. Gradinariu, S. Messika, and P.R. Parvédy. Fault-tolerant and self-stabilizing mobile robots gathering. In *Proceedings of DISC 2006*, pages 46–60, 2006.

[19] E. D. Demaine, S. P. Fekete, and S. Gal. Online searching with turn cost. *Theoretical Computer Science*, 361(2):342–355, 2006.

[20] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment. In *Proceedings of FOCS*, pages 298–303. IEEE, 1991.

[21] Y. Dieudonné, A. Pelc, and D. Peleg. Gathering despite mischief. *ACM Transactions on Algorithms (TALG)*, 11(1):1, 2014.

[22] O. Feinerman, A. Korman, Z. Lotker, and J. S. Sereni. Collaborative search on the plane without communication. In *Proceedings of the 2012 ACM symposium on Principles of distributed computing*, pages 77–86. ACM, 2012.

[23] P. Flocchini, D. Ilcinkas, A. Pelc, and N. Santoro. How many oblivious robots can explore a line. *Information processing letters*, 111:1027–1031, 2011.

[24] B. Gluss. An alternative solution to the lost at sea problem. *Naval Research Logistics Quarterly*, 8(1):117–122, 1961.

[25] F. Hoffmann, C. Icking, R. Klein, and K. Kriegel. The polygon exploration problem. *SIAM Journal on Computing*, 31(2):577–600, 2001.

[26] J. Hromkovič, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting & gossiping). In *Combinatorial network theory*, pages 125–212. Springer, 1996.

[27] J. R. Isbell. Pursuit around a hole. *Naval Research Logistics Quarterly*, 14(4):569–571, 1967.

[28] T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita. The gathering problem for two oblivious robots with unreliable compasses. *SIAM Journal on Computing*, 41(1):26–46, 2012.

[29] J. Kleinberg. On-line search in a simple polygon. In *Proceedings of SODA*, pages 8–15. SIAM, 1994.

[30] F. Kuhn, N. Lynch, and R. Oshman. Distributed computation in dynamic networks. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 513–522. ACM, 2010.

[31] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

[32] N. A. Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.

[33] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[34] C. H Papadimitriou and M. Yannakakis. Shortest paths without a map. In *Proceedings of ICALP, LNCS*, volume 372, pages 610–620. Springer, 1989.

[35] S. Schuierer. Lower bounds in on-line geometric searching. *Computational Geometry*, 18(1):37–53, 2001.

[36] S. Souissi, X. Défago, and M. Yamashita. Gathering asynchronous mobile robots with inaccurate compasses. *Principles of Distributed Systems*, pages 333–349, 2006.

[37] S. Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5):335–363, 2001.

[38] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of 2nd international conference on Autonomous agents*, pages 47–53. ACM, 1998.

[39] Y. Yang, S. Souissi, X. Défago, and M. Takizawa. Fault-tolerant flocking for a group of autonomous mobile robots. *Journal of Systems and Software*, 84(1):29–36, 2011.