Fence Patrolling with Two-speed Robots

Jurek Czyzowicz[1], Konstantinos Georgiou[2], Evangelos Kranakis[3], Fraser MacQuarrie[3] and Dominik Pajak[4]

[1]*Departemant d'Informatique, Universite du Quebec en Outaouais, Gatineau, Québec, Canada*

[2]*Department of Mathematics, Ryerson University, Toronto, Ontario, Canada*

[3]*School of Computer Science, Carleton University, Ottawa, Ontario, Canada*

[4]*Department of Computer Science FFPT, Wroclaw University of Technology, Wroclaw, Poland*

Keywords: Idleness, Mobile Robots, Patrolling, Speed, Walking, Scheduling.

Abstract: A fence is to be patrolled collectively by n robots. At any moment a robot may move in one of the two possible states: *walking* or *patrolling*. Each state is associated with a maximal moving speed which cannot be exceeded. We want to schedule the perpetual movements of the robots so as to minimize the *idleness*, defined as the smallest time interval within which every point is always visited by some robot. First, we give a centralized algorithm constructing schedules with optimal idleness, and subsequently we show an interesting application to a transportation problem concerning *Scheduling with Regular Delivery*. Our main contribution is the study of distributed, dynamical schedules for patrolling robots with only primitive capabilities. Surprisingly, we are able to design a dynamic schedule for very weak collections of two robots (silent, oblivious, passively mobile), achieving the optimal idleness. Part of our contribution is a very technical analysis of the dynamics of special families of dynamical systems of n robots that we call *regular*. For such systems we also propose a highly non-trivial $O(n^2)$ algorithm to decide whether or not robots converge to a stable configuration thus verifying if the dynamic schedule is optimal.

1 INTRODUCTION

Fence patrolling is the act of perpetual monitoring of a domain, modelled as a unit segment, by a set of mobile robots. As the robots cannot continuously monitor all points of the domain, a standard efficiency measure of patrolling is the notion of *idleness* - the size of the minimal time interval for which all points of the domain are always visited (independently of the start of this interval). Patrolling algorithms attempt to produce the schedules, (i.e. the trajectories of the robots in time) minimizing idleness.

In previous research, the robots were supposed to have either the same or different maximal speeds of their patrolling movements. In this paper we assume that the patrolling activity requires more elaborate work than just walking within the domain, e.g., when *foraging* or *harvesting* which take longer than walking, and in computing applications, *web page indexing, forensic search, code inspection, packet sniffing* which require a more involved inspection. Consequently, we assume the maximal patrolling speeds to be strictly smaller than the maximal walking speeds.

We suppose that each robot is capable of patrolling in only one direction (arbitrarily chosen) of the segment while walking is permitted in both directions. It is worth noting that the fence problem for robots patrolling in both directions appears to be quite difficult. Indeed, optimal idleness algorithms were found only for fence patrolling of up to three robots, (cf.(Czyzowicz et al., 2011; Dumitrescu et al., 2014; Kawamura and Kobayashi, 2012)). Such one-way patrolling assumption is natural for humans, e.g. in text inspection (cf. the difficulty of the backward spelling game). Several mechanical devices operate actively in one direction only, e.g. combine and forage harvesters, snow-plowers. Also traveling up the hill, swimming against river flow, carrying heavy loads - all these activities require slower operating speeds.

We solve optimally the patrolling problem in the centralized scenario. We also show how to apply our algorithm to natural optimization problem in transportation concerning *Scheduling with Regular Delivery*. Next, and more surprisingly, using a very weak

Czyzowicz, J., Georgiou, K., Kranakis, E., MacQuarrie, F. and Pajak, D.
Fence Patrolling with Two-speed Robots.
DOI: 10.5220/0005687102290241
In *Proceedings of 5th the International Conference on Operations Research and Enterprise Systems (ICORES 2016)*, pages 229-241
ISBN: 978-989-758-171-7

collection of mobile robots with only primitive capabilities, we are able to design a distributed strategy converging to the same optimal solution thus achieving the same idleness as the optimal centralized algorithm. Our robots are anonymous, oblivious, silent (no communication permitted) and they cannot process any information, as well as they are not aware of their numbers or their speeds. They are subject to *passive mobility*, i.e. their movement is dynamically controlled by their contacts with the environment. The robots are only capable of walking or patrolling with maximal speed and their perception mechanism permits only to recognize a collision with the environment (i.e. the endpoints of the segment or another robot). This is the first study of the patrolling problem in a decentralized setting thus leading to the design of a distributed, self-stabilizing algorithm (cf (Dijkstra, 1982)[EWD386, pp 34-35], which discusses a solution to a cyclic relaxation problem).

1.1 Related Work

Patrolling is the act of surveillance, consisting of walking perpetually around an area in order to protect or supervise it, and has been studied intensively in robotics (Almeida et al., 2004; Chevaleyre, 2004; Elmaliach et al., 2009; Elmaliach et al., 2008; Hazon and Kaminka, 2008; Machado et al., 2002; Yanovski et al., 2003) where it is often viewed as a version of terrain *coverage*, a central task in robotics. It is useful in ecological monitoring, detecting intrusion, monitoring and locating objects or humans (that may need to be rescued from a disaster), detecting network failures or even discovering web pages which need to be indexed by search engines (Machado et al., 2002). Boundary and area patrolling have been studied in (Agmon et al., 2008; Elmaliach et al., 2009; Elmaliach et al., 2008; Pasqualetti et al., 2010; Hare et al., 2015) with approaches placing more emphasis on experimental results.

Idleness is the accepted measure of algorithmic efficiency of patrolling and is related to the frequency with which the points of the environment are visited (Almeida et al., 2004; Alpern et al., 2011; Chevaleyre, 2004; Elmaliach et al., 2009; Elmaliach et al., 2008; Machado et al., 2002)) (this last criterion was first introduced in (Machado et al., 2002)). Diverse approaches to patrolling based on idleness criteria are discussed in (Almeida et al., 2004). Patrolling as a game between patrollers and intruder is studied in (Alpern et al., 2009; Alpern et al., 2011; Amigoni et al., 2010). Elsewhere patrolling is studied based on swarm or ant-based algorithms (Elor and Bruckstein, 2010; Marino et al., 2009; Yanovski et al., 2003). Robots are memoryless (or having small memory), decentralized (Marino et al., 2009) with no explicit communication permitted either with other robots or the central station, and may have local sensing capabilities (Elor and Bruckstein, 2010). Ant-like algorithms usually mark the visited nodes of the graph and (Yanovski et al., 2003) presents an evolutionary process. This paper shows that a team of memoryless robots, by leaving marks at the nodes while walking through them, after relatively short time stabilizes to the patrolling scheme in which the frequency of the traversed edges is uniform to a factor of two (i.e., the number of traversals of the most often visited edge is at most twice the number of traversal of the least visited one).

Theoretical graph-based approaches to patrolling can be found in (Chevaleyre, 2004). In (Pasqualetti et al., 2010), polynomial-time patrolling solutions for lines and trees are proposed. For the case of cyclic graphs, (Pasqualetti et al., 2010) proves the NP-hardness of the problem and a constant-factor approximation is proposed.

Optimal patrolling with same-speed robots in mixed domains, where regions to be traversed are fragmented by components that do not need to be monitored, is studied in (Collins et al., 2013). Patrolling with robots that do not necessarily have identical speeds offers several surprises both in terms of the difficulty of the problem as well as in terms of the algorithmic results obtained. Such a study has been initiated in (Czyzowicz et al., 2011) and investigated further in (Dumitrescu et al., 2014; Kawamura and Kobayashi, 2012). The partition strategy, where each robot patrols and walks along a separate area, has been proven to work for two robots in (Czyzowicz et al., 2011), and for three in (Kawamura and Kobayashi, 2012).

Standard capabilities of mobile robots usually include communication, computation, and environment perception. For many reasons (e.g. production cost, limited or specific applications) one may wish to deal with robots of reduced ability, especially if they are needed in large numbers. In such cases, feasibility issues, rather than computation efficiency are sought (Angluin et al., 2006; Angluin et al., 2007; Beauquier et al., 2010; Cieliebak et al., 2012). (Angluin et al., 2006) introduced *population protocols* (see also (Angluin et al., 2007; Beauquier et al., 2010)), where robots are subject to *passive mobility*, also used in our paper. Passive mobility aims to model volatile environments like water flow, wind or unstable mobility of agents' carriers. Further, (Beauquier et al., 2010) considered different speed of such agents. (cf. also (Czyzowicz et al., 2013)).

1.2 Our Model, Problem Definition & Notation

Consider a set R of n mobile robots $r_1,\ldots,r_n$ each associated with some *patrolling speed* p_i and some *walking speed* w_i, where $p_i < w_j$ for $i,j = 1,\ldots,n$. Robots perpetually move along the unit segment $[0,1]$ in both directions. At any moment, a robot r_i may be in a *walking state* in which it moves at its walking speed w_i or in the *patrolling state* and moving at its patrolling speed p_i. Each robot may walk in both directions but its patrolling is always done in the same direction of the interval $[0,1]$. Hence we can say that each robot is associated with its patrolling direction (positive or negative direction of the interval) which does not change throughout its entire movement.

The Fence Patrolling Problem: Let S denote a *patrolling schedule*, i.e. an algorithm associated with each of the robots dictating on what speed and in what direction every robot moves at each moment of time. Given a schedule produced by algorithm A, the idle time $I_A(P,t)$ of a point P at time t is defined as the amount of time needed to the next visit of P after time t by any robot of R: $I_A(P,t) = \min\{t_v > t : \exists i(0 \le i \le n \text{ such that } r_i(t_v) = P)\} - t$, where we denote by $r_j(t)$ the point of the segment visited by r_j at time t, for $t \in [0,\infty)$. We are interested in an algorithm minimizing the maximal idle time taken over all points P and all time moments t. However, as it may be impossible to design an algorithm which achieves the best possible idle time right from the start of the schedule (e.g. robots may be in an unsuitable position to achieve this) we will be allowing any finite delay for checking idle times of segment points. More exactly, the *idleness* of algorithm A is defined as $I_A = \inf_{T\ge 0}\sup_{0\le p\le 1} I_A(P,T)$. An optimal patrolling algorithm is a schedule which optimizes the idleness among all possible patrolling algorithms under consideration.

It will be useful to observe that our goal may be viewed as the following equivalent maximization task. Suppose that the robots operate in an infinite line. What is the maximal length segment $[0,L]$ and the perpetual movement of the robots of R, such that in each unit time interval $[t,t+1]$ each point of $[0,L]$ is visited at least once by a robot in patrolling state. We call such a length L the *patrolling range* of R.

The Distributed Model: In this model, robots have only very primitive capabilities allowing them to execute separate schedules which change according to their perception of the environment. They are *oblivious* and *silent* (cannot communicate) and they cannot process any information. Besides two-speed mobility they can perceive the environment by recognizing obstacles (i.e., endpoints) or other robots that they do not even need to recognize. They are not aware of their patrolling or walking speeds nor of the length of the segment. In the schedules produced by our distributed algorithms the robots function according to the *bouncing-rule*: if a robot collides with another robot or a segment endpoint, then it changes direction as well as moving-state. A patrolling schedule of n robots is in a *stable configuration* $(x_1^*,\ldots,x_n^*)$ if robot i moves within the interval $[x_{i-1}^*, x_i^*]$ (we set $x_0^* = 0$ and $x_n^* = 1$), bouncing always at its endpoints.

Scheduling with Regular Delivery: Our techniques allow us to solve the following transportation problem. Suppose that at point 1 of the unit segment there is an infinite quantity of a commodity that is transported to point 0 by robots R. Each robot may carry one item of the commodity using a speed not exceeding p_i or it may travel with no load with a speed not exceeding w_i. At any time a robot r_i may drop the item it is carrying at the point currently occupied by r_i or it may pick up an item present at its current position. In particular, when two robots meet at a point an item being carried by one of them may be transferred to the other one. At all times no robot may carry more than one commodity item. What is the smallest value I, such that during any time interval $(kI,(k+1)I]$ a new item is delivered to point 0.

1.3 Our Contributions & Organization of the Paper

We attempt to solve the Fence Patrolling Problem using centralized or distributed algorithms. As a warm-up, we give in Section 2 an optimal centralized algorithm for the problem, whose solution serves as a benchmark for subsequent sections. We conclude in Section 2.2 with an interesting application to an optimization transportation problem. Our main technical contributions appear in Section 3 where we study the Fence Patrolling Problem in a distributed setting and where robots have only primitive capabilities. First, in Section 3.1 we optimally solve Fence Patrolling with 2 primitive robots. In the same section, we also develop the main ideas we built upon for patrolling with an arbitrary number of robots. Then, in Section 3.2 we introduce a generic distributed solution for patrolling with primitive robots. Our solution induces a complex dynamical system, whose analysis is the main focus of the remaining of the sections. Section 3.3 proposes a technical and highly non-trivial analysis of the dynamics of primitive robots, and con-

cludes with an efficient and analytic algorithm for deciding whether the system of robots converges to an optimal solution. Finally, Section 3.4 studies restricted, yet natural families of primitive robot collections that have the potential of inducing dynamic systems that converge to stable and optimal solutions. With non-trivial and technical arguments, we conclude by showing that special families of three and four primitive robots do solve the Fence Patrolling problem optimally.

2 OPTIMALLY SOLVING FENCE PATROLLING BY CENTRALIZED ALGORITHMS

In this section we give the general, centralized algorithm generating optimal patrolling schedules for the Fence Patrolling Problem and for any number of robots.

2.1 Schedule Obtained by a Centralized Algorithm

Consider the schedule defined by Algorithm 1. The

Algorithm 1: Centralized Schedule.

Input: Set of robots R with associated walking and patrolling speeds
Output: Schedule of R
1: Let $\sigma_j = \sum_{i=1}^{j} \frac{1}{1/p_i+1/w_i}$ for $j = 1,\ldots,n$
2: **for** $j = 1,\ldots,n$ **do**
3: Place robot r_j at initial position $x_j = \sigma_j/\sigma_n$
4: **repeat forever**
5: In patrolling state, move (left) at speed p_j until reaching point x_{j-1}
6: In walking state, move (right) at speed w_j until reaching point x_j

idea of the algorithm is to divide the unit interval into subsegments, and to make each robot operate only in the subsegment associated with it. Robot r_i is associated with a subsegment of size proportional to $\frac{p_i w_i}{p_i+w_i}$. Each robot zigzags between the endpoints of its subsegment patrolling in one direction and walking in the opposite direction.

Theorem 2.1. *Algorithm 1 produces an optimal patrolling schedule.*

Before we provide the proof of Theorem 2.1 we show Lemmata 2.2 and 2.3.

Lemma 2.2. *The idleness I_1 of Algorithm 1 satisfies $I_1 = 1/\sum_{i=1}^{n} \frac{1}{1/p_i+1/w_i}$.*

Proof. As σ_j, $j = 0,\ldots,n$, is an increasing sequence, each robot r_j operates within the subsegment $[x_{j-1},x_j]$, for $j = 1,\ldots,n$, having interior disjoint with all other subsegments. Consider any index j and any point $x \in [x_{j-1},x_j]$. Denote by t^* a time when r_j is in the patrolling state and $r_j(t^*) = x$. Before r_j visits x in patrolling state the next time, it has to traverse from x to x_{j-1} patrolling followed by walking the subsegment $[x_{j-1},x_j]$ and patrolling from x_j to x. The time needed for this equals

$$\frac{x-x_{j-1}}{p_j} + \frac{x_j-x_{j-1}}{w_j} + \frac{x_j-x}{p_j}$$
$$=(x_j-x_{j-1})\left(\frac{1}{p_j}+\frac{1}{w_j}\right) = \frac{1}{\sum_{i=1}^{n}\frac{1}{1/p_i+1/w_i}}$$

■

Lemma 2.3. *Let $D > 1$ be the distance patrolled by robot r_i during some time interval. Then in the same time interval r_i must walk distance at least $D-1$.*

Proof. Suppose, by symmetry, that r_i can patrol only in the right-to-left (i.e. negative) direction of the interval. The difference between the initial and the final position of the robot equals to $R-L$, where L denotes the sum of the lengths of its left-to-right moves and R denotes the sum of the lengths of its right-to-left moves. Observe that $R-L \leq 1$. As the total patrolling distance is at most R and the total walking distance is at least L we have the claim of the lemma. ■

We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1. Consider any algorithm $\mathcal{A}$ and its idleness $I_{\mathcal{A}}$. It is sufficient to show, that for any $\varepsilon > 0$ there exists a time interval $T = [t^*, t^* + I_1 - \varepsilon]$ during which some point of the segment is not patrolled by any robot. Let κ be an integer such that $\kappa > \frac{\sum_{i=1}^{n}\frac{1/w_i}{1/p_i+1/w_i}}{\varepsilon\sum_{i=1}^{n}\frac{1}{1/p_i+1/w_i}}$. Consider any time interval K of size $= \kappa I_{\mathcal{A}}$. We prove that K must contain interval T with the property from the claim made above.

Let d_i denote the distance traversed by r_i while patrolling during interval K. As each point of the segment must be patrolled at least κ times during time interval K we have $\sum_{i=1}^{n} d_i \geq \kappa$. By Lemma 2.3 we have $\frac{d_i}{p_i} + \frac{d_i-1}{w_i} \leq \kappa I_{\mathcal{A}}$, hence $d_i \leq \frac{\kappa I_{\mathcal{A}}+1/w_i}{1/p_i+1/w_i}$. Therefore $\kappa \leq \sum_{i=1}^{n} d_i \leq \kappa I_{\mathcal{A}}\sum_{i=1}^{n}\frac{1}{1/p_i+1/w_i} + \sum_{i=1}^{n}\frac{1/w_i}{1/p_i+1/w_i}$ and

$$I_{\mathcal{A}} \geq \frac{1}{\sum_{i=1}^{n}\frac{1}{1/p_i+1/w_i}} - \frac{\sum_{i=1}^{n}\frac{1/w_i}{1/p_i+1/w_i}}{\kappa\sum_{i=1}^{n}\frac{1}{1/p_i+1/w_i}}$$
$$\geq \frac{1}{\sum_{i=1}^{n}\frac{1}{1/p_i+1/w_i}} - \varepsilon$$

This proves Theorem 2.1. ■

Due to Lemma 2.2, it follows that almost all points of the segment have the same idle time (with the exception of the endpoints of subsegments which may be alternately visited by two robots). From Algorithm 1, it follows that each robot r_i operates solely inside a subsegment of size $\frac{1}{1/p_i+1/w_i} / \sum_{i=1}^{n} \frac{1}{1/p_i+1/w_i}$, for $i = 1,\ldots,n$. By scaling down the consideration to the idle time of 1 we easily obtain:

Corollary 2.4. *The patrolling range of robot r_i having patrolling speed p_i and walking speed w_i equals $\frac{1}{1/p_i+1/w_i}$. The patrolling range of a set of robots equals the sum of their patrolling ranges.*

2.2 Application to Transportation: Scheduling with Regular Delivery

In this section we show an interesting application of our previous findings in *Scheduling with Regular Delivery* (see Section 1.2 for the definition). The following Proposition shows that the idleness of Algorithm 1 is the optimal value for the problem concerning Scheduling with Regular Delivery.

Proposition 2.5. *The solution (i.e., the optimal time interval) of the problem concerning Scheduling with Regular Delivery for the set of robots R equals I_1 - the idleness of Algorithm 1.*

Proof. Each robot of Algorithm 1 revisits the left endpoint of its subinterval (and also its right endpoint) regularly at time intervals of I_1. It is then possible to synchronize robot movements so that each robot gets to its subsegment left endpoint (or right endpoint) exactly at the same when arrives there the left (resp. right) neighbouring robot. Suppose that robot r_n picks up an item at each visit to point 1 of the segment, and that, each time a robot reaches the left endpoint of its subinterval, it transfers the carried item to its left neighbour. Then each robot r_i either walks left-to-right unload with speed w_i or transports one item traveling right-to-left. Consequently, r_1 delivers one new item to point 0 at regular time intervals I_1. The optimality follows by the argument similar to that used in the proof of Theorem 2.1. ■

3 FENCE PATROLLING WITH PRIMITIVE ROBOTS / THE DISTRIBUTED CASE

In this section we consider the case when the collection of patrolling robots acts in a distributed way (see Section 1.2 for the model). We are interested in an algorithm having the same idleness as the one of the optimal centralized algorithm even if our robots are very weak. For notational reference we will assume that there exist two motionless robots r_0 and r_{n+1} (i.e. $w_0 = p_0 = w_{n+1} = p_{n+1} = 0$), positioned at the left and right endpoint, respectively. This way every robot r_i, for $i = 1,\ldots,n$, bounces at its left or right neighbor. We have the following lemma.

Lemma 3.1. *For any collection R of robots there exists a centralized algorithm producing an optimal schedule in stable configuration, in which the robots behave according to the bouncing rule.*

Proof. Consider the partition of the unit interval as in line 1 of Algorithm 1. By Corollary 2.4 each robot executing Algorithm 1, within the same time interval, independently covers a segment subinterval proportional to its patrolling range. It is then possible to reschedule the robots' starting times so that each robot arrives at the left endpoint of its subinterval exactly at the same time as when its left neighbor arrives at the right endpoint of its interval, resulting in a meeting. ■

We will attempt to design a distributed algorithm producing robots trajectories converging to a schedule of a stable configuration of robots. The task seems of special interest, given that robots are assumed to be oblivious and silent.

3.1 Distributed Optimal Schedule for Two Robots

The purpose of this section is to demonstrate that two primitive robots can optimally solve the Fence Patrolling Problem.

Let I_{opt} be the optimal idleness of the offline schedule for two robots. We design an algorithm, for which for any $\varepsilon > 0$ there exists a time t^*, such that in every time interval $[t, t + I_{opt} + \varepsilon]$, with $t \geq t^*$, each point of the segment is visited by some robot. Obviously, using such weak robots, it is impossible to design an algorithm which achieves optimal idle time only after some finite time of their operation. This would need robots capable of recognizing the parameters of the environment (e.g. patrolling and walking speeds of robots, distance traveled, time between collisions, etc.). Since robots react only when colliding with each other, or when they reach one endpoint (as if they do not know when collisions will occur or where the endpoints are) we slightly abuse standard terminology and we call our algorithm online.

We show that Algorithm 2 is the optimal one, i.e. its idleness equals the idleness of the optimal offline

Algorithm 1. The first critical observation is that collision points converge to a stable configuration.

Algorithm 2: Online Schedule for Two Robots.

Input: Two robots r_1, r_2 placed at the two segment endpoints

Output: Schedule of R

1: Both robots start in patrolling state moving towards each other.
2: Each robot switches state and direction when colliding either with the other robot or with an endpoint.

Lemma 3.2. *The sequence of collision points of Algorithm 2 converges to the point* $\frac{\frac{1}{p_2}+\frac{1}{w_2}}{\frac{1}{p_1}+\frac{1}{w_1}+\frac{1}{p_2}+\frac{1}{w_2}}$.

Proof. Suppose that the two robots following the schedule produced by Algorithm 2 meet at a point x, $0 < x < 1$. Suppose also that before the meeting both robots were in the patrolling state and moving towards each other. We show first that the next meeting occurs at a point x' such that

$$x' = -\frac{\frac{1}{w_1}+\frac{1}{w_2}}{\frac{1}{p_1}+\frac{1}{p_2}}x + \frac{\frac{1}{p_2}+\frac{1}{w_2}}{\frac{1}{p_1}+\frac{1}{p_2}} \quad (1)$$

Indeed, as $p_1 < w_2$ and $p_2 < w_1$ no robot can be caught from behind by another one, while walking along the segment. Consequently, both robots reach the segment endpoints and they restart patrolling while moving towards each other, eventually colliding at x'. As both robots spend the same

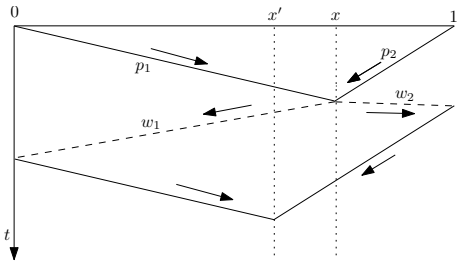

Figure 1: Two robots r_1, r_2 start by patrolling in opposite directions. They first collide at a point x and they both change to walking. After bouncing (not necessarily at the same time) at the endpoints of the interval $[0,1]$ they change to patrolling and meet anew at point x'. The vertical line indicates time.

time while traveling from x to x' (cf. Figure 1) we have $\frac{x}{w_1}+\frac{x'}{p_1} = \frac{1-x}{w_2}+\frac{1-x'}{p_2}$. Therefore $x'\left(\frac{1}{p_1}+\frac{1}{p_2}\right) = -x\left(\frac{1}{w_1}+\frac{1}{w_2}\right)+\frac{1}{p_2}+\frac{1}{w_2}$. We see from Identity (1) that $x' = -Ax+B$ where

$$A := \frac{\frac{1}{w_1}+\frac{1}{w_2}}{\frac{1}{p_1}+\frac{1}{p_2}}, B := \frac{\frac{1}{p_2}+\frac{1}{w_2}}{\frac{1}{p_1}+\frac{1}{p_2}}. \quad (2)$$

If x_1 is the initial collision point then we see that the kth collision point satisfies the recurrence $x_k = -Ax_{k-1}+B$. This yields a geometric series from which it follows that $x_k = (-A)^k x_0 + B\frac{1-(-A)^k}{1+A}$. Since $A < 1$ as $k \to \infty$ we get $x_k \to \frac{B}{1+A} = \frac{\frac{1}{p_2}+\frac{1}{w_2}}{\frac{1}{p_1}+\frac{1}{w_1}+\frac{1}{p_2}+\frac{1}{w_2}}$ which completes the proof. ■

Our intention is to generalize the lemma above for dynamic systems with arbitrarily many robots. Till then, we show that Algorithm 2 produces the optimal schedule.

Theorem 3.3. *The idleness of the schedule of Algorithm 2 equals I_1 - the idleness of the optimal schedule produced by the centralized algorithm.*

Proof. It is sufficient to show that for any $\varepsilon > 0$ there exists a point in time t^* such that for any $t > t^*$ and any time interval $T = [t, t+I_1+\varepsilon]$ each point of the segment is patrolled by some robot. Observe that from the start to the first meeting point x_0 robot r_1 patrols the interval $[0, x_0]$, while during the same time interval r_2 patrols the interval $[x_0, 1]$. Hence we have $\frac{x_0}{p_1} = \frac{1-x_0}{p_2}$. Solving for x_0 we get $x_0 = \frac{1}{p_2}/\left(\frac{1}{p_1}+\frac{1}{p_2}\right)$. Observe that the distance $D = |x_k - x_{k-1}|$ between the $(k-1)$-th and k-th meeting points (defined in the proof of Lemma 3.2)

$$\begin{aligned} D &= \left| \begin{array}{l} \left((-A)^k x_0 + B\frac{1-(-A)^k}{1+A}\right) \\ -\left((-A)^{k-1}x_0 + B\frac{1-(-A)^{k-1}}{1+A}\right) \end{array} \right| \\ &= \left|((-A)^k - (-A)^{k-1})(x_0 - \frac{B}{1+A})\right| \\ &= |(-A)^k(B - x_0(A+1))| \\ &= A^k(B - x_0(A+1)) \\ &\qquad \text{(since } A > 0 \ \&\ (B - x_0(A+1)) > 0) \end{aligned}$$

is converging to 0, since $|A| < 1$. As x_k and x_{k-1} are on the different sides of the convergence point $\frac{B}{1+A}$, within every time interval $I_1 + \frac{D}{\min(p_1,p_2)}$ the entire segment is jointly patrolled by both robots. Let K be such that $K \geq \log_A \frac{\varepsilon \cdot \min(p_1,p_2)}{B - x_0(A+1)}$ and let $t^* > (K+1)(\frac{1}{p_1}+\frac{1}{w_1}+\frac{1}{p_2}+\frac{1}{w_2})$. As $(\frac{1}{p_1}+\frac{1}{w_1}+\frac{1}{p_2}+\frac{1}{w_2})$ is the time between two consecutive bounces between robots r_1, r_2, after time t^* the robots bounced at least k times. Hence (as $0 < A < 1$) for $t > t^*$ the idle time $I(p,t)$ of every point p is $I(p,t) \leq I_1 + \frac{D}{\min(p_1,p_2)} =$

$I_1 + \frac{A^k(B-x_0(A+1))}{\min(p_1,p_2)} \leq I_1 + \varepsilon$, which proves the theorem. ■

Observe that Algorithm 2 works even if the robots do not necessarily start their respective schedules at the same time. Indeed, because $p_i < w_j$, $i,j = 1,2$, when the second robot wakes up and starts patrolling it cannot meet the robot which started first while this robot is in the walking state. Therefore the robots meet when they are both in the patrolling state and the subsequent meetings converge to the same point as before.

Note that, when robots start walking rather than patrolling, their subsequent meeting points do not converge.[1] On the other hand if one robot starts in the walking state while the other one in the patrolling state the convergence is possible in at most one of the two symmetric cases, depending which among the two values $1/p_1 + 1/w_2$ or $1/w_1 + 1/p_2$ is larger. In particular, when $1/p_1 + 1/w_2 = 1/w_1 + 1/p_2$ (e.g. in the case of identical robots) the meeting points alternate between two symmetric positions on the segment and the idleness is clearly suboptimal (cf. Fig. 2).

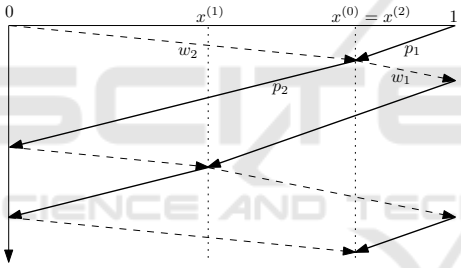

Figure 2: Two robots r_1, r_2 perpetually alternate between $x(0)$ and $x(1)$ (r_1 starts by patrolling and r_2 by walking). Here $w_1 = 5, p_1 = 3\frac{1}{7}, w_2 = 10, p_2 = 4$.

3.2 Distributed Schedule for n Primitive Robots

In this section we propose a distributed solution for an arbitrary number of primitive robots that induces an interesting dynamic system, which we analyze in subsequence sections.

As observed for the case of two robots, the convergence was attested if the two robots were colliding always in the patrolling state. For more than two robots this condition can no longer be guaranteed as all but two robots collide with both neighbors. We begin our analysis by proposing Algorithm 3, an intuitive distributed schedule that assumes the bouncing rule, i.e. that robots can respond to bounces by flipping their moving state (e.g. from walking to patrolling) and their moving direction. Moreover, as in our setting robots will start walking simultaneously at the same segment endpoint we naturally assume that $p_i \neq p_j$ (and that $w_i \neq w_j$), otherwise identical robots would always stay together and only one of them would contribute to the patrolling algorithm. As all robots pa-

Algorithm 3: Distributed Schedule for Many Robots.

Input: A collection R of robots with distinct patrolling and walking speeds
Output: Schedule of R
1: All robots start from the rightmost endpoint of the interval, in patrolling state moving right-to-left.
2: Each robot switches state and direction while colliding with the other robot or with an endpoint.

trol in the same direction and they change states only when meeting we can conclude with the following:

Observation 3.4. *Algorithm 3 produces a dynamic schedule with Regular Delivery, for a given set of robots.*

Notice that our algorithm defines a complex dynamical system of memoryless robots moving back and forth in an interval. The analysis of the system dynamics is very complicated, given that robot collisions might occur either between robots moving in opposite directions, or in the same directions (i.e. a collision may happen from behind). In what follows we analyze the dynamics under the assumption that we have one type of collisions, which as we shall see in the next subsections naturally arise by restricting the configuration of the robot speeds to what we later call monotone speeds.

We call the dynamical system that arises from Algorithm 3 *regular* if collisions occur only between robots that move in opposing directions, and therefore collisions happen only between a robot that is in patrolling state moving right-to-left and a robot in walking state moving left-to-right. We are interested in answering whether regular dynamical systems converge in a stable configuration, and whether this configuration has optimal idle time. Below we propose a highly efficient algorithm for verifying whether a regular dynamical system has this property (for any number of robots). Then we answer this question in the positive for up to 4 robots, under the assumption that speeds satisfy a natural condition.

[1] This follows from the proof of Theorem 3.3, as in this case, the critical value of quantity $A = \frac{1/p_1+1/p_2}{1/w_1+1/w_2} > 1$ (the roles of patrolling and walking speeds are swapped in the equation for x').

3.3 Dynamics for Regular Systems of Primitive Robots

Dynamic systems of primitive robots induced by Distributed Algorithm 3 are highly complex and difficult to analyze. The purpose of this section is to provide a deep and technical analysis of the dynamics of regular systems. We conclude the section by proposing a highly non-trivial algorithm for deciding convergence of primitive robots in a stable configuration (where robots eventually move in disjoint subintervals).

Notice that in every dynamical system, and at every point in the time horizon, robots will appear on the interval in the same order. We rename the robots so that robot $i+1$ is always to the right of robot i, after they develop according to Algorithm 3. Below we denote by x_t^i the point in the interval where robots r_i, r_{i+1} bounce for the t-th time. The purpose of the next lemma is to predict points x_t^i. The reader may view it as the analogue of (part of the proof of) Lemma 3.2 that dealt with only two robots.

Lemma 3.5. *In a regular system we have*

$$\left(\frac{1}{p_{i+1}}+\frac{1}{w_i}\right)x_{t+1}^i-\left(\frac{1}{p_i}+\frac{1}{w_i}\right)x_{t+1}^{i-1}$$
$$=\left(\frac{1}{p_{i+1}}+\frac{1}{w_{i+1}}\right)x_t^{i+1}-\left(\frac{1}{p_i}+\frac{1}{w_{i+1}}\right)x_t^i$$

Proof. First we claim that $x_t^{i-1}<x_t^i,\ i=1,\ldots,n$, and that robots r_{i-1},r_i bounce at x_t^{i-1} before r_i,r_{i+1} bounce at x_t^i. Indeed, let $\tau(x_t^i)$ denote the time of this bounce. Note that robot r_1 (while patrolling) bounces first at the origin and on its way back (now walking) bounces with robot r_2 (which is patrolling). After robot r_2 bounces with robot r_1, it begins walking and eventually bounces with robot r_3 which moves in opposite direction and is patrolling, etc. This reasoning shows that $x_1^{i-1}<x_1^i,\ i=1,\ldots,n$ as well as that r_{i-1},r_i bounce for the first time before r_i,r_{i+1} do. Since the system is regular, each robot alternates its bounces between both neighbors r_{i-1}, r_{i+1} which is sufficient to conclude the claim.

Next observe that between time $\tau(x_t^i)$ and $\tau(x_{t+1}^i)$ robot r_{i-1} first patrols right-to-left the interval $[x_{t+1}^{i-1},x_t^i]$ then it walks left-to-right the interval $[x_{t+1}^{i-1},x_{t+1}^i]$. During the same time interval r_i first walks left-to-right the interval $[x_t^i,x_t^{i+1}]$ then it patrols right-to-left the interval $[x_{t+1}^i,x_t^{i+1}]$ (see Figure 3). Comparing both times we get $\frac{x_t^i-x_{t+1}^{i-1}}{p_i}+\frac{x_{t+1}^i-x_{t+1}^{i-1}}{w_i}=\frac{x_t^{i+1}-x_t^i}{w_{i+1}}+\frac{x_t^{i+1}-x_{t+1}^i}{p_{i+1}}$. Regrouping terms implies the lemma. ■

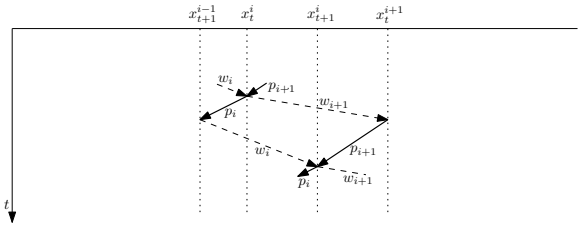

Figure 3: Two time consecutive bounces between robots r_i, r_{i+1}.

We can rewrite the recurrence in Lemma 3.5 in a more concise matrix form. Define the following matrices.

$$A=\operatorname{diag}\left(\frac{1}{p_{i+1}}+\frac{1}{w_i}\right)_{i=1,\ldots,n-1}$$
$$-\text{L-diag}\left(\frac{1}{p_i}+\frac{1}{w_i}\right)_{i=2,\ldots,n-1} \quad (3)$$
$$B=\operatorname{diag}\left(\frac{1}{p_i}+\frac{1}{w_{i+1}}\right)_{i=1,\ldots,n-1} \quad (4)$$
$$-\text{U-diag}\left(\frac{1}{p_i}+\frac{1}{w_i}\right)_{i=2,\ldots,n-1} \quad (5)$$
$$c^T=\left(0,\ldots,\frac{1}{p_n}+\frac{1}{w_n}\right),$$

where by L-diag and U-diag we mean the low and upper diagonal matrices of dimension $(n-1)\times(n-1)$ and entries as indicated placed below and above the main diagonal, respectively.

Theorem 3.6. *Consider a regular dynamical system of n robots (produced by Algorithm 3) and let A,B,c be the matrices defined in Equations* (4)*. If the moduli (norms) of all eigenvalues of the matrix $A^{-1}B$ are less than 1, then the schedule of Algorithm 3 converges to a schedule in stable configuration which is also optimal (w.r.t. to centralized algorithms). In particular, for every $\varepsilon>0$, after $\Theta(\log 1/\varepsilon)$ bounces (iterations) of any pair of neighboring robots the idle time $I_3(p,t)$ is such that $I_3(p,t)\le(1+\varepsilon)\frac{1}{\sum_{i=1}^n \frac{1}{1/p_i+1/w_i}}$.*

Proof. Let $X_t\in\mathbb{R}^{n-1}$ be the vector $\left(x_t^1,\ldots,x_t^{n-1}\right)^T$. Then the recurrence of Lemma 3.5 can be rewritten in matrix form as $AX_{t+1}+BX_t=c$. From this, we derive that

$$X_t=\begin{array}{l}(-1)^t\left(A^{-1}B\right)^t\\+\left(I+A^{-1}B\right)^{-1}\left(I-(-1)^{-1}\left(A^{-1}B\right)^t\right)A^{-1}c.\end{array}$$

Next consider the eigenvalue decomposition $A^{-1}B=Q\Lambda Q^T$, where Q is an orthogonal matrix. Then $\left(A^{-1}B\right)^t=Q\Lambda^tQ^T$, and since $\lim_{t\to\infty}\Lambda^t=\mathbf{0}$ as all eigenvalues have norm less than 1, we conclude

that $\lim_{t\to\infty} X_t$ exists, i.e. the sequence converges to $X^* = \left(I + A^{-1}B\right)^{-1} A^{-1}c = (A+B)^{-1}c$ and the convergence is linear. From the definition of the recurrence, it follows that the schedule of Algorithm 3 converges to the schedule S which is in stable configuration X^*.

By Corollary 2.4 which, in view of Lemma 3.1, applies also to stable configurations the patrolling range of the collection of robots equals $\sum_{i=1}^{n} \frac{1}{1/p_i+1/w_i}$ and by the rate of convergence, after $\Theta(\log 1/\varepsilon)$ bounces of neighboring robots, the idle time is already no more than than $(1+\varepsilon)\frac{1}{\sum_{i=1}^{n} \frac{1}{1/p_i+1/w_i}}$. ∎

Note that already Theorem 3.6 suggests a numerical method for checking whether the dynamical system arising from Algorithm 3 converges or not; given patrolling and walking speeds p_i, w_i, first compute matrix A^{-1}, and then calculate all eigenvalues of $A^{-1}B$ and verify that their norm is less than 1. Matrix inversing can be done explicitly and efficiently, say by Gauss-Jordan elimination or by LU decomposition. Finding however the eigenvalues of the non-Hermitian matrix $A^{-1}B$ is at least as difficult as finding the roots of high degree polynomials. In light of Abel's impossibility theorem, one has to rely on numerical methods to verify that the moduli of the eigenvalues of $A^{-1}B$ are indeed less than 1. In fact, a number of sophisticated numerical methods have been proposed to efficiently find eigenvalues of special families of matrices.

We depart from this approach, and in contrast to numerical methods, we propose an explicit, symbolic and efficient algorithm for verifying the precondition of Theorem 3.6 without explicitly computing the eigenvalues of $A^{-1}B$. Our strategy is to first give an explicit expression of A^{-1}, which in turn will allow us to calculate the characteristic polynomial of $A^{-1}B$. Finally, we invoke a powerful theorem that characterizes the range of polynomial roots (without finding them), and that can be exploited algorithmically.

We begin by calculating the characteristic polynomial of $A^{-1}B$. Note that for a group R of n robots, the characteristic polynomial of $A^{-1}B$ is of degree $n-1$. Also, any $r \times r$ leading principal minor $A^{-1}B$ can be computed from the $r \times r$ leading principal minors of A, B, which only depend on robots $1, \ldots, r+1$. In fact the $r \times r$ leading principal minor $A^{-1}B$ is exactly the critical matrix whose eigenvalues determine the convergence of Algorithm 3 for input robots $1, \ldots, r+1$. So, we are motivated in denoting by $D_r(\lambda)$ the characteristic polynomial of the $r \times r$ leading principal minor $A^{-1}B$ (i.e. $D_{n-1}(\lambda) = |A^{-1}B - \lambda I|$). We choose to abbreviate $D_r(\lambda)$ by D_r. The next lemma provides two alternative recursive relations for D_r (each will be convenient in different arguments) that allow us to compute D_{n-1}, and will be also used later to establish convergence for special cases of robots.

Lemma 3.7. *Consider matrices A, B as defined in* (4)*, and introduce the abbreviations $a_i := \frac{1}{p_{i+1}} + \frac{1}{w_i}, b_i := \frac{1}{w_{i+1}} + \frac{1}{p_i}, c_i := -\frac{1}{p_{i+1}} - \frac{1}{w_{i+1}}$, and $\alpha_i := \frac{b_i}{a_i} - \frac{c_{i-1}^2}{a_{i-1}a_i}, \beta_i := \frac{c_{i-1}^2}{a_{i-1}a_i}$. Then for the characteristic polynomials D_r, the following equivalent recurrences hold true for all $r \geq 2$*

$$D_r = (\alpha_r - \lambda) \cdot D_{r-1} + \sum_{t=1}^{r-2} \left(\alpha_{t+1} \cdot \prod_{j=t+2}^{r} \beta_j \right) \cdot D_t + \frac{b_1}{a_1} \prod_{j=2}^{r} \beta_j, \quad (6)$$

$$D_r = \left(\frac{b_r}{a_r} - \lambda \right) D_{r-1} + \frac{c_{r-1}^2}{a_{r-1}a_r} \lambda D_{r-2} \quad (7)$$

and with initial conditions $D_1 = \frac{b_1}{a_1} - \lambda$, $D_0 = 1$.

Proof. The inverse of A is a lower triangular matrix and can be easily verified to be defined as

$$A_{ij}^{-1} := (-1)^{i+j} \frac{\prod_{t=j}^{i-1} c_t}{\prod_{r=j}^{i} a_r}, \quad \text{if } j \leq i$$

and is 0 otherwise. Therefore, for the (so-called Hessenberg matrix, as it has zero entries above the first subdiagonal) matrix $A^{-1}B$ we have that

$$\left(A^{-1}B\right)_{ij} = (-1)^{i+j} \left(\frac{b_j}{a_i} \prod_{t=j}^{i-1} \frac{c_t}{a_t} - \frac{c_{j-1}}{a_i} \prod_{t=j}^{i} \frac{c_{t-1}}{a_{t-1}} \right)$$

if $i \geq j-1$, and 0 otherwise, with the understanding that $c_0 = 0$. The following interesting relation holds for the entries of $A^{-1}B$, that is useful in finding the characteristic polynomial of the matrix.

$$\left(A^{-1}B\right)_{i,j} = -\frac{c_{i-1}}{a_i} \left(A^{-1}B\right)_{i-1,j}, \quad \forall i > j. \quad (8)$$

Next we introduce D'_r to denote a small variation of D_r. D'_r is the determinant of the same principal minor of $A^{-1}B - \lambda I$ (up to entry (r,r)) with the only difference that the entry (r,r) is replaced by $\left(A^{-1}B\right)_{r,r}$, instead of $\left(A^{-1}B - \lambda I\right)_{r,r}$.

With this notation, we can evaluate $\left|A^{-1}B - \lambda I\right|$ by expanding the determinant with respect to the entries $(n-1, n-1)$ and $(n-2, n-1)$. Using Equation (8), we observe that

$$D_r = \left(\frac{b_r}{a_r} - \frac{c_{r-1}^2}{a_{r-1}a_r} - \lambda \right) D_{r-1} + \frac{c_{r-1}^2}{a_{r-1}a_r} D'_{r-1}, \quad (9)$$

$$D'_r = \left(\frac{b_r}{a_r} - \frac{c_{r-1}^2}{a_{r-1}a_r} \right) D_{r-1} + \frac{c_{r-1}^2}{a_{r-1}a_r} D'_{r-1}, \quad (10)$$

where the recurrence ends at $D_1 = \frac{b_1}{a_1} - \lambda$ and $D'_1 = \frac{b_1}{a_1}$. Repeated substitution of (10) to (9) and some direct calculations imply recurrence (6). Recurrence (7) is obtained from (6) by subtracting two consecutive terms of the sequence D_r. ■

Notice that Lemma 3.7, and in particular Equation (7), allows us to calculate the characteristic polynomial D_{n-1} of $A^{-1}B$ by performing no more than $\Theta(n^2)$ arithmetic operations (additions, multiplications and divisions) between speeds p_i, w_i. Next we give an efficient algorithm for deciding whether the moduli of the roots of an arbitrary polynomial $f : \mathbb{R} \mapsto \mathbb{R}$ are all less than 1. Our intention is to run Algorithm 4 with input D_{n-1}, i.e. the characteristic polynomial of $A^{-1}B$.

Algorithm 4: Decide Convergence.

Input: A polynomial $f : \mathbb{R} \mapsto \mathbb{R}$ of degree t of the form $\sum_{i=0}^{t} \gamma_i \lambda^i$

1: Set $\gamma_i^{(0)} = \gamma_i$, for $i = 0, \ldots, t$.

2: For $j = 0, \ldots, t-1$ and for $k = 0, \ldots, j+1$ compute $\gamma_k^{(j+1)} = \gamma_0^{(j)} \gamma_k^{(j)} - \gamma_{n-j}^{(j)} \gamma_{n-j-k}^{(j)}$.

3: Compute $\delta_{j+1} := \gamma_0^{(j+1)} = \left(\gamma_0^{(j)}\right)^2 - \left(\gamma_{n-j}^{(j)}\right)^2$ for $j = 0, \ldots, t-1$.

Output: YES if and only if $\delta_1 < 0$ and $\delta_j > 0$ for $j = 2, \ldots, t$.

Theorem 3.8. *A set of n robots R for which the output of Algorithm 3 gives a regular dynamical system converges to a stable configuration if and only if Algorithm 4 outputs YES on input D_{n-1}. As a result, convergence can be decided in $\Theta(n^2)$ arithmetic operations.*

Proof. By Theorem 3.6, the regular dynamical system converges to a stable configuration if and only if all eigenvalues of $A^{-1}B$ (as defined in (4)) have moduli less than 1. The characteristic polynomial of $A^{-1}B$ can be computed in $\Theta(n^2)$ many operations, as a corollary of Lemma 3.7. Clearly, Algorithm 3.8 requires no more than $\Theta(n^2)$ arithmetic operations. Therefore,we can decide convergence in $\Theta(n^2)$ arithmetic operations as long as we can show that Algorithm 4 correctly decides whether the input polynomial f has all its roots (real or complex) strictly inside the unit circle.

Correctness of Algorithm 4 is an immediate corollary of Theorem 42,1, p. 150 in (Marden, 1949): "Set $\Delta_r = \prod_{j=1}^{t} \delta_j, \ \ , r = 1, \ldots, t$ and suppose that k many of the products Δ_r are negative, and the remaining $t-k$ of them are positive. Then f has exactly k roots strictly inside the unit circle, exactly $t-k$ roots strictly outside the unit circle (and hence no roots on the unit circle)." ■

3.4 Monotone Robot Collections, and Convergence

In this section we demonstrate some special families of primitive robots that can solve Fence Patrolling optimally. The analysis even of the restricted families of three or four robots remains surprisingly technical and non-trivial.

Our technical results of Section 3.3 on regular dynamical systems raise the question whether such systems exist. A natural family of robots is when either the sum or the product of patrolling and walking speeds is constant for all robots or when some constant "power" of a robot may be used for improving its patrolling ability at the expense of its walking ability. In such a collection of robots, all patrolling speeds are dominated by the walking speeds, and the non-increasing order of patrolling speeds is the inverse order of that of the walking speeds. We make the definition formal.

Definition 3.9. *The collection R of n robots is called monotone if for $i, j = 1, \ldots, n$: 1) $p_i < w_i$, 2) $p_i \neq p_j$, and 3) $p_i < p_j \implies w_i > w_j$.*

A natural example of a monotone collection of robots is one where each robot i independently decides how to distribute it's energy e, which is the same for all robots, to walking and patrolling speeds w_i, p_i respectively, such that $w_i + p_i = e$. As it is observed before, a collection of robots that develop according to Algorithm 3 preserve the order they appear on the line. Without loss of generality we may assume that their indices are consecutive along the segment, i.e. that $w_n > w_{n-1} > \cdots > w_1 > p_1 > p_2 > \cdots > p_n$.

Lemma 3.10. *For a monotone collection of robots R, the dynamical system that arises from Algorithm 3 is regular (i.e. collisions occur while robots approach each other, the left one being in the walking state and the right one in the patrolling state.)*

Proof. Initially all robots walk right-to-left until the fastest walking robot collides with the left endpoint and starts walking left-to-right. Any "head on" collision results in the right robot switching to patrolling left-to-right and the left robot switching to patrolling right-to-left. So it is sufficient to prove that collisions from behind never take place. Suppose to the contrary, that there exists such a collision between a pair of consecutive robots on the segment, r_i, r_{i+1}, for $i = 1, \ldots, n-1$. Obviously r_i, r_{i+1} cannot collide when r_i moves left and r_{i+1} moves right. If both

robots move right-to-left then, by assumption, they must be walking, and since $w_i < w_{i+1}$, r_i cannot catch r_{i+1}. Similarly, if both robots move left-to-right, by assumption, they are patrolling and since $p_i > p_{i+1}$, r_{i+1} cannot catch r_i. ■

As an immediate observation, we also obtain that

$$\frac{b_i}{a_i} < 1 \quad \& \quad \frac{(c_{i-1})^2}{a_{i-1}a_i} < 1 \tag{11}$$

for all $i = 1, \ldots, n-1$ and $i = 2, \ldots, n-1$ respectively, and for all regular collections of n robots, where a_i, b_i, c_i are as in Lemma 3.7. In fact, the characteristic polynomial D_{n-1} has leading coefficient $(-1)^n$, while it is also immediate from (7) that the constant coefficient is $\prod_{i=1}^{n-1} \frac{b_i}{a_i} < 1$. This automatically shows that the condition $\delta_1 < 0$, of Algorithm 4, holds true for monotone collections of robots. In fact, we conjecture that monotone collections of robots always converge to a stable configuration, i.e. that $\delta_j > 0$ for $j = 2, \ldots, n-1$, but a general proof is eluding us. Still the proof of convergence for up to $n \leq 3$ robots is easy to establish. The proof of the next proposition relies on (11).

Proposition 3.11. *For a monotone collection of $n \leq 3$ robots the schedule produced by Algorithm 3 has the optimal idleness.*

Proof. We show that the conditions of Theorem 3.6 are satisfied. The case of 1 robot is straightforward. For $n = 2$ robots, the characteristic polynomial is $D_1 = \frac{b_1}{a_1} - x$, which by (11) has one real root with absolute value less than 1.

Now we turn our attention to $n = 3$, and by (7) the characteristic polynomial $D_2(\lambda)$ has the form $P_W(\lambda) := (U - \lambda)(V - \lambda) + W\lambda = \lambda^2 - (U + V - W)\lambda + UV$, where U, V, W are non negative constants which by (11) are strictly less than 1.

We show that the moduli of the roots of $P_W(\lambda)$ are less than 1. First we observe that $P_0(\lambda)$ has this property.

Case 1: If $P_W(\lambda)$ has real roots, then these are $\rho_{1,2}^{(W)} = \frac{U+V-W \pm \sqrt{(U+V-W)^2 - 4UV}}{2}$, with the understanding that $\rho_1^{(W)}, \rho_2^{(W)}$ correspond to the square root having positive and negative sign respectively. Then we observe that $\rho_1^{(W)} < \rho_1^{(0)} < 1$, while also $\rho_1^{(W)} > -W/2 > -1/2$ (by ignoring the positive terms). Hence $-1/2 < \rho_1^{(W)} < 1$. Similarly, we see that $\rho_2^{(W)} < (U+V)/2 < 1$ (by ignoring the negative terms). And finally note that $U + V - \sqrt{(U+V-W)^2 - 4UV} > -W$, since $(U+V+W)^2 > (U+V-W)^2 - 4UV$. Therefore, $\rho_2^{(W)} > (-2W)/2 = -1$, concluding that $-1 < \rho_2^{(W)} < 1$, as well.

Case 2: If $P_W(\lambda)$ has complex roots, say σ_1, σ_2, then it must be the case that $\|\sigma_1\|^2 = \|\sigma_2\|^2 = \sigma_1\sigma_2 = UV < 1$. ■

We now prove convergence of monotone collections of $n = 4$ robots by using a refinement of monotonicity.

Definition 3.12. *The collection R of n robots is called strongly monotone if it is monotone and for all r we have* $\left(\frac{1}{p_r} + \frac{1}{w_{r+1}}\right)\left(\frac{1}{p_r} + \frac{1}{w_{r-1}}\right) > \left(\frac{1}{p_r} + \frac{1}{w_r}\right)^2$.

Due to the definitions of a_i, b_i and that of α_i in Lemma 3.7, asking that a collection of robots is strongly monotone is equivalent to asking that $\alpha_r > 0$ for every r (see (6)). This allows us to show that characteristic polynomials associated with such robots have no negative real roots. We can now prove that the characteristic polynomial of every strongly monotone collection of robots has real roots less than 1 in absolute value.

Theorem 3.13. *For every monotone (not necessarily strongly) collection of n robots, $D_r(\lambda)$ preserves sign for all $\lambda \geq 1$. If in addition robots are strongly monotone, then $D_r(\lambda)$ preserves sign (and is actually positive) for all $\lambda < 0$. As a result all real roots of D_r lie strictly between 0 and 1.*

Proof. The less technical proof concerns the strongly monotone collections of robots. For this consider the cone $\mathcal{C}$ of polynomials of the form $\sum_{t=0}^{r}(-1)^t \rho_t \lambda^t$, where $\rho_t > 0$, i.e. polynomials whose odd-degree monomial coefficients are negative, and whose even-degree monomial coefficient are positive. Clearly, any polynomial $p(\lambda) \in \mathcal{C}$ is positive for every $\lambda < 0$ (and is actually decreasing).

We claim that for all $r \geq 0$, $D_r \in \mathcal{C}$. To that end, we first observe that the statement is true for $r = 0, 1$. For any $r \geq 2$, we invoke (6). Since all α_i, β_j are positive reals, we can show that $D_r \in \mathcal{C}$ as long as we can verify that $(\alpha_r - \lambda) \cdot D_{r-1} \in \mathcal{C}$ (the rest of summands in (6) are conical combinations of polynomials in $\mathcal{C}$). It is straightforward now to check that $-\lambda D_{r-1} \in \mathcal{C}$, hence $(\alpha_r - \lambda) \cdot D_{r-1} = \alpha_r D_{r-1} + (-\lambda D_{r-1}) \in \mathcal{C}$, as wanted.

Now we focus on a monotone (not necessarily strongly) collection of robots. We prove by induction on r that for all $\lambda \geq 1$, D_r is a polynomial which is

$$\begin{cases} \text{positive and increasing,} & \text{if } r \text{ is even} \\ \text{negative and decreasing,} & \text{if } r \text{ is odd} \end{cases}$$

Indeed, the statement is true for $r = 1, 2$. Next we turn our attention to $r \geq 3$. We have in mind to invoke (7). Now fix any $\lambda_0 \geq 1$. Note that $\frac{b_r}{a_r} - \lambda_0 < 0$. Next observe that if r is even, then $D_{r-1}(\lambda_0) < 0$ and $D_{r-2}(\lambda_0) > 0$, so that $D_r(\lambda_0) > 0$. Similarly, if r is

odd, then $D_{r-1}(\lambda_0) > 0$ and $D_{r-2}(\lambda_0) < 0$, so that $D_r(\lambda_0) < 0$, exactly as wanted.

Next we show the promised monotonicity. Let's denote by $D'_r(\lambda)$ the first derivative of $D_r(\lambda)$ with respect to λ. Then we see that

$$D'_r(\lambda) = \begin{array}{l} -\lambda D_{r-1}(\lambda) + \left(\frac{b_r}{a_r} - \lambda\right) D'_{r-1}(\lambda) \\ + \frac{c_{r-1}^2}{a_{r-1}a_r} D_{r-2}(\lambda) + \frac{c_{r-1}^2}{a_{r-1}a_r} \lambda D'_{r-2}(\lambda) \end{array}$$

If r is even then we argue that $D'_r(\lambda) > 0$ for all $\lambda \geq 1$. Indeed, we have that

$$\begin{aligned} -\lambda D_{r-1}(\lambda) &= - \cdot + \cdot - = + \\ \left(\frac{b_r}{a_r} - \lambda\right) D'_{r-1}(\lambda) &= - \cdot - = + \\ \frac{c_{r-1}^2}{a_{r-1}a_r} D_{r-2}(\lambda) &= + \cdot + = + \\ \frac{c_{r-1}^2}{a_{r-1}a_r} \lambda D'_{r-2}(\lambda) &= + \cdot + \cdot + = + \end{aligned}$$

Since all summands of $D'_r(\lambda)$ are positive, $D_r(\lambda)$ is increasing.

Similarly, if r is odd, we show that $D'_r(\lambda) < 0$ for all $\lambda \geq 1$. Indeed, we have that

$$\begin{aligned} -\lambda D_{r-1}(\lambda) &= - \cdot + \cdot + = - \\ \left(\frac{b_r}{a_r} - \lambda\right) D'_{r-1}(\lambda) &= - \cdot + = - \\ \frac{c_{r-1}^2}{a_{r-1}a_r} D_{r-2}(\lambda) &= + \cdot - = - \\ \frac{c_{r-1}^2}{a_{r-1}a_r} \lambda D'_{r-2}(\lambda) &= + \cdot + \cdot - = - \end{aligned}$$

Since all summands of $D'_r(\lambda)$ are negative, $D_r(\lambda)$ is decreasing. ■

In order to show that a group of strongly monotone robots converges to a stable configuration, it remains to prove all complex roots of D_r have norm < 1; this is the main idea behind the proof of Proposition 3.14.

Proposition 3.14. *For a strongly monotone collection of $n = 4$ robots the schedule produced by Algorithm 3 has the optimal idleness.*

Proof of Proposition 3.14. Again, we show that the conditions of Theorem 3.6 are satisfied. When $n = 4$, and using (7), we can write the characteristic polynomial we need to study, that has the form $D_3(\lambda) = (A_3 - \lambda)(A_2 - \lambda)(A_1 - \lambda) + \lambda(B_2(A_3 - \lambda) + B_3(A_1 - \lambda))$, where by A_i we abbreviate b_i/a_i and by B_i we abbreviate $c_{i-1}^2/a_{i-1}a_i$. Next we argue that all roots of D_3 have norm less than 1. By assuming strong monotonicity, Theorem 3.13 says that all real roots have norm between 0 and 1. Hence, we only need to check any complex roots. All we need to use below is that $0 \leq A_i, B_i < 1$, and this follows by assuming simple (speed) monotonicity.

Since D_3 is of degree 3, it always has a real root, call it r, and at most two complex roots (that are conjugate to each other), say with norm $\|\rho\|$. Since the constant term of D_3 is $A_1A_2A_3$, it follows that $\|\rho\| = \frac{A_1A_2A_3}{r}$. Next we prove that $r \geq \min\{A_1, A_2, A_3\}$, concluding what we need. Indeed, consider the polynomial

$$\begin{array}{l} \frac{1}{B_2 + B_3} D_3(\lambda) \\ \quad = \begin{array}{l} \frac{1}{B_2+B_3}(A_3 - \lambda)(A_2 - \lambda)(A_1 - \lambda) \\ + \lambda\left(\frac{B_2}{B_2+B_3}(A_3 - \lambda) + \frac{B_3}{B_2+B_3}(A_1 - \lambda)\right) \end{array} \end{array}$$

which clearly has the same roots as $D_3(\lambda)$. Now, the root r above is a value for λ that satisfies the following equality $\frac{1}{B_2+B_3}(A_3 - \lambda)(A_2 - \lambda)(A_1 - \lambda) = -\lambda\left(\frac{B_2}{B_2+B_3}(A_3 - \lambda) + \frac{B_3}{B_2+B_3}(A_1 - \lambda)\right)$. The left-hand-side polynomial, which is of degree 3, has real roots A_1, A_2, A_3, and most importantly it is decreasing for all $x \leq \min\{A_1, A_2, A_3\}$ and for all $\lambda \geq \max\{A_1, A_2, A_3\}$. The right-hand-side polynomial is of degree 2, and has two real roots. One of them is 0, and the other, call it $\bar{r}$, is a convex combination of A_1, A_3, hence we have $\min\{A_1, A_3\} \leq \bar{r} \leq \max\{A_1, A_3\}$. Moreover, the degree 2 polynomial is negative for $0 < \lambda < \bar{r}$ and positive for $\lambda > \bar{r}$, and is increasing for all $\lambda \geq \bar{r}$. Since $\bar{r}$ is in the line segment between $\min\{A_1, A_3\}, \max\{A_1, A_3\}$, it must be the case that the graphs of the two polynomials intersect for some λ between $\min\{A_1, A_2, A_3\}$ and $\max\{A_1, A_2, A_3\}$.Therefore, $r \geq \min\{A_1, A_2, A_3\}$ as wanted. ■

4 CONCLUSION

Patrolling a given domain with a swarm of two-speed robots is a challenging problem with interesting tradeoffs. Its difficulty, even for patrolling a segment, is due to the fact that there are many patrolling strategies to be taken into account. We gave an optimal offline algorithm for any robot collection and optimal dynamic schedules for two robots. We also gave an efficient algorithm deciding self-stabilization of a distributed schedule for the case of regular dynamical systems. We proved that the distributed algorithm is self-stabilizing for up to four robots whose speeds satisfy a certain monotonicity property. However, convergence of our distributed algorithm for more than

four robots (whose speeds satisfy a certain monotonicity property) remains open. The case of different domains is interesting and can be surprisingly demanding.

ACKNOWLEDGEMENTS

For the first and third author, this research was supported in part by NSERC grants.

REFERENCES

Agmon, N., Kraus, S., and Kaminka, G. A. (2008). Multi-robot perimeter patrol in adversarial settings. In *ICRA*, pages 2339–2345.

Almeida, A., Ramalho, G., Santana, H., Tedesco, P. A., Menezes, T., Corruble, V., and Chevaleyre, Y. (2004). Recent advances on multi-agent patrolling. In *SBIA*, pages 474–483.

Alpern, S., Morton, A., and Papadaki, K. (2009). Optimizing randomized patrols. *Operational Research Group, London School of Economics and Political Science*.

Alpern, S., Morton, A., and Papadaki, K. (2011). Patrolling games. *Operations research*, 59(5):1246–1257.

Amigoni, F., Basilico, N., Gatti, N., Saporiti, A., and Troiani, S. (2010). Moving game theoretical patrolling strategies from theory to practice: An usarsim simulation. In *ICRA*, pages 426–431.

Angluin, D., Aspnes, J., Diamadi, Z., Fischer, M., and Peralta, R. (2006). Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18(4):235–253.

Angluin, D., Aspnes, J., Eisenstat, D., and Ruppert, E. (2007). The computational power of population protocols. *Distributed Computing*, 20(4):279–304.

Beauquier, J., Burman, J., Clement, J., and Kutten, S. (2010). On utilizing speed in networks of mobile agents. In *Proceeding of the 29th ACM SIGACT-SIGOPS Symposium on Principles of distributed computing*, pages 305–314. ACM.

Chevaleyre, Y. (2004). Theoretical analysis of the multi-agent patrolling problem. In *IAT*, pages 302–308.

Cieliebak, M., Flocchini, P., Prencipe, G., and Santoro, N. (2012). Distributed computing by mobile robots: Gathering. *SIAM J. Comput.*, 41(4):829–879.

Collins, A., Czyzowicz, J., Gasieniec, L., Kosowski, A., Kranakis, E., Krizanc, D., Martin, R., and Morales Ponce, O. (2013). Optimal patrolling of fragmented boundaries. In *SPAA*.

Czyzowicz, J., Gasieniec, L., Kosowski, A., and Kranakis, E. (2011). Boundary patrolling by mobile agents with distinct maximal speeds. *Algorithms–ESA 2011*, pages 701–712.

Czyzowicz, J., Kranakis, E., and Pacheco, E. (2013). Localization for a system of colliding robots. In *ICALP (2)*, pages 508–519.

Dijkstra, E. W. (1982). *Selected writings on computing: a personal perspective*. Springer-Verlag New York, Inc.

Dumitrescu, A., Ghosh, A., and Csaba, D. T. (2014). On fence patrolling by mobile agents. *CoRR*, abs/1401.6070.

Elmaliach, Y., Agmon, N., and Kaminka, G. A. (2009). Multi-robot area patrol under frequency constraints. *Ann. Math. Artif. Intell.*, 57(3-4):293–320.

Elmaliach, Y., Shiloni, A., and Kaminka, G. A. (2008). A realistic model of frequency-based multi-robot polyline patrolling. In *AAMAS (1)*, pages 63–70.

Elor, Y. and Bruckstein, A. M. (2010). Autonomous multi-agent cycle based patrolling. In *ANTS*, pages 119–130.

Hare, J., Gupta, S., and Wilson, J. (2015). Decentralized smart sensor scheduling for multiple target tracking for border surveillance. In *ICRA*, pages 3265–3270. IEEE.

Hazon, N. and Kaminka, G. A. (2008). On redundancy, efficiency, and robustness in coverage for multiple robots. *Robotics and Autonomous Systems*, 56(12):1102–1114.

Kawamura, A. and Kobayashi, Y. (2012). Fence patrolling by mobile agents with distinct speeds. In *ISAAC*, pages 598–608.

Machado, A., Ramalho, G., Zucker, J.-D., and Drogoul, A. (2002). Multi-agent patrolling: An empirical analysis of alternative architectures. In *MABS*, pages 155–170.

Marden, M. (1949). *The Geometry of the Zeros of a Polynomial in a Complex Variable*, volume 3 of *Math. Surv.* AMS.

Marino, A., Parker, L. E., Antonelli, G., and Caccavale, F. (2009). Behavioral control for multi-robot perimeter patrol: A finite state automata approach. In *ICRA*, pages 831–836.

Pasqualetti, F., Franchi, A., and Bullo, F. (2010). On optimal cooperative patrolling. In *CDC*, pages 7153–7158.

Yanovski, V., Wagner, I. A., and Bruckstein, A. M. (2003). A distributed ant algorithm for efficiently patrolling a network. *Algorithmica*, 37(3):165–186.