Patrolling a Path Connecting a Set of Points with Unbalanced Frequencies of Visits* 

Huda Chuangpishit¹, Jurek Czyzowicz²**, Leszek Gąsieniec³***, Konstantinos Georgiou¹**, Tomasz Jurdziński⁴†, and Evangelos Kranakis⁵**

¹ Department of Mathematics, Ryerson University, Toronto, Canada, hoda.chuang@gmail.com, konstantinos@ryerson.ca
² Département d’informatique, Université du Québec en Outaouais, Canada, Jurek.Czyzowicz@uqo.ca
³ Department of Computer Science, University of Liverpool, Liverpool, UK, L.A.Gasieniec@liverpool.ac.uk
⁴ Instytut Informatyki, Uniwersytet Wrocławski, Wrocław, Poland, Tomasz.Jurdzinski@ii.uni.wroc.pl
⁵ School of Computer Science, Carleton University, Ottawa, Canada, evankranakis@gmail.com

Abstract. Patrolling consists of scheduling perpetual movements of a collection of mobile robots, so that each point of the environment is regularly revisited by any robot in the collection. In previous research, it was assumed that all points of the environment needed to be revisited with the same minimal frequency.

In this paper we study efficient patrolling protocols for points located on a path, where each point may have a different constraint on frequency of visits. The problem of visiting such divergent points was recently posed by Gąsieniec et al. in [13], where the authors study protocols using a single robot patrolling a set of \( n \) points located in nodes of a complete graph and in Euclidean spaces.

The focus in this paper is on patrolling with two robots. We adopt a scenario in which all points to be patrolled are located on a line. We provide several approximation algorithms concluding with the best currently known \( \sqrt{3} \)-approximation.

1 Introduction

In this paper we study efficient patrolling protocols by two robots for a collection of \( n \) points distributed arbitrarily on a path or a segment of length 1. Each

* This is the full version of the paper with the same title which will appear in the proceedings of SOFSEM 2018, 44th International Conference on Current Trends in Theory and Practice of Computer Science, January 29 - February 2, 2018, Krems an der Donau, Austria.

** Research supported in part by NSERC.

*** This work was in part supported by Networks Sciences and Technologies (NeST).

† Research supported by the Polish National Science Centre grant DEC-2012/06/M/ST6/00459.
point needs to be attended perpetually with known but often distinct minimal frequency, i.e., some points need to be visited more often than others.

The problem was recently studied in [13] where a collection of \( n \) points was monitored with use of a single mobile robot. The points to be patrolled in [13] are located in nodes of a complete graph with edges of uniform (unit) length, as well as in Euclidean spaces, where the points are distributed arbitrarily. In their work the frequency constraints refer to urgency factors \( h_i \), meaning that during a unit of time the urgency of point \( p_i \) grows by an additive term \( h_i \), and the task is to design a schedule of perpetual visits to nodes which minimizes the maximum ever observed urgency on all points. In complete graphs and for any distribution of frequencies (urgency factors) the authors of [13] proposed a \( 2 \)-approximation algorithm based on a reduction to the pinwheel scheduling problem, see, e.g., [6,14,15,18]. They also discuss more tight approximations for the cases with more balanced urgency factors. In Euclidean spaces [13] proposes several lower bounds and concludes with an \( O(\log n) \)-approximation for an arbitrary distribution of points and urgency factors.

In our formulation, we assume that both robots have unit speed, and we try to minimize the relative violation of visitation-frequency requirements, i.e. the worst case time between two visitations over the required largest waiting time of each point. Equivalently, one may think of the problem of finding the minimum possible speed \( s \) that both robots should patrol with that induces no violation for the visitation-frequency requirements. In such setting, our patrolling result refers naturally to a competitive ratio, which is defined by the ratio of the speed the robots attain in our algorithm divided by the speed in the optimal solution.

Specific to our model is the use of two robots, for which, as we show, one can achieve \( \sqrt{3} \)-approximation patrolling schedules. Notably, and maybe counter-intuitively, reducing the number of robots from two to one does not lead to constant approximation. An instructive example is when the central point has a very large visiting frequency (we can dedicate one robot to this point) comparing to the rest of the points on the line.

In the previous research on boundary and fence patrolling (cf. [10,11]) all points of the patrolled environment were supposed to be revisited with the same frequency. However, assigning different importance to distinct portions of the monitored boundary seems natural and observable in practice. A particular variation of this problem was studied in [9], where the authors focus on monitoring vital (possibly disconnected) parts of a linear environment, while the remaining neutral portions of the boundary need not be attended at all.

The problem of distinct attendance assigned to different portions of the environment, while of inherent combinatorial interest, is also observed in perpetual testing of virtual machines in cloud systems [1]. In such systems the frequency with which virtual machines are tested for undesirable symptoms may vary depending on the importance of dedicated cloud operational mechanisms.

The problem studied here is also a natural extension of several classical combinatorial and algorithmic problems referring to monitoring and mobility. This includes the Art Gallery Problem [19,20] and its dynamic variant called the \( k-\)
Watchmen Problem. In a more recent work on fence patrolling the efficiency of patrolling is measured by the idleness of the protocol, which is the time where a point remains unvisited (maximized over all time moments and all points of the environment). In one can find a study on monitoring of linear environments by robots prone to faults. In the authors assume robots have distinct maximum speeds which makes the design of patrolling protocols more complex, in which case the use of some robots becomes obsolete.

In a very recent work Liang and Shen consider a line of points attributed with uniform urgency factors. For robots with uniform speeds, they give a polynomial-time optimal solution, and for robots with constant number of speeds they present a 2-approximation algorithm. For an arbitrary number of velocities they design a 4-approximation algorithm, which can be extended to a $2\alpha$-approximation algorithm family scheme, where integer $\alpha > 1$ is the tradeoff factor to balance the time complexity and approximation ratio.

2 Problem Statement & Definitions

An instance of the Path Patrolling Problem of Points with Unbalanced Frequencies (PUF) consists of points $S = \{y_i\}_{i=1,...,n}$ in the unit interval, where $0 = y_1 < y_2 < \ldots < y_n = 1$. Each point $y_i$ is associated with its idleness time $I(y_i) \in \mathbb{R}^+$, a positive real number which is also referred to as visitation frequency requirement.

A perpetual movement schedule of two robots $r_1, r_2$ of speed 1 will be referred to as a patrolling schedule (robots may change movement direction instantaneously, and at no cost). Given a patrolling schedule $A$, we define the waiting time $w_A(y_i)$ of each point $y_i$ as the supremum of the time difference between any two subsequent visitations by any of $r_1, r_2$. When the patrolling schedule is clear from the context, we will drop the subscript in $w_A$. A patrolling schedule $A$ is called feasible if for all $i$, $w_A(y_i) \leq I(y_i)$. Schedule $A$ is called $c$-feasible, or $c$-approximation, if $w_A(y_i)/I(y_i) \leq c$, for each $i = 1, \ldots, n$. Thus a feasible patrolling schedule is also 1-approximation, or 1-feasible.

An instance of PUF that admits a feasible patrolling schedule will be called feasible. In this paper we are concerned with the combinatorial optimization problem of minimizing the worst (normalized) violation of the idleness times for feasible instances, i.e., we are concerned with finding good approximation patrolling schedules, in which robots’ trajectories can be determined efficiently in the size of the given input. We will call such patrolling schedules efficient.

The problem considered here is a close relative of Pinwheel scheduling modeled by points with non-uniform deadlines (visitation-frequencies) spanned by a complete network with edges of uniform length. The complexity of Pinwheel scheduling depends on its representation. In particular we know that in the standard multi-set representation the problem is in NP, however, we still don’t known whether it is NP-hard. One can try to get closer to this answer either by studying particular instances of the problem which can be decided or instead by seeking approximate solutions. In this paper we adopted the latter.
We use the following concepts in the analysis of our patrolling schedules. We associate each point \( y_i \) with its range defined as the closed intervals 
\[
R(y_i) = \left[ \max\left\{ 0, y_i - \frac{f(y_i)}{2} \right\}, \min\left\{ 1, y_i + \frac{f(y_i)}{2} \right\} \right].
\]
Intuitively, \( R(y_i) \) is the ball around \( y_i \) within which a robot can be moving introducing no violation to the visitation frequency requirement of \( y_i \). We also group points \( y_i \) with respect to whether the extreme points fall within their range, i.e., we introduce:
\[
S_{00} := \{ y_i \in S : 0, 1 \not\in R(y_i) \}, S_{01} := \{ y_i \in S : 0 \not\in R(y_i) \not\ni 1 \}, S_{10} := \{ y_i \in S : 0 \in R(y_i) \not\ni 1 \}, S_{11} := \{ y_i \in S : 0, 1 \in R(y_i) \}.
\]

3 Summary of Results & Paper Organization

Our main contribution pertains to efficient patrolling schedules (algorithms) of feasible PUF instances. In particular, the patrolling schedules we propose are highly efficient and simple, meaning that robots’ trajectories can be determined by a few critical turning points, which can be computed in linear time in the number of points of the PUF instance. In order to do so, we provide in Section 4 some useful properties that all feasible PUF instances exhibit, and in particular a characterization of instances with “no problematic points”. For the latter instances, we also provide optimal feasible schedules (Theorem 1). Then we turn our attention to arbitrary feasible PUF instances. As a warm-up, we present in Section 5 a simple efficient 4-approximation patrolling schedule that does not require coordination between robots. Section 6 is devoted to the introduction of an elaborate and efficient \( \sqrt{3} \)-approximation patrolling schedule. The execution of the patrolling schedule requires robots to remember at most two special turning points (that can be found efficiently), and, in some cases, their coordination so that they never come closer than a predetermined critical distance. Its performance analysis is based on further properties of feasible PUF instances that are presented in Section 6.1. In particular, the \( \sqrt{3} \)-feasible patrolling schedule is the combination of Algorithms 1 and 2, presented in Sections 6.2 and 6.3 respectively, each of them performing well for a different spectrum of a special structural parameter of the given instance that we call expansion. Finally in Appendices A, B we also show that the analyses we provide for all our proposed algorithms are actually tight.

4 Characterization of (Some) Feasible PUF Instances

In this section we characterize feasible instances of PUF for which at least one of the extreme points falls within the range of each point.

**Theorem 1.** An instance of PUF with \( S_{00} = \emptyset \) is feasible if and only if the following conditions are satisfied:

1. \( S_{10} \subset \bigcap_{x \in S_{01}} R(x) = X_{10}, \) and \( 0 \in X_{10}. \)
2. \( S_{01} \subset \bigcap_{x \in S_{01}} R(x) = X_{01}, \) and \( 1 \in X_{01}. \)
(3) \( S \subset \bigcap_{x \in S_{10}} R(x) \cup \bigcap_{x \in S_{01}} R(x) = X_{10} \cup X_{01} \)

Moreover, if conditions (1)-(3) are satisfied, then there exists an efficient 1-approximation partition-based patrolling schedule, i.e. a schedule in which every \( y_i \) is visited only by one robot.

In order to prove Theorem 1, we need a few observations.

**Observation 1** Assume \( A \) is a feasible patrolling schedule. Then, for each \( x \in S \) and each time window of length at least \( \frac{I(x)}{2} \) during an execution of \( A \), at least one robot is in \( R(x) \).

**Proof.** Reset time to \( t_0 = 0 \). Aiming at contradiction, assume there is no robot in \( R(x) \) at \( t \geq \frac{I(x)}{2} \). Since both robots have speed 1, no robot visited \( x \) in the period \( [t - \frac{I(x)}{2}, t] \) and no robot is able to visit \( x \) in the period \( [t, t + \frac{I(x)}{2}] \). Thus, \( A \) is not a feasible patrolling schedule. \( \square \)

For simplicity, we may also assume that in any patrolling schedule (hence in feasible schedules as well), the position of robot \( r_1 \) in the unit interval is always to the left of the position of \( r_2 \), as otherwise we can exchange the roles of the robots whenever they swap while they meet. We summarize as follows.

**Observation 2** In any patrolling schedule of PUF, \( r_1 \) (\( r_2 \)) is the only robot patrolling \( y_1 = 0 \) (\( y_n = 1 \)), and \( r_1 \) stays always to the left of \( r_2 \).

We are now ready to prove Theorem 1.

**Proof (Theorem 1).** First, we show implication \( (\Rightarrow) \) by contraposition. If Condition (1) is not satisfied, then there exists \( x \in S_{10} \) such that \( x \not\in X_{10} \). Fix a feasible schedule \( A \). By Observation 1, we may assume that \( r_1 \) stays to the left of \( r_2 \), throughout the execution of the schedule. By Observation 1, there must be a robot in \( X_{10} \) at each time \( t \). Thus, \( r_1 \) must be in \( X_{10} \) at each time \( t \). Consequently, \( x \in S_{10} \setminus X_{10} \) is visited only by \( r_2 \). But \( r_2 \) has to visit point \( y_n = 1 \), and by definition of \( S_{10} \) we know that \( 1 \not\in R(x) \). Therefore, \( A \) is not a feasible schedule. By definition of \( S_{10} \), for all \( x \in S_{10} \), we have \( 0 \in R(x) \). Therefore \( 0 \in X_{10} \). A similar argument proves that Condition (2) is satisfied.

By (1) and (2), there exist \( a, b \in (0, 1) \) such that \( X_{10} = [0, a] \) and \( X_{01} = [b, 1] \). Now suppose that Condition (3) is not satisfied. Then \( a < b \), and there is a point \( x \in S \) such that \( a < x < b \), and therefore neither \( r_1 \) nor \( r_2 \) can visit \( x \).

For implication \( (\Leftarrow) \), assume that (1)-(3) are satisfied. Consider a partition traversal \( A \), where \( r_1 \) is searching \( X_{10} \setminus X_{01} \) and \( r_2 \) is searching \( X_{01} \). Then, by the definition of the ranges \( R(x) \), \( X_{10} \) and \( X_{01} \), the traversal \( A \) is feasible. \( \square \)

The complication of instances when \( S_{00} \) is non-empty is that in a feasible solution, points in \( S_{00} \) have to be interchangeably patrolled by both \( r_1, r_2 \), which further requires appropriate synchronization between them. Even though a characterization of feasibility for such instances is eluding us, we provide below a necessary condition. This condition will be useful also later on.
Lemma 1. For every feasible instance of PUF, we have $S_{00} \subset \bigcap_{x \in S} R(x)$.

Proof. Suppose to the contrary, that there are $x \in S_{00}$ and $y \in S$, such that $x \notin R(y)$. By Observation 1, a robot is always present inside $R(y)$. Therefore the other robot must visit $x$. Without loss of generality assume that $y < x$. The robot that visits $y$ cannot pass the point $y + \frac{I(y)}{2} < x$. Also the robot that visits $x$ cannot pass the point $x + \frac{I(x)}{2}$. Since $x \in S_{00}$ then $x + \frac{I(x)}{2} < 1$. This means that no robot can visit point $y_n = 1$. □

5 A Simple 4-Approximation Patrolling Schedule

In light of Theorem 1, it is interesting to study feasible instances of PUF that may have points that cannot be patrolled by one robot, i.e. for which $S_{00} \neq \emptyset$. As a warm-up, we provide a 4-feasible patrolling schedule for such instances. The advantage of this schedule is that robots’ trajectories are simple and no coordination is required.

Theorem 2. Feasible instances of PUF admit an 4-approximate patrolling schedule.

Proof (Theorem 2). Let $A$ be a feasible solution. Let $I = \min_{y \in S} I(y)$ and let $x \in S$ be such that $I(x) = I$. If $I \geq \frac{1}{4}$, then one robot patrolling the interval $[0,1]$ gives a 4-approximation solution. Thus, we may assume that $I \leq \frac{1}{4}$.

According to Observation 1, at least one robot stays in $R(x)$ during $A$, at each time $t$. We claim that a nested traversal $A$ in which one robot traverses $[0,1]$ and the other robot traverses $R(x)$ is a 4-approximation.

We split the interval $[0,1]$ into $A = [0,a]$, $R(x) = [a,a + I] = [a,1 - b]$ and $B = [1 - b,1]$, where $a + I + b = 1$. First, note that the waiting time of each $y \in R(x)$ during $A$ is $w_A(y) = 2I = 2I(x) \leq 2I(y)$. Thus, it remains to show that $w_A(y) \leq 4I(y)$ for each point $y \in A \cup B$.

Without loss of generality assume that $|A| = a < b = |B|$. Using the assumption $I \leq \frac{1}{4}$ and $a + I + b = 1$, we have $a + b \geq \frac{1}{2}$, and therefore $b \geq \frac{1}{4}$. Using Observation 2 we consider a feasible schedule $B$ in which $r_1$ is always to the left of $r_2$. By Observation 1 at least one robot stays in $R(x)$ at each time during $B$. We consider the following cases:

(Case $y \in A$): As at each time moment there must exist a robot in $R(x)$, then in $B$ robot $r_1$ has to stay in $R(x)$ while $r_2$ is traversing $B = [1 - b,1]$ twice to visit $y_n = 1$ and return to $R(x)$. Therefore the waiting time $w_B$ satisfies $I(y) \geq w_B \geq 2b \geq 2 \cdot \frac{1}{4} = \frac{1}{2}$. On the other hand $w_A(y) = 2 = 4 \frac{1}{2} \leq 4I(y)$.

(Case $y \in B$): Let $y' = y - (a + I)$, thus $y'$ is the distance of $y$ to $R(x)$. Consider a time $t$ during the execution of $B$ at which $r_1$ leaves $R(x)$ in order to visit the point 0. As $r_2$ must be in $R(x)$ at $t$, the last visit of $y$ before $t$ was at time $t' \leq t - y'$. Then, it has to stay in $R(x)$ for at least $2a + y'$. The time between two consecutive visits at $y$ is at least $t + 2a + y' - \left(t - y'\right) = 2a + 2y'$. On the other hand, in order to visit 1, $r_2$ has also time at least $2\left(1 - y'\right)$ between two
consecutive visits of $y$. Altogether $w_B(y) \geq \max\{2(a + y'), 2(b - y')\}$. Thus $w_B(y) \geq \frac{1}{2}[2(a + y') + 2(b - y')] = a + b \geq \frac{1}{2}$. On the other hand $w_A(y) = 2$ and thus $w_A(y) \leq 4w_B \leq 4I(y)$. 

6 A \sqrt{3}\text{-Approximation Patrolling Schedule}

The bottleneck toward patrolling instances of PUF is caused by points which require the coordination of both robots in order to be patrolled, i.e. instances in which $S_{00} \neq \emptyset$. In order to improve upon the 4-feasible schedule of Theorem 2 we need to understand better the visitation requirements of points in $S_{00}$, as well as their relative positioning in the path to be patrolled. The result of our analysis, and our main contribution, is an elaborate $\sqrt{3}$-feasible patrolling schedule.

**Theorem 3.** Feasible instances of PUF admit an efficient $\sqrt{3}$-approximate patrolling schedule.

In what follows, we explicitly assume that $S_{00} \neq \emptyset$, as otherwise, due to Theorem 1 we can easily find feasible schedules for instances of PUF that admit feasible solutions. Next, we introduce a key notion to our algorithms.

**Definition 1.** Given an instance of PUF we identify critical points $x_1, \ldots, x_4$ that are defined as follows: $\bigcap_{x \in S_{00}} R(x) = [x_1, x_4]$, and $x_2, x_3$ are the leftmost and rightmost points point in $S_{00}$, respectively. The instance is called $\alpha$-expanding if $x_1 = \frac{\alpha}{1+\alpha}x_4$.

**Theorem 3** is an immediate corollary of the following Lemmata 2, 3 that we prove in subsequent Sections 6.2, 6.3, respectively. Thelemmata are interesting in their own right, since they explicitly guarantee good approximate schedules as a function of the expansion of the given instance.

**Lemma 2.** Feasible $\alpha$-expanding instances of PUF admit an efficient $(1+2\alpha)$-approximate patrolling schedule.

**Lemma 3.** Feasible $\alpha$-expanding instances of PUF admit an efficient $\frac{2+\alpha}{1+\alpha}$-approximate patrolling schedule.

**Lemma 2** above imply that any feasible $\alpha$-expanding instance admits a $\min\{1 + 2\alpha, \frac{2+\alpha}{1+\alpha}\}$ feasible patrolling schedule. The achieved approximation is the worst when the instance is $\frac{\sqrt{3} - 1}{2}$-expanding, in which case, the patrolling schedule is $\sqrt{3}$-feasible. This concludes the proof of Theorem 3.

Notably, our feasibility bounds above are tight. In Appendices A and B we show that for every $\alpha$, there are feasible $\alpha$-expanding PUF instances for which the performance of our patrolling schedules that prove Lemma 2 and Lemma 3 (see Sections 6.2, 6.3) is equal to the proposed bound. Hence, the performance analysis of our patrolling schedule showing Theorem 3 cannot be improved.
6.1 Useful Observations for Feasible PUF Instances

In an \(\alpha\)-expanding instance of PUF we have that \(x_1 = \alpha(x_4 - x_1)\). If the instance is also feasible, then by Lemma 1 we have that \(S_{00} \subseteq \bigcap_{x \in S} R(x)\). Since \(S_{00} \subseteq S\), we obtain that \(S_{00} \subseteq \bigcap_{x \in S_{00}} R(x) = [x_1, x_4]\). Also, it is easy to see that for the critical points \(x_1, \ldots, x_4\) we have that \(x_1 \leq x_2 < x_4\) and that \(x_1 < x_3 \leq x_4\). In particular we may assume, without loss of generality, that \(x_1 < 1 - x_4\), as otherwise we flip the order of all points. Also using Observation 2 we assume that the feasible schedule to the PUF instance has robot \(r_1\) stay always to the left of \(r_2\).

**Lemma 4.** Consider a feasible patrolling schedule \(A\) for a PUF instance. Then

1. there is always a robot inside the interval \([x_1, x_4]\).
2. the interval \([0, x_1]\) is only traversed by \(r_1\) and the interval \((x_4, 1]\) is only traversed by \(r_2\).
3. \(0 \in R(x)\) for all \(x \in [0, x_1]\), and \(1 \in R(x)\) for all \(x \in (x_4, 1]\).
4. \(x_4 - x_3 \leq x_3 - x_1\) and \(x_2 - x_1 \leq x_4 - x_1\).

**Proof.** The proof of (1) is a direct consequence of Observation 1 and the fact that \([x_1, x_4]\) is the intersection of the ranges of all of the points of \(S_{00}\).

During the execution of \(A\) a robot needs to visit 0 and 1. Also, by (1) we know that there is always a robot inside \([x_1, x_4]\). Therefore while the robot \(r_2\) is traversing \((x_4, 1]\) the robot \(r_1\) has to stay inside \([x_1, x_4]\), and while robot \(r_1\) is traversing \([0, x_1]\) the robot \(r_2\) has to stay inside \([x_1, x_4]\). This implies that \(r_1\) never passes \(x_4\) and \(r_2\) never passes \(x_1\). This proves (2). Part (3) follows directly from (2).

We now prove the first inequality of (4). Suppose to the contrary that \(x_4 - x_3 > x_3 - x_1\), and thus \(x_3 < \frac{x_1 + x_4}{2}\). For all \(x \in S_{00}\) we have that \(x_4 \in R(x)\).

Therefore for all \(x \in S_{00}\), \(x_4 \leq x + \frac{I(x)}{2}\). Moreover \(x_3\) is the rightmost point of \(S_{00}\), hence \(x \leq x_3 < \frac{x_1 + x_4}{2}\). Consequently \(x_4 \leq x + \frac{I(x)}{2} \leq x_3 + \frac{I(x)}{2} < \frac{x_1 + x_4}{2} + \frac{I(x)}{2}\).

This implies that \(I(x) > \frac{x_4 - x_3}{2}\). So for all \(x \in S_{00}\) we have \(x - \frac{I(x)}{2} < x - \frac{x_4 - x_3}{2} = x_1\). Therefore there is a point \(y \in (0, 1)\) such that for all \(x \in S_{00}\), \(x - \frac{I(x)}{2} \leq y < x_1\). Hence \(y \in \bigcap_{x \in S_{00}} R(x)\) and \(y < x_1\). This contradicts the fact that \(x_1\) is the leftmost point of the intersection of the ranges of all the points of \(S_{00}\). The proof of the second inequality of (4) follows by an analogous argument.

\(\square\)

**Lemma 5.** If there is a feasible solution for patrolling with two robots then the idle time of the points of \(S\) satisfy the following inequalities.

\[
I(x) \geq \begin{cases} 
\max\{2x, 2(1 - x - x_4 + x_1), x_4 - x_1\} & x \in [0, x_1) \\
2\max\{x_4 - x, x - x_1\} & x \in [x_1, x_4] \\
\max\{2(1 - x), 2(x - x_4 + x_1), x_4 - x_1\} & x \in (x_4, 1]\end{cases}
\]

**Proof.** Let \(A\) be a feasible solution and \(x \in S\).

First assume that \(x \in [0, x_1)\). By (2) of Lemma 4 in \(A\) the points of \([0, x_1)\) are only visited by \(r_1\) and \(0 \in R(x)\). Thus, \(I(x) \geq w_A(x) \geq 2x\). Moreover robot
Lemma 5. \( w \) has \( x \) to stay inside the interval \([x_1, x_4]\) for at least \(2(1 - x_4)\) while the robot \( r_2 \) is traversing the interval \([x_4, 1]\) to visit 1. The time length for \( r_1 \) to traverse from \( x \) to \( x_1 \), stay for at least \(2(1 - x_4)\) inside \([x_1, x_4]\), and then traverse from \( x_1 \) to \( x \) is at least \(2[(x_1 - x) + (1 - x_4)]\). Therefore, \( I(x) \geq w_{4(x)} \geq 2[(x_1 - x) + (1 - x_4)] \).

On the other hand, by Lemma 1, we know that \( x_3 \in R(x) \), and thus \( \frac{I(x)}{2} \geq x_3 - x \geq x_3 - x_1 \). By (3) of Lemma 4, \( x_4 - x_3 \leq x_3 - x_1 \leq \frac{I(x)}{2} \). Therefore, \( x_4 - x_1 = (x_4 - x_3) + (x_3 - x_1) \leq \frac{I(x)}{2} + \frac{I(x)}{2} = I(x) \). By the above discussion, and for all \( x \in [0, x_1] \), we have \( I(x) \geq \max\{2x, 2(1 - x - x_3 + x_1), x_4 - x_1\} \). A similar argument shows that for \( x \in (x_4, 1) \) we have that \( I(x) \geq \max\{2x, 2(1 - x - x_4 + x_1), x_4 - x_1\} \).

Now assume that \( x \in [x_1, x_4] \). Then \( x_1, x_4 \in R(x) \), and therefore \( x - \frac{I(x)}{2} \leq x_1 \leq x_4 \leq x + \frac{I(x)}{2} \). This implies that \( 2(x - x_1) \leq I(x) \) and \( 2(x_4 - x) \leq I(x) \). So for all \( x \in [x_1, x_4] \) we have \( I(x) \geq \max\{x - x_1, x_4 - x\} \).

6.2 \((1 + 2\alpha)\)-Approximate Patrolling Schedules (Proof of Lemma 2)

Given a feasible \( \alpha \)-expanding instance of PUF and using its critical points as in Definition 1, we propose the following algorithm.

**Algorithm 1**

1: Robot \( r_1 \) starts anywhere in \([0, x_3]\), and robot \( r_2 \) starts anywhere in \([x_3, 1]\).
2: Repeat forever
3: Robot \( r_1 \) zigzags inside \([0, x_3]\) and robot \( r_2 \) zigzags inside \([x_3, 1]\).

Next we show that Algorithm 1 is \((1 + 2\alpha)\)-feasible, effectively proving Lemma 2. For this we analyze the waiting time \( w(x) \) for all points \( x \in S \).

Assume that \( x \in [0, x_1] \). By Lemma 1 we know that \( x_3 \in R(x) \). Moreover by (3) of Lemma 4 \( 0 \in R(x) \). Since \( r_1 \) zigzags inside \([0, x_3]\) then \( w(x) \leq I(x) \).

Similarly, for \( x \in (x_3, 1]\), by Lemma 1 and Lemma 4 we have \( \{x_3, 1\} \subset R(x) \). Since \( r_2 \) zigzags inside \([x_3, 1]\) then \( w(x) \leq I(x) \).

Finally, let \( x \in [x_1, x_3] \). First assume that \( x \leq x_3 \). Then in Algorithm 1 the point \( x \) is only visited by \( r_1 \). Since \( r_1 \) zigzags inside \([0, x_3]\) we have that \( w(x) = 2\max\{x, x_3 - x\} \). We now compute the feasibility ratio. Clearly for the points \( x \in [0, x_1] \cup (x_3, 1] \) we have that \( \frac{w(x)}{I(x)} \leq 1 \). So when \( x \in [x_1, x_3] \), then by Lemma 3 \( \frac{w(x)}{I(x)} \leq \frac{\max\{x, x_3 - x\}}{\max\{x - x_1, x_4 - x\}} \). First let \( \max\{x - x_1, x_4 - x\} = x_4 - x \). Then \( x \leq \frac{x_1 + x_4}{2} \). If \( x \leq \frac{x_3 - x_1}{2} \), as \( x_3 \leq x_4 \) we have that \( \frac{w(x)}{I(x)} \leq 1 \). If ,
on the other hand, \( \max\{x, x_3 - x\} = x \), then we have

\[
\frac{w(x)}{I(x)} \leq \frac{x}{x_4 - x} \leq \frac{x_1 + x_4}{x_4 - x_1} = \frac{(x_4 - x_1) + 2x_1}{x_4 - x_1} = 1 + \frac{2x_1}{x_4 - x_1}
\]

\[
= 1 + \frac{2x_1}{x_4 - x_1} \quad \text{[Using } x_1 = \alpha(x_4 - x_1)\]}

\[
= 1 + 2\alpha.
\]

Now let \( \max\{x - x_1, x_4 - x\} = x - x_1 \). Then \( x \geq \frac{x_1 + x_4}{2} \). Moreover by (4) of Lemma 4 we have \( x_3 \geq \frac{x_1 + x_4}{2} \). Therefore \( x_3 - x \leq x - x_1 \). If \( \max\{x, x_3 - x\} = x_3 - x \) then \( \frac{w(x)}{I(x)} \leq 1 \). So assume that \( \max\{x, x_3 - x\} = x \), in which case

\[
\frac{w(x)}{I(x)} \leq \frac{x}{x - x_1} \leq \frac{x - x_1 + x_1}{x - x_1} = 1 + \frac{x_1}{x - x_1}
\]

\[
\leq 1 + \frac{x_1}{x_1 + x_4 - x_1} = 1 + \frac{2x_1}{x_4 - x_1} = 1 + \frac{2x_1}{\frac{x_4}{\alpha}} = 1 + 2\alpha.
\]

Therefore, Algorithm 4 is a \((1 + 2\alpha)\)-approximation algorithm. Our analysis above is tight. For details, see Appendix A.

### 6.3 \( \frac{2+\alpha}{1+\alpha} \)-Approximate Patrolling Schedules (Proof of Lemma 3)

The distributed algorithm that achieves feasibility performance \( \frac{2+\alpha}{1+\alpha} \) is quite elaborate. At a high level, the two robots maintain some distance that never drops below a certain carefully chosen threshold. During the execution of the patrolling schedule, there will always be some robot patrolling (zigzagging within) a certain subinterval defined by critical points of the given instance. When the robots move towards each other, and their distance reaches the certain threshold, then robots exchange roles; the previously zigzagging robot abandons the subinterval and goes to patrol its extreme point, while the other robot starts zigzagging within the subinterval. The formal description of our algorithm follows. The reader may also consult Figure 1.

Fig. 1. The red arrow determines the patrolling area of \( r_1 \) and the blue arrow determines the patrolling area of \( r_2 \).
Algorithm 2

1: Let \( d = \frac{1}{1+\alpha} \min\{x_1, x_4 - x_1\} \).
2: Robot \( r_1 \) starts at \( x_1 - d \) and robot \( r_2 \) at \( x_1 \).
3: Repeat forever
4: \hspace{1em} \textit{Patrolling Schedule of} \( r_1 \):
5: \hspace{2em} \textbf{while} \( r_1 \) is inside the interval \([x_1, x_4]\) and the distance between the locations of \( r_1 \) and \( r_2 \) is at least \( d \) \textbf{do}
6: \hspace{3em} Zigzag between points \( x_1 \) and \( x_4 \).
7: \hspace{2em} Visit point 0, then visit point \( x_1 \), and then go to step 5.
8: \hspace{1em} \textit{Patrolling Schedule of} \( r_2 \):
9: \hspace{2em} \textbf{while} \( r_2 \) is inside the interval \([x_1, x_4]\) and the distance between the locations of \( r_2 \) and \( r_1 \) is at least \( d \) \textbf{do}
10: \hspace{3em} Zigzag between points \( x_1 \) and \( x_4 \).
11: \hspace{2em} Visit point 1, then visit point \( x_4 \), and then go to step 9.

Note that each robot has an explicit segment in which the points are visited by only that robot, i.e. \([0, x_1]\) is the explicit segment of \( r_1 \) and \([x_4, 1]\) is the explicit segment of \( r_2 \). The trajectories of the robots overlap at \([x_1, x_4]\) where the points are visited by both \( r_1 \) and \( r_2 \). The movements of the robots have two states: zigzagging inside \([x_1, x_4]\) and traversing their explicit segments twice. More precisely, once a robot enters \([x_1, x_4]\) it zigzags inside \([x_1, x_4]\) until the other robot is at distance \( d \). Then it leaves \([x_1, x_4]\), traverses its explicit segment twice, and the same process repeats perpetually.

Note that by the definition of \( d \), we know that \( \min\{x_1, x_4 - x_1, 1 - x_4\} \geq d \). Therefore, the original placement of \( r_1 \) at \( x_1 - d \) is compatible with Algorithm 2. The remaining of the section is devoted to proving that Algorithm 2 is \( \frac{2+\alpha}{1+\alpha} \) approximate, effectively proving Lemma 3. As a first step, we calculate the worst case waiting times \( w(x) \) of all points in \( S \).

**Lemma 6.** The waiting times of points in \( S \) for Algorithm 2 are as follows.

\[
\begin{aligned}
    w(x) & \begin{cases} 
        = 2\max\{1 - x - d\}, & x \in [0, x_1) \\
        \leq 2\max\{x - x_1, x_4 - x\} + d, & x \in [x_1, x_4] \\
        = 2\max\{1 - x, x - d\}, & x \in (x_4, 1] 
    \end{cases} \\
\end{aligned}
\]

**Proof.** Recall that \( x_1 \leq 1 - x_4 \), and in particular \( \min\{x_1, x_4 - x_1, 1 - x_4\} \geq d \).

**Case** \( 0 \leq x < x_1 \): Point \( x \) is only visited by robot \( r_1 \). We now calculate the time interval between two consecutive visitations of \( x \) by \( r_1 \). We distinguish two subcases.

**First** consider the subcase where \( r_1 \) is moving to the left when it visits \( x \). Before \( r_1 \) visits \( x \) again, it has to visit 0 and then return to \( x \). Therefore, the time between the two visitations of \( x \) is \( 2x \).

**Second** consider the subcase in which \( r_1 \) is moving to the right when it visits \( x \). Before \( r_1 \) visits \( x \) again, it has to visit \( x_1 \) (i.e. enter interval \([x_1, x_4]\)), zigzag between points \( x_1 \) and \( x_4 \) until its distance to the other robots becomes \( d \), and then \( r_1 \) exits the interval \([x_1, x_4]\) and return to \( x \). Below we compute the total time between these two visitations of \( x \) by \( r_1 \).
Using Lemmas 5 and 6, we see that the approximation ratio of Algorithm 2 is 2.

\[ \text{Algorithm 2} \]

We conclude that the total time between two consecutive visitations of \( x \) by \( r_1 \) is 2.

Taking into consideration both subcases, the overall (worst case) waiting time of \( x \) is 2.

\[ \text{Case } x_4 < x \leq 1: \text{ The analysis is analogous to the previous case.} \]

\[ \text{Case } x_1 \leq x \leq x_4: \text{ Point } x \text{ is visited by both } r_1 \text{ and } r_2. \text{ We consider two subcases} \]

1. The two consecutive visits of \( x \) are by the same robot \( r_1 \) or \( r_2 \); this case occurs when either of \( r_1 \) or \( r_2 \) zigzags inside the interval \([x_1, x_4]\). Therefore \( w(x) = 2 \max\{x_4 - x, x - x_1\} \).

2. The two consecutive visits of \( x \) are by different robots \( r_1 \) and \( r_2 \); this case occurs when one robot is exiting the interval \([x_1, x_4]\) and the other one is entering it.

First suppose that \( r_1 \) visits \( x \) and the next visit of \( x \) is performed by \( r_2 \). The worst waiting time in this case occurs when \( r_1 \) is about to visit \( x \) but the distance between \( r_1 \) and \( r_2 \) reduces to \( d \) and \( r_1 \) turns away from \( x \). Then \( r_2 \) visits \( x \) after at most \( d \) time steps. Note that since \( x_1 \geq d \) the visit of \( x \) by \( r_2 \) is guaranteed. Therefore \( w(x) \leq 2(x - x_1) + d \). Now assume that \( r_2 \) visits \( x \) and the next visit of \( x \) is performed by \( r_1 \). By a similar discussion we have that \( w(x) \leq 2(x_4 - x) + d \). This implies that

\[ w(x) \leq 2 \max\{x - x_1, x_4 - x\} + d. \]

By Subcases 1, 2 above we conclude that \( w(x) \leq 2 \max\{x - x_1, x_4 - x\} + d \), for all \( x \in [x_1, x_4] \).

\[ \square \]

The proof of Lemma 3 follows by upper bounding \( \max_{x \in \mathcal{S}} \frac{w(x)}{f(x)} \). Using Lemmas 4 and Lemma 6, we see that the approximation ratio of Algorithm 2 is no more than

\[ \frac{w(x)}{f(x)} \leq \begin{cases} 
\frac{2 \max\{x_1 - x - d\}}{\max\{2d, 2(1 - x - x_4 + x_1), x_4 - x_1\}}, & x \in [0, x_1) \\
\frac{2 \max\{x_4 - x_1 - d\}}{2 \max\{x_1 - x, x_4 - x\} + d}, & x \in [x_1, x_4] \\
\frac{2 \max\{x_4 - x_1 - d\}}{\max\{2(1 - x), 2(x - x_4 + x_1), x_4 - x_1\}}, & x \in (x_4, 1] 
\end{cases} \]

Using that \( d = \frac{\min\{x_1, x_4 - x_1\}}{1 + \alpha} \), and the fact that the given instance is \( \alpha \)-expanding, i.e. that \( x_1 = \alpha(x_4 - x_1) \), and after some tedious and purely algebraic calculations, we see that \( \frac{w(x)}{f(x)} \leq \frac{2 + \alpha}{1 + \alpha} \) for all \( x \in \mathcal{S} \), as wanted. Details can be found in Appendix C.
The paper investigated the problem of patrolling a line segment by two robots when time-patrolling constraints are placed on the frequency of visitation of all the points of the line. As shown in this study, finding “efficient” trajectories that satisfy the requirements or even deciding on their existence for two robots turns out to be a highly intricate problem. Nothing better is known aside from the $\sqrt{3}$-approximation algorithm for two robots on a line presented in this work. The patrolling problem with constraints is also open for more general graph topologies (e.g., cycles, trees, etc.). Further, the case of patrolling with constraints for multiple robots is completely unexplored in all topologies, including for the line segment.

References

1. S Alishamrani, D.R. Kowalski, L. Gąsieniec, How Reduce Max Algorithm Behaves with Symptoms Appearance on Virtual Machines in Clouds, In Proc. IEEE International Conference CIT/IUC/DASC/PICOM, 2015, pp 1703-1710.
2. S. K. Baruah, N. K. Cohen, C. G. Plaxton, and D. A. Varvel. Proportionate progress: A notion of fairness in resource allocation. *Algorithmica*, 15(6):600–625, June 1996.
3. S. K. Baruah and S.-S. Lin. Pfair scheduling of generalized pinwheel task systems. *IEEE Transactions on Computers*, 47(7):812–816, July 1998.
4. M. A. Bender, S. P. Fekete, A. Kröller, J.S.B. Mitchell, V. Liberatore, V. Polishchuk, J. Suomela. The Minimum Backlog Problem. *Theoretical Computer Science*, 605:51–61, November 2015.
5. M. H. L. Bodlaender, C. A. J. Hurkens, V. J. J. Kusters, F. Staals, G. J. Woeginger, and H. Zantema Cinderella versus the Wicked Stepmother. In Proc. TCS 2012, LNCS v. 6942, pages 57–71. Springer, 2012.
6. M. Y. Chan and F. Y. L. Chin. General schedulers for the pinwheel problem based on double-integer reduction. *IEEE Trans. on Comp.*, 41(6):755–768, June 1992.
7. M. Y. Chan and F. Chin. Schedulers for larger classes of pinwheel instances. *Algorithmica*, 9(5):425–462.
8. M. Chrobak, J. Csirik, C. Imreh, J. Noga, J. Sgall, G.J. Woeginger. The Buffer Minimization Problem for Multiprocessor Scheduling with Conflicts. In Proc. ICALP 2001, LNCS v. 2076, pages 862–874. Springer 2001.
9. A. Collins, J. Czyzowicz, L. Gąsieniec, A. Kosowski, E. Kranakis, D. Krizanc, R. Martin, and O. Morales Ponce. Optimal Patrolling of Fragmented Boundaries. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA ’13, pages 241–250, New York, USA, 2013.
10. J. Czyzowicz, L. Gąsieniec, A. Kosowski, and E. Kranakis. Boundary Patrolling by Mobile Agents with Distinct Maximal Speeds. In Proc. ESA 2011, number 6942 in Lecture Notes in Computer Science, pages 701–712. Springer, September 2011.
11. J. Czyzowicz, L. Gąsieniec, A. Kosowski, E. Kranakis, D. Krizanc, and N. Taleb. When Patrolmen Become Corrupted: Monitoring a Graph using Faulty Mobile Robots In Proc. ISAAC 2015, 26th International Symposium on Algorithms and Computation, pp. 343-354.
12. P. C. Fishburn and J.C, Lagarias, Pinwheel Scheduling: Achievable Densities. *Algorithmica*, 34(1):14–38.
13. L. Gąsieniec, R. Klasing, C. Levcopoulos, A. Lingas, J. Min, T. Radzik, Bamboo Garden Trimming Problem (Perpetual Maintenance of Machines with Different Attendance Urgency Factors), In Proc. SOFSEM 2017, pp. 229–240.
14. R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel. The pinwheel: a real-time scheduling problem. In II: Software Track, Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences, 1989. Vol. volume 2, pages 693–702 vol.2, January 1989.
15. R. Holte, L. Rosier, I. Tulchinsky, and D. Varvel. Pinwheel scheduling with two distinct numbers. *Theoretical Computer Science*, 100(1):105–135, June 1992.
16. A. Kawamura and Y. Kobayashi, Fence patrolling by mobile agents with distinct speeds, Distributed Computing 28(2): 147-154, 2015.
17. D. Liang and H. Shen, Point Sweep Coverage on Path, unpublished work available at [https://arxiv.org/abs/1704.04332](https://arxiv.org/abs/1704.04332).
18. S.-S. Lin and K.-J. Lin. A Pinwheel Scheduler for Three Distinct Numbers with a Tight Schedulability Bound. *Algorithmica*, 19(4):411–426.
19. S. Ntafos. On gallery watchmen in grids. *Information Processing Letters*, 23(2):99–102, 1986.
20. J. O’Rourke. Art gallery theorems and algorithms. Oxford University Press, Vol. 57, 1987.
21. T. H. Romer and L. E. Rosier. An algorithm reminiscent of Euclidean-gcd for computing a function related to pinwheel scheduling. *Algorithmica*, 17(1):1–10, ’97.
22. P. Serafini and W. Ukovich. A Mathematical Model for Periodic Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581, November 1989.
23. J. Urrutia. Art gallery and illumination problems. *Handbook of computational geometry*, 1(1):973–1027, 2000.
A Tightness Analysis of Algorithm 1

For every $\alpha$, we provide a feasible $\alpha$-expanding PUF instance for which the performance of Algorithm 1 is exactly equal to $1 + 2\alpha$. The instance is as follows. Choose $x_1 \in (0, 1)$ such that $2x_1 + \frac{x_1}{\alpha} = 1$. Consider three points $y_0 = 0$, $y_1 = x_1(1 + \frac{1}{2\alpha})$, and $y_2 = 1$. Let $I(y_0) = I(y_2) = 4x_1 + \frac{x_1}{\alpha}$ and $I(y_1) = \frac{x_1}{\alpha}$.

Let $\mathcal{A}$ be a solution as follows. At the beginning, the robot $r_1$ locates at $x_1$ and the robot $r_2$ locates at $y_1$, and both robots move to the right. See Figure 2. The patrolling segment of $r_1$ is $[y_0, y_1]$ and the patrolling segment of $r_2$ is $[y_1, y_2]$. The robot $r_1$ moves back and forth inside the interval $[y_0, y_1]$ and each time it visits $y_1$ stays at $y_1$ for $2x_1$. Similarly the robot $r_2$ moves back and forth inside the interval $[y_1, y_2]$ and each time it visits $y_1$ stays at $y_1$ for $2x_1$. Then $w_{\mathcal{A}}(y_0) = w_{\mathcal{A}}(y_2) = 4x_1 + \frac{x_1}{\alpha} = I(y_0) = I(y_1)$ and $w_{\mathcal{A}}(y_1) = \frac{x_1}{2\alpha} < I(y_1)$. Therefore $\mathcal{A}$ is a feasible solution.

Now consider Algorithm 1. For the above example we have $x_2 = x_3 = y_1$. Therefore $r_1$ zigzags inside $[0, y_1]$ and $r_2$ zigzags inside $[y_1, 1]$. Since the movement of the robots is not coordinated in Algorithm 1 the worst waiting time of $y_1$ in Algorithm 1 is $2x_1 + \frac{x_1}{\alpha}$. This implies that

$$\frac{w(x)}{I(x)} = \frac{x_1(2 + \frac{1}{\alpha})}{\frac{x_1}{\alpha}} = 1 + 2\alpha.$$

B Tightness Analysis of Algorithm 2

For every $\alpha$, we provide a feasible $\alpha$-expanding PUF instance for which the performance of Algorithm 2 is exactly equal to $\frac{1 + \alpha}{1 + \alpha}$. For this we consider two cases.

Case 1. $\alpha < 1$: Choose $x_1 \in (0, 1)$ such that $x_1 + 2\frac{x_1}{\alpha} = 1$. Consider four points $y_0 = 0$, $y_1 = x_1 + \frac{x_1}{\alpha}$, $y_2 = \frac{2x_1}{\alpha}$, and $y_3 = 1$. See Figure 3. Let $I(y_0) = I(y_3) = x_1(2 + \frac{1}{\alpha})$, $I(y_1) = \frac{x_1}{\alpha}$, and $I(y_2) = \frac{2x_1}{\alpha}$.

Let $\mathcal{A}$ be a solution as follows.
Fig. 3. The red arrow determines the patrolling area of \( r_1 \) in \( A \) and the blue arrow determines the patrolling area of \( r_2 \) in \( A \).

(1) At the beginning, the robot \( r_1 \) locates at \( x_1 \) and the robot \( r_2 \) locates at \( y_1 \), and both robots move to the right.

(2) The patrolling segment of \( r_1 \) is \( [y_0, y_1] \) and the patrolling segment of \( r_2 \) is \( [y_1, y_3] \).

(3) The robot \( r_1 \) moves back and forth inside the interval \( [y_0, y_1] \) and each time \( r_1 \) visits \( y_1 \) stays at \( y_1 \) until \( r_2 \) is at point \( y_1 + \frac{x_1}{\alpha} \) and is moving to the left. Then \( r_1 \) moves to the left, visits \( 0 \) and returns to \( y_1 \).

(4) The robot \( r_2 \) moves back and forth inside the interval \( [y_1, y_3] \) and each time it visits \( y_1 \) stays at \( y_1 \) for \( 2x_1 \). Then \( r_2 \) moves to the right, visits \( y_2 \) and \( 1 \) and returns to \( y_1 \).

First we analyze the movement of the robots in \( A \). The robot \( r_1 \) leaves \( y_1 \) when the distance between \( r_1 \) and \( r_2 \) is \( x_1 \alpha \). Note that this is possible since the length of \( [y_1, y_3] \) is greater than \( x_1 \alpha \). The next visit of \( r_1 \) from \( y_1 \) occurs after

\[
\frac{x_1}{2\alpha} + 2x_1 + \frac{x_1}{2\alpha} = \frac{x_1}{\alpha} + 2x_1,
\]

which is the time that \( r_1 \) spends to visit \( 0 \) and return to \( y_1 \). During the time \( \frac{x_1}{\alpha} + 2x_1 \) the robots \( r_1 \) traverse from \( y_1 + \frac{x_1}{\alpha} \) to \( y_1 \) and stays there for \( 2x_1 \). Therefore by the time \( r_1 \) arrives at \( y_1 \), the robot \( r_2 \) leaves \( y_1 \). We now compute the waiting time of \( A \).

The point \( y_0 \) is only visited by \( r_1 \). So the waiting time of \( 0 \) is equal “two times the length of the interval [0, 1] plus the time \( r_1 \) stops at \( y_1 \)”. The stop time of \( r_1 \) at \( y_1 \) is equal the time \( r_2 \) traverse from \( y_1 \) to \( 1 \) and return to the point \( y_1 + \frac{x_1}{\alpha} \) which is \( \frac{2x_1}{\alpha} \). Therefore

\[
w_A(y_0) = \frac{2x_1}{\alpha} + 2x_1 + \frac{x_1}{\alpha} = x_1(2 + \frac{3}{\alpha}) = I(x).
\]

The point \( y_1 \) is visited by both \( r_1, r_2 \). First suppose that \( r_1 \) is waiting at point \( y_1 \). The robot \( r_1 \) leaves \( y_1 \) when \( r_2 \) is at distance \( \frac{x_1}{\alpha} \) of \( y_1 \). So the next visit of \( y_1 \) occurs after \( \frac{x_1}{\alpha} \) time by \( r_2 \). Also, as we discussed above by the time \( r_2 \) is ready to leave \( y_1 \) the robot \( r_1 \) arrives at \( y_1 \). So in this case there is no time interval between the visits of \( r_2 \) and \( r_1 \). Therefore

\[
w_A(y_1) = \frac{x_1}{\alpha} = I(x).
\]
The point $y_2$ is only visited by $r_2$. If $r_2$ is moving to the left when it visits $y_2$ then the next visit of $y_2$ occurs in $\frac{2x_1}{\alpha}$. This is the time $r_2$ spends to visit $y_1$ and stays at it for $2x_1$ and returns to $y_2$. If $r_2$ is moving to the right when it visits $y_2$ then clearly the next visit of $y_2$ occurs in $2(1 - \frac{2x_1}{\alpha}) = 2x_1$. Therefore

$$w_A(y_2) = \max\{2x_1, \frac{2x_1}{\alpha}\} = \frac{2x_1}{\alpha} = I(x).$$

For $y_3 = 1$ it is easy to see that

$$w_A(y_3) = 2(1 - y_1) + 2x_1 = 2(1 - x_1 - \frac{x_1}{2\alpha}) + 2x_1 = 2 - \frac{x_1}{\alpha} = x_1(2 + \frac{3}{\alpha}).$$

The last equation follows from the fact that $x_1 + \frac{2x_1}{\alpha} = 1$. Therefore $A$ is a feasible solution.

Now consider Algorithm 2. For the above example we have $d = \frac{x_1}{1+\alpha}$, $x_2 = x_3 = y_1$, and $x_4 = x_1 + \frac{x_1}{\alpha} < \frac{x_1}{1+\alpha}$. Therefore by Lemma 6

$$w(y_2) = 2\max\{x_1, \frac{2x_1}{\alpha} - \frac{x_1}{1+\alpha}\} = \frac{2x_1}{\alpha} - \frac{x_1}{1+\alpha}.$$Therefore

$$\frac{w(y_2)}{I(y_2)} = \frac{\frac{2x_1}{\alpha} - \frac{x_1}{1+\alpha}}{\frac{2x_1}{\alpha}} = 1 + \frac{x_1(1+\alpha)}{\frac{x_1}{\alpha}} = \frac{2 + \alpha}{1 + \alpha}.$$

**Case 2.** $\alpha \geq 1$: Choose $x_1 \in (0, 1)$ such that $2x_1 + \frac{x_1}{\alpha} = 1$ and let $0 < \epsilon < x_1$. Consider four points $y_0 = 0$, $y_1 = x_1 - \epsilon$, $y_2 = x_1 + \frac{x_1}{\alpha}$, and $y_3 = 1$. See Figure 4. Let $I(y_0) = I(y_3) = 2(1 - x_1)$, $I(y_1) = 2(x_1 + \epsilon)$, and $I(y_2) = \frac{x_1}{\alpha}$.

**Fig. 4.** The red arrow determines the patrolling area of $r_1$ in $A$ and the blue arrow determines the patrolling area of $r_2$ in $A$.

Let $A$ be a solution as follows.

1. At the beginning, the robot $r_1$ locates at $x_1$ and the robot $r_2$ locates at $x_1 + \frac{x_1}{\alpha}$, and both robots move to the right.
2. The patrolling segment of $r_1$ is $[y_0, y_2]$ and the patrolling segment of $r_2$ is $[y_2, y_3]$. 
(3) The robot \(r_1\) moves back and forth inside the interval \([y_0, y_2]\) and each time it visits \(y_1\) stays at \(y_1\) for \(\frac{x_1}{x}\). Then \(r_1\) moves to the left, visits \(y_1\) and 0 and returns to \(y_1\).

(4) The robot \(r_2\) moves back and forth inside the interval \([y_2, y_3]\) and each time it visits \(y_1\) stays at \(y_1\) for \(\frac{x_1}{x}\). Then \(r_2\) moves to the right, visits 1 and returns to \(y_1\).

First we analyze the movement of the robots in \(A\). The robot \(r_1\) leaves \(y_1\) when the distance between \(r_1\) and \(r_2\) is \(\frac{x_1}{x}\), and similarly the robot \(r_2\) leaves \(y_1\) when the distance between \(r_1\) and \(r_2\) is \(\frac{x_1}{x}\). This is possible since the length of the intervals \([0, y_2]\) and \([y_2, 1]\) is \(x_1 + \frac{x_1}{x}\) and \(x_1 \geq \frac{x_1}{x}\). It is easy to see that the

\[
[w_A(y_0) = w_A(y_3) = 2(1 - x_1)].
\]

The point \(y_1 = x_1 - \epsilon\) is only visited by \(r_1\) and so

\[
w(y_1) = 2\max\{y_1, y_2 - y_1 + \frac{x_1}{2x}\} = 2\max\{x_1 - \epsilon, \frac{x_1}{x} + \epsilon\} \leq 2(x_1 + \epsilon).
\]

In \(A\) the point \(y_2 = x_1 + \frac{x_1}{2x}\) is patrolled by both \(r_1\) and \(r_2\) in such a way that one robot stays at \(y_2\) until the other robot is moving towards \(y_2\) and is at distance \(\frac{x_1}{x}\) from \(y_2\). This implies that \(w_A(y_2) \leq \frac{x_1}{x} = I(x)\).

From the above discussion we have that \(A\) is a feasible solution. Now consider Algorithm 2. For the above example we have \(d = \frac{x_1}{x(1+\alpha)}, x_2 = x_3 = y_2\). Since \(y_1 = x_1 - \epsilon < x_1\) by Lemma 6 we have that

\[
w(y_1) = 2\max\{x_1 - \epsilon, 1 - (x_1 - \epsilon) - \frac{x_1}{\alpha(1 + \alpha)}\}
\]

\[
= 2\max\{x_1 - \epsilon, x_1 + \epsilon + x_1(\frac{1}{1 + \alpha})\} = x_1 + \epsilon + x_1(\frac{1}{1 + \alpha}).
\]

In the above equation we use the facts that \(2x_1 + \frac{x_1}{x} = 1\) and \(\alpha \geq 1\). Therefore

\[
\frac{w(y_1)}{I(y_1)} = \frac{x_1 + \epsilon + x_1(\frac{1}{1 + \alpha})}{x_1 + \epsilon} = 1 + \frac{1}{1 + \alpha} \frac{x_1}{x_1 + \epsilon}.
\]

This implies that as \(\epsilon\) converges to 0 the feasibility ratio of Algorithm 2 converges to \(1 + \frac{1}{1 + \alpha} = \frac{2 + x}{1 + \alpha}\).

C Omitted Proofs of Section 6.3

Using that \(d = \min\{x_1, x_4 - x_1\}\), and the fact that the given instance is \(\alpha\)-expanding, i.e. that \(x_1 = \alpha(x_4 - x_1)\), we prove that \(w(x)/I(x)\), as it reads in [1], is upper bounded by \(\frac{2 + x}{1 + \alpha}\). Naturally, we distinguish cases for the location of \(x\) with respect to the three subintervals \([1, x_1]\), \([x_1, x_4]\), \((x_4, 1 - x_4]\). For the sake of exposition, we split the proof in 3 corresponding Lemmata.
Lemma 7. For all $0 \leq x < x_1$ we have $\frac{w(x)}{I(x)} \leq \frac{2 + \alpha}{1 + \alpha}$.

Proof. Using (1) we have that

$$\frac{w(x)}{I(x)} \leq \frac{\max\{x, 1 - x - d\}}{\max\{x, 1 - x - x_4 + x_1, x_4 - x_1\}}.$$ 

If $x > \frac{1 - d}{2}$ then $\max\{x, 1 - x - d\} = x$, and so $\frac{w(x)}{I(x)} \leq 1$. So assume that $x \leq \frac{1 - d}{2}$.

We consider two cases:

Case 1: $\alpha \leq 1$. Then $x_1 = \alpha(x_4 - x_1) \geq x_4 - x_1$, and so $d = \frac{1}{1 + \alpha}x_1$.

$$\frac{w(x)}{I(x)} \leq \frac{1 - x - \frac{x_1}{1 + \alpha}}{1 - x - \frac{x_1}{\alpha}} \leq \frac{1 - (1 - \frac{2x_1}{\alpha}) - \frac{x_1}{1 + \alpha}}{1 - x - \frac{x_1}{\alpha}} = \frac{2x_1}{\alpha} - \frac{x_1}{1 + \alpha} = 1 + \frac{x_1(1 - \frac{1}{1 + \alpha})}{\frac{x_1}{\alpha}} = 1 + \frac{1}{\frac{1}{1 + \alpha}} = 2 + \alpha.$$ 

Now let $\max\{1 - x - \frac{x_1}{\alpha}, x_1\} = 1 - x - \frac{x_1}{\alpha}$. Then $\frac{x_1}{\alpha} \leq 1 - x - \frac{x_1}{\alpha}$, which implies that $x \leq 1 - \frac{2x_1}{\alpha}$.

$$\frac{w(x)}{I(x)} \leq \frac{1 - x - \frac{x_1}{\alpha}}{1 - \frac{x_1}{\alpha}} = \frac{1 - x - \frac{x_1}{\alpha} + \frac{x_1}{\alpha} - \frac{x_1}{1 + \alpha}}{1 - x - \frac{x_1}{\alpha}} = 1 + \frac{\frac{x_1}{\alpha(1 + \alpha)}}{1 - \frac{x_1}{\alpha}} \leq 1 + \frac{\frac{x_1}{(1 - \frac{2x_1}{\alpha}) - \frac{x_1}{1 + \alpha}}}{1 + \frac{1}{\frac{1}{1 + \alpha}}} = 2 + \alpha.$$ 

Case 2: $\alpha \geq 1$. Then $x_1 = \alpha(x_4 - x_1) \geq x_4 - x_1$, and $d = \frac{1}{1 + \alpha}(x_4 - x_1) = \frac{1}{\alpha}(1 + \alpha)x_1$.

$$\frac{w(x)}{I(x)} \leq \frac{1 - x - \frac{x_1}{\alpha(1 + \alpha)}}{\max\{x, 1 - x - \frac{x_1}{\alpha}, \frac{x_1}{\alpha}\}} \leq \frac{1 - x - \frac{x_1}{\alpha(1 + \alpha)}}{\max\{1 - x - \frac{x_1}{\alpha}\}}.$$ 

First let $\max\{x, 1 - x - \frac{x_1}{\alpha}\} = x$. Then $x \geq 1 - x - \frac{x_1}{\alpha}$, which implies that $x \geq \frac{1}{2} - \frac{x_1}{2\alpha}$.

$$\frac{w(x)}{I(x)} \leq \frac{1 - x - \frac{x_1}{\alpha(1 + \alpha)}}{x} \leq \frac{1 - (\frac{1}{2} - \frac{x_1}{2\alpha}) - \frac{x_1}{\alpha(1 + \alpha)}}{\frac{1}{2} - \frac{x_1}{2\alpha}} = \frac{\frac{1}{2} - \frac{x_1}{2\alpha} + (\frac{x_1}{\alpha} - \frac{x_1}{\alpha(1 + \alpha)})}{\frac{1}{2} - \frac{x_1}{2\alpha}} = \frac{1}{\frac{1}{2} - \frac{x_1}{2\alpha}}.$$
By assumption we know that \( x_1 \leq 1 - x_4 \), and \( x_4 - x_1 = \frac{2 x_1}{\alpha} \). Moreover \( x_1 + (x_4 - x_1) + (1 - x_4) = 1 \). Therefore \( x_1 + \frac{2 x_1}{\alpha} + x_1 \leq 1 \), which implies that \( x_1 \leq \frac{\alpha}{2\alpha + 1} \).

Therefore
\[
1 + \frac{x_1(\frac{1}{1 + \alpha})}{2 - \frac{2 x_1}{\alpha}} \leq 1 + \frac{\frac{2 \alpha + 1}{\alpha} (\frac{1}{1 + \alpha})}{2 - \frac{2 (2\alpha + 1)}{\alpha}} = 1 + \frac{\frac{2 \alpha + 1}{\alpha} (\frac{1}{1 + \alpha})}{\frac{1}{\alpha} + \frac{2}{\alpha}} = 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.
\]

Now let \( \max\{x, 1 - x - \frac{2 x_1}{\alpha}\} = 1 - x - \frac{2 x_1}{\alpha} \). Then \( x \leq 1 - x - \frac{2 x_1}{\alpha} \), which implies that \( x \geq 1 - \frac{2 x_1}{\alpha} \). In the following calculations we use the fact that \( x \leq x_1 \).

\[
w(x) \leq \frac{1 - x - \frac{x_1}{\alpha(1 + \alpha)}}{1 - x - \frac{2 x_1}{\alpha}} = \frac{1 - x - \frac{x_1}{\alpha} - \frac{2 x_1}{\alpha} - \frac{x_1}{\alpha(1 + \alpha)}}{1 - x - \frac{2 x_1}{\alpha}} \leq 1 + \frac{x_1(\frac{1}{1 + \alpha})}{1 - x - \frac{2 x_1}{\alpha}} \leq 1 + \frac{\frac{2 \alpha + 1}{\alpha} (\frac{1}{1 + \alpha})}{1 - \frac{2 x_1}{\alpha}} = 1 + \frac{\frac{2 \alpha + 1}{\alpha} (\frac{1}{1 + \alpha})}{\frac{2 x_1}{\alpha}} = 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.
\]

This proves that for all \( x \in [0, x_1) \) the ratio \( \frac{w(x)}{I(x)} \) is at most \( \frac{2 + \alpha}{1 + \alpha} \).  \( \square \)

**Lemma 8.** For all \( x_1 \leq x \leq x_4 \) we have \( \frac{w(x)}{I(x)} \leq \frac{2 + \alpha}{1 + \alpha} \).

**Proof.** Using \( \text{[1]} \) we have that
\[
\frac{w(x)}{I(x)} \leq \frac{\max\{x - x_1, x_4 - x\} + \frac{d}{2}}{\max\{x - x_1, x_4 - x\}} = 1 + \frac{d}{\max\{x - x_1, x_4 - x\}}.
\]

If \( x - x_1 \leq x_4 - x \) then \( x \leq \frac{x_1 + x_4}{2} \) and \( \max\{x - x_1, x_4 - x\} = x_4 - x \). Moreover \( x_4 - x \geq x_4 - \frac{x_1 + x_4}{2} = x_4 - x_1 \). Therefore
\[
\frac{w(x)}{I(x)} \leq 1 + \frac{d}{x_4 - x_1}.
\]

Since \( d \leq \frac{1}{1 + \alpha} (x_4 - x_1) \) we have
\[
\frac{w(x)}{I(x)} \leq 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.
\]

Now let \( x_4 - x \leq x - x_1 \). Then \( x \geq \frac{x_1 + x_4}{2} \) and \( \max\{x - x_1, x_4 - x\} = x - x_1 \). Moreover, \( x - x_1 \geq \frac{x_1 + x_4}{2} - x_1 = x_4 - x_1 \), and thus
\[
\frac{w(x)}{I(x)} \leq 1 + \frac{d}{x_4 - x_1} \leq 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.
\]

\( \square \)
Lemma 9. For all $x_4 < x \leq 1$ we have $\frac{w(x)}{I(x)} \leq \frac{2 + \alpha}{1 + \alpha}$.

Proof. Using (1) we have that

$$\frac{w(x)}{I(x)} \leq \frac{\max\{1 - x, x - d\}}{\max\{1 - x, x - x_3 + x_1, x_4 - x_1\}}.$$

If $x > \frac{1 - d}{2}$ then $\max\{1 - x, x - d\} = 1 - x$, and so $\frac{w(x)}{I(x)} \leq 1$. So assume that $x \leq \frac{1 - d}{2}$. We consider two cases:

**Case 1:** $\alpha \leq 1$. Then $x_1 = \alpha(x_4 - x_1) \leq x_4 - x_1$, and so $d = \frac{1}{1 + \alpha} x_1$.

First let $\max\{x - \frac{x_1}{\alpha}, \frac{x_1}{\alpha}\} = \frac{x_1}{\alpha}$. Then $\frac{x_1}{\alpha} \leq x - \frac{x_1}{\alpha}$, which implies that $x \leq \frac{2x_1}{\alpha}$.

Therefore

$$\frac{w(x)}{I(x)} \leq \frac{x - \frac{x_1}{\alpha}}{\frac{x_1}{\alpha}} \leq \frac{2x_1}{\alpha} - \frac{x_1}{1 + \alpha} = \frac{2x_1}{\alpha} - \frac{x_1}{1 + \alpha} = 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.$$

Now let $\max\{x - \frac{x_1}{\alpha}, \frac{x_1}{\alpha}\} = x - \frac{x_1}{\alpha}$. Then $\frac{x_1}{\alpha} \leq x - \frac{x_1}{\alpha}$, which implies that $x \geq \frac{2x_1}{\alpha}$.

$$\frac{w(x)}{I(x)} \leq \frac{x - \frac{x_1}{\alpha}}{x - \frac{x_1}{\alpha}} = \frac{x - \frac{x_1}{\alpha} + \frac{x_1}{\alpha} - \frac{x_1}{1 + \alpha}}{x - \frac{x_1}{\alpha}} = 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.$$

**Case 2:** $\alpha \geq 1$. Then $x_1 = \alpha(x_4 - x_1) \geq x_4 - x_1$, and $d = \frac{1}{1 + \alpha}(x_4 - x_1) = \frac{1}{\alpha(1+\alpha)} x_1$.

$$\frac{w(x)}{I(x)} \leq \frac{x - \frac{x_1}{\alpha}}{\max\{1 - x, x - \frac{x_1}{\alpha}, \frac{x_1}{\alpha}\}} \leq \frac{x - \frac{x_1}{\alpha}}{\max\{1 - x, x - \frac{x_1}{\alpha}\}}.$$

First let $\max\{1 - x, x - \frac{x_1}{\alpha}\} = 1 - x$. Then $1 - x \geq x - \frac{x_1}{\alpha}$, which implies that $x \leq \frac{1}{2} + \frac{x_1}{2\alpha}$.

$$\frac{w(x)}{I(x)} \leq \frac{x - \frac{x_1}{\alpha}}{1 - x} \leq \frac{\frac{1}{2} + \frac{x_1}{2\alpha} - \frac{x_1}{\alpha(1+\alpha)}}{1 - x} = 1 + \frac{x_1}{1 + \alpha}.$$
By assumption we know that \( x_1 \leq 1 - x_4 \), and \( x_4 - x_1 + (1 - x_4) = 1 \). Therefore \( x_1 + \frac{x_1}{\alpha} + x_1 \leq 1 \), which implies that \( x_1 \leq \frac{\alpha}{2\alpha + 1} \). Therefore

\[
1 + x_1 \left( \frac{1 + \alpha}{\alpha} \right) \leq 1 + \frac{\alpha}{2\alpha + 1} \left( \frac{1 + \alpha}{1} \right) = 1 + \frac{\alpha}{2\alpha + 1} \left( \frac{1 + \alpha}{\alpha} \right)
\]

\[
= 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.
\]

Now let \( \max\{1 - x, x - x_1 \alpha\} = x - \frac{x_1}{\alpha} \). In the calculations we use the fact that \( x \geq x_4 \) and \( x_4 = \frac{1 + \alpha}{\alpha} x_1 \) (recall that \( x_1 = \alpha(x_4 - x_1) \)).

\[
\frac{w(x)}{I(x)} \leq \frac{x - \frac{x_1}{\alpha(1 + \alpha)}}{x - \frac{x_1}{\alpha}} = \frac{x - \frac{x_1}{\alpha} + \frac{x_1}{\alpha} - \frac{x_1}{\alpha(1 + \alpha)}}{x - \frac{x_1}{\alpha}}
\]

\[
= 1 + x_1 \left( \frac{1 - \frac{1}{\alpha(1 + \alpha)}}{x - \frac{x_1}{\alpha}} \right) \leq 1 + x_1 \left( \frac{1 + \alpha}{x_4 - \frac{x_1}{\alpha}} \right)
\]

\[
= 1 + \frac{x_1 \left( \frac{1 + \alpha}{\alpha} \right)}{\frac{x_1}{\alpha} - \frac{x_1}{\alpha}} = 1 + \frac{1}{1 + \alpha} = \frac{2 + \alpha}{1 + \alpha}.
\]

This proves that for all \( x \in (x_4, 1] \) the ratio \( \frac{w(x)}{I(x)} \) is at most \( \frac{2 + \alpha}{1 + \alpha} \). \( \square \)