Boundary Patrolling by Mobile Agents with Distinct Maximal Speeds

Jurek Czyzowicz[1], Leszek Gąsieniec[2], Adrian Kosowski[3], and Evangelos Kranakis[4]

[1] Université du Québec en Outaouais, Gatineau, Québec J8X 3X7, Canada
[2] University of Liverpool, Liverpool L69 3BX, UK
[3] INRIA Bordeaux Sud-Ouest, LaBRI, 33405 Talence, France
[4] Carleton University, Ottawa, Ontario K1S 5B6, Canada

Abstract. A set of k mobile agents are placed on the boundary of a simply connected planar object represented by a cycle of unit length. Each agent has its own predefined maximal speed, and is capable of moving around this boundary without exceeding its maximal speed. The agents are required to protect the boundary from an intruder which attempts to penetrate to the interior of the object through a point of the boundary, unknown to the agents. The intruder needs some time interval of length τ to accomplish the intrusion. Will the intruder be able to penetrate into the object, or is there an algorithm allowing the agents to move perpetually along the boundary, so that no point of the boundary remains unprotected for a time period τ? Such a problem may be solved by designing an algorithm which defines the motion of agents so as to minimize the *idle time* I, i.e., the longest time interval during which any fixed boundary point remains unvisited by some agent, with the obvious goal of achieving $I < \tau$.

Depending on the type of the environment, this problem is known as either *boundary patrolling* or *fence patrolling* in the robotics literature. The most common heuristics adopted in the past include the *cyclic strategy*, where agents move in one direction around the cycle covering the environment, and the *partition strategy*, in which the environment is partitioned into sections patrolled separately by individual agents. This paper is, to our knowledge, the first study of the fundamental problem of boundary patrolling by agents with distinct maximal speeds. In this scenario, we give special attention to the performance of the cyclic strategy and the partition strategy. We propose general bounds and methods for analyzing these strategies, obtaining exact results for cases with 2, 3, and 4 agents. We show that there are cases when the cyclic strategy is optimal, cases when the partition strategy is optimal and, perhaps more surprisingly, novel, alternative methods have to be used to achieve optimality.

Keywords: mobile agents, boundary patrolling, fence patrolling, idleness.

C. Demetrescu and M.M. Halldórsson (Eds.): ESA 2011, LNCS 6942, pp. 701–712, 2011.

1 Introduction

Consider a Jordan curve $\mathcal{C}$ forming the boundary of a geometric, planar, simply connected object. On the curve are placed k mobile agents, each agent capable of moving at any speed without exceeding its speed limit. The maximal speeds of the agents may be distinct, and the agents are allowed to walk in both directions along $\mathcal{C}$. The main goal is to design the agents' movements so that the maximal time between two consecutive visits to any fixed point of the boundary is minimized. The studied problem has genuine applications. For example, in order to prevent an intruder from penetrating into a protected region, the boundary of the region must be monitored, often with the aid of moving agents such as walking guards, illumination rays, cameras, etc. Since the feasibility of an intrusion likely depends on the time during which the intruder remains undiscovered, it is important to design patrolling protocols which minimize the time during which boundary points are unprotected.

Notation. We assume here that the mobile agents are traversing a continuous, rectifiable curve, where we are interested in the total distance travelled along the curve where some parts can be visited more than once. If the curve is also closed it is called a cycle $\mathcal{C}$. Without loss of generality we may assume that the cycle is represented by a *circle*. The set of *mobile agents* $a_1, a_2, \ldots, a_k$ are moving along $\mathcal{C}$. The speed of each agent a_i may vary during its motion, but its absolute value can never exceed its predefined maximal speed v_i. We assume that a positive speed corresponds to the counterclockwise traversal of the circle and the negative speed to the clockwise movement. Without loss of generality we suppose that the agents are numbered so that $v_1 \geq v_2 \geq \ldots \geq v_k > 0$. Using a scaling argument we assume that the length of the circle is equal to 1 (unit of length), and that in one unit of time an agent using constant speed 1 (one unit of speed) makes exactly one complete counterclockwise tour around the circle.

The position of agent a_i at time $t \in [0, \infty)$ is described by the continuous function $a_i(t)$. Hence respecting the maximal speed v_i of agent a_i means that for each real value $t \geq 0$ and $\epsilon > 0$, s.t., $\epsilon v_i < 1/2$, the following condition is true

$$dist(a_i(t), a_i(t+\epsilon)) \leq v_i \cdot \epsilon \tag{1}$$

where $dist(a_i(t), a_i(t+\epsilon))$ denotes the distance along the cycle between the positions of agent a_i at times t and $t+\epsilon$.

Definition 1 (Traversal Algorithm). *A traversal algorithm on the cycle for k mobile agents is a k-tuple $\mathcal{A} = (a_1(t), a_2(t), \ldots, a_k(t))$ which satisfies Inequality (1), for all $i = 1, 2, \ldots, k$.*

Definition 2 (Idle time). *Let $\mathcal{A}$ be a traversal algorithm for a system of k mobile agents traversing the perimeter of a circle with the circumference 1.*

1. *The idle time induced by $\mathcal{A}$ at a point x of the circle, denoted by $I_{\mathcal{A}}(x)$, is the infimum over positive reals $T > 0$ such that for each $K \geq 0$ there exists $1 \leq i \leq k$ and $t \in [K, K+T]$ such that $a_i(t) = x$.*

2. *The idle time of the system of k mobile agents induced by $\mathcal{A}$ is defined by $I_{\mathcal{A}} = \sup_{x \in \mathcal{C}} I_{\mathcal{A}}(x)$, the supremum taken over all points of the circle.*
3. *Finally, the idle time, denoted by I_{opt}, of the system of k mobile agents is defined by $I_{opt} = \inf_{\mathcal{A}} I_{\mathcal{A}}$, the infimum taken over all traversal algorithms $\mathcal{A}$.*

Related Work. Patrolling has been intensely studied in robotics, especially in the last 4-5 years (cf. [3,9,10,11,14,16,21]). It is often viewed as a version of *coverage*, a central task in robotics. It is defined as the act of surveillance consisting in walking around an area in order to protect or supervise it. Patrolling is useful, e.g., to determine objects or humans that need to be rescued from a disaster environment. Network administrators may use mobile agent patrols to detect network failures or to discover web pages which need to be indexed by search engines, cf. [16]. Patrolling is usually defined as a perpetual process performed in a static or in a dynamically changing environment.

Notwithstanding several interesting applications and its scientific interest, the problem of boundary and area patrolling has been studied very recently (cf. [2,10,11,19]). On multiple occasions, patrolling has been dealt with using an ad-hoc approach with emphasis on experimental results (e.g. [16]), uncertainty of the model and robustness of the solutions when failures are possible (e.g., [10,11,14]) or non-deterministic solutions (e.g., [2]). In the largely experimental paper [16] one can also find several fundamental theoretical concepts related to patrolling, including models of agents (e.g., visibility or depth of perception), means of communication or motion coordination, as well as measures of algorithm efficiency. In most papers in the domain of patrolling, and also in our paper, algorithm efficiency is measured by its capacity to optimize the frequency of visits to the points of the environment (cf. [3,9,10,11,16]). This criterion was first introduced in [16] under the name of *idleness*. Depending on the approach the idleness is sometimes viewed as the average ([10]), worst-case ([21,5]), probabilistic ([2]) or experimentally verified ([16]) time elapsed since the last visit of the node (see also [3,9]). In some papers the terms of *blanket time* ([21]) or *refresh time* ([19]) are used instead, meaning the similar measure of algorithm efficiency.

Diverse approaches to patrolling based on the idleness criteria were surveyed in [3] — they discussed machine learning methods, paths generated using negotiation mechanisms, heuristics based on local idleness, or approximation to the Traveling Salesmen Problem (TSP). In [4] patrolling is studied as a game between patrollers and the intruder. Some papers solved patrolling problem based on swarm or ant-based algorithms ([12,18,21]). In these approaches agents are supposed to be memoryless (or having small memory), decentralized ([18]), i.e., with no explicit communication permitted with other agents or the central station, with local sensing capabilities (e.g., [12]). Ant-like algorithms usually mark the visited nodes of the graph. The authors of [21] present an evolutionary process. They show that a team of memoryless agents, 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), see also [5].

The author of [9] brings up a theoretical analysis of the approaches to patrolling in graph-based models. The two fundamental methods are referred to as *cyclic strategies*, where a cycle spanning the graph is constructed and the agents consecutively traverse this cycle in the same direction, and *partition-based strategies*, where the region is split into either disjoint or overlapping portions assigned to be patrolled by different agents. The environment and the time considered in the studied models are usually discrete and set in a graph environment. In geometric environments, the skeletonization technique is often applied, where the terrain is first partitioned into cells, and then graph-theoretic methods are used. Usually, cyclic strategies rely either on TSP-related solutions or spanning tree-based approaches. For example, spanning tree coverage, a technique first introduced in [13], was later extended and used in [1,10,14]. This technique is a version of the skeletonization approach where the two-dimensional grid approximating the terrain is constructed and a Hamiltonian path present in the grid is used for patrolling. In the recent paper [19], polynomial-time patrolling solutions for lines and trees are proposed. For the case of cyclic graphs [19] proves the NP-hardness of the problem and a constant-factor approximation is proposed.

Related to our problem is the *lonely runner* conjecture, given this lifelike name in [8], but first stated by J.M. Willis in 1967, [20]. It concerns k runners ($k \geq 2$) running laps on a unit length circular track with constant but pairwise different speeds. It is conjectured that every runner gets at a distance at least $1/k$ along the circular track to every other runner at some time. The conjecture has been proved for up to seven runners. For related work we refer the reader to two recent papers [6] and [7]. Very recently, substantial progress has been announced in an unpublished work [15] using dynamical systems theory. In private communication, the authors point out an equivalence between problems similar to the lonely runner conjecture and certain types of problems from elementary number theory (including Littlewood's, Goldbach's, and Polignac's conjectures).

2 Boundary Patrolling Algorithms

This section contains our results for the patrolling problem with variable-speed agents. The layout of the section is inspired by the categorization from [9] of approaches to patrolling into partition-based strategies (when the environment is partitioned into parts monitored by individual agents) and cyclic strategies (when all agents patrol the environment walking in the same direction along some cycle). In Subsection 2.1 we consider the problem of *fence patrolling*, for which proportional partition-based strategies appear to be the most natural approach. We provide a non-trivial proof that this strategy is indeed optimal for any configuration of speeds for $k = 2$ agents. Next, we consider cyclic strategies for *patrolling a circular boundary.* In Subsection 2.2 we show that such strategies are optimal on the circle for all configurations of speeds with $k \leq 4$ agents, under the additional constraint that all the agents are restricted to motion in the same direction around the boundary. In Subsection 2.3 we show by a technical analysis that the cyclic strategy is optimal for $k = 2$ even for agents which can change direction of motion. Surprisingly, we also show that in this general setting, the cyclic strategy is no longer optimal for $k = 3$ agents, and that a new type of strategy which is neither partition-based nor cyclic achieves a shorter idle time.

Note. Due to space constraints, most proofs (especially the involved proofs of lower bounds) are not present in this version of the paper.

2.1 Fence Patrolling: The Proportional Solution

We first consider the special case with an additional restriction that the boundary contains a special *cutting point* through which no agent is permitted to cross during its movement. This corresponds to the problem of patrolling a segment $S = [0, 1]$ and is known as *fence patrolling* in the robotics literature. We assume that positive speed corresponds to the traversal of the unit segment in the direction from left to right, while negative speed traversal in the opposite direction. We propose the following algorithm:

```
Algorithm $\mathcal{A}_1$ {for $k$ agents to patrol a segment}
1. Partition the unit segment $S$ into $k$ segments, such that the length of the
     $i$-th segment $s_i$ equals $\frac{v_i}{v_1+v_2+\cdots+v_k}$.
2. For each $i = 1, \ldots, k$ place agent $a_i$ at any point of segment $s_i$.
3. For each $i = 1, \ldots, k$ agent $a_i$ moves perpetually at maximal speed
     alternately visiting both endpoints of $s_i$.
```

Proposition 1. *Traversal algorithm $\mathcal{A}_1$ achieves idle time $I = \frac{2}{v_1+v_2+\cdots+v_k}$.*

Proof. Since each agent covers a non-overlapping segment of the circle (except for its endpoints, which may be visited by two agents) the interior points of each segment s_i are visited by the same agent a_i. The infimum of the frequency of visits of point x inside s_i is achieved for x being its endpoint. Since between two consecutive visits to the endpoint x of s_i agent a_i traverses, using its speed v_i, the segment s_i of length $\frac{v_i}{v_1+v_2+\cdots+v_k}$ twice, the idle time of such a point is $I = 2\frac{v_i}{v_1+v_2+\cdots+v_k}/v_i = \frac{2}{v_1+v_2+\cdots+v_k}$. ■

We prove below that the algorithm $\mathcal{A}_1$ is optimal for the case of two agents patrolling a segment.

Theorem 1. *The optimal traversal algorithm for two agents patrolling unit segment $S = [0, 1]$ achieves idle time $I_{opt} = \frac{2}{v_1+v_2}$.*

Proof idea. We suppose, by contradiction, that there exists an algorithm $\mathcal{A}$ with an idle time of $I_{\mathcal{A}} = I_{\mathcal{A}_1} - \epsilon$ for some $\epsilon > 0$. We consider the subsegments S_1 and S_2 forming a decomposition of S, of lengths proportional to the speed bounds of agents a_1 and a_2, respectively ($|S_1| \geq |S_2|$), such that each agent belongs to its subsegment at some specific time. We show that the first agent to visit the common endpoint of S_1 and S_2 has to be the slower agent a_2 (otherwise the other endpoint of S_2 cannot be revisited in time). This forces the faster agent a_1 to visit the other endpoint of S_1, since a_2, in turn, cannot do this in time. As a consequence, a meeting of a_1 and a_2 has to occur. When this meeting happens, we bound from below four values of time, describing the times elapsed from the last visit to each endpoint of S and the times to the next visit to these endpoints. We use for this purpose the information about their speeds and the distance of the point of meeting of the agents from each of the endpoints. We show that the sum of the four considered times is at least equal to twice the idle time of $\mathcal{A}_1$, leading to the conclusion that one of the endpoints must remain unvisited for at least a time of $I_{\mathcal{A}_1}$, a contradiction with $I_{\mathcal{A}} < I_{\mathcal{A}_1}$. ■

We conjecture that Theorem 1 extends to the case of any number of agents.

Conjecture 1. The optimal traversal algorithm for n agents patrolling a segment achieves idle time $I_{opt} = \frac{2}{v_1+v_2+\ldots+v_k}$.

The approach from Proposition 1 results in a 2-approximation.

Proposition 2. *For any environment (a segment or a circle), the idle time of an optimal traversal algorithm for k agents with maximal speeds $v_1, v_2, \ldots, v_k$ is lower-bounded by $I_{opt} \geq \frac{1}{v_1+v_2+\cdots+v_k}$.*

Proof. During any time interval of length I_{opt} all points of the boundary have to be visited by at least one agent, hence the segments of the boundary covered by the corresponding mobile agents must cover the entire boundary. The maximum length of the segment traversed by agent a_i during time I_{opt} equals $I_{opt}v_i$. Since $\sum_{i=1}^{k} I_{opt}v_i \geq 1$ we have $I_{opt} \geq \frac{1}{v_1+v_2+\cdots+v_k}$. ■

The idea of algorithm $\mathcal{A}_1$ was to balance the work of all agents according to their maximal speeds. Hence the unit segment was partitioned in such a way that the idle time for each sub-segment was equal. The above algorithm seems to imply that we should use all available agents in the patrolling process, i.e., not using some of the agents results in a worse idle time. Indeed, it seems that, if some agent a_i is not being used (i.e., it stays motionless), the sub-segment patrolled by agent a_i in algorithm $\mathcal{A}$ must be covered, entirely or partially, by some other agent a_j. Since a_j must then cover a longer segment using the same maximal speed, this would result in longer idle time for its sub-segment and, consequently, in longer idle time for the algorithm. However the results of the next section indicate that this intuitive observation is not true in the case of patrolling a circle.

Patrolling a segment seems to suffer from an inherent weakness. An agent, after reaching an endpoint of the segment (or its sub-segment), performs a traversal in the opposite direction, first moving through points which were visited very recently, and only revisiting its starting location after two complete traversals of the segment. On the other hand, algorithms on the circle can be designed so that at any time the agent re-visits the location which has been waiting the longest. Consequently, optimal algorithms for a single agent of maximal speed 1 offer idle time 1 for the unit circle and idle time 2 for the unit segment. Therefore, in order to profit from the circle topology it may seem natural to try to traverse the cycle always in the same direction. In the next section we consider the case when all the agents are *required* to do so.

2.2 Patrolling a Unidirectional Boundary

We consider the case of the circle which must be traversed by all agents always in the same direction, say counterclockwise. In another words, every agent must use a strictly positive speed at each period of its movement. We now show an algorithm for which we can prove optimality for a small number of agents. The idea of the algorithm is to use only a subset of r agents having sufficiently high maximal speeds. These agents are spaced on the circle at even distances and ordered to move counterclockwise using the constant speed v_r — the maximal speed of the slowest agent from the subset. The value of r is chosen such that the idle time is minimized.

```
Algorithm $\mathcal{A}_2$ {for $k$ agents to patrol a unidirectional circle}
1. Let $r$ be such that $\max_{1 \le i \le k} iv_i = rv_r$.
2. Place the agents $a_1, a_2 \dots a_r$ at equal distances of $\frac{1}{r}$ around the circle
3. For each $i = 1, \dots, r$ agent $a_i$ moves perpetually counterclockwise
     around the circle at speed $v_r$.
4. None of the agents $a_{r+1}, a_{r+2} \dots a_k$ are used by the algorithm.
```

Theorem 2. *Consider k agents patrolling a unit circle having positive speeds not exceeding the maximal values $v_1 \ge v_2 \ge \dots \ge v_k > 0$, respectively. Then algorithm $\mathcal{A}_2$ achieves the idle time $I = \frac{1}{\max_{1 \le i \le k} iv_i}$.*

Proof. Suppose that i mobile agents spaced at equal distances around the circle walk with speed v_i. Consider any time t. Each agent a_j must visit at some time $t + \Delta$ the point x which was visited at time t by another agent which was predecessor of a_j on the circle. The distance to this point x is equal to $1/i$. Using speed v_i reaching point x takes time $\Delta = \frac{1}{iv_i}$. ■

Note that the value of $1 \le r \le k$ such that $rv_r = \max_{1 \le i \le k} iv_i$ is the best possible, since

$$\frac{1}{rv_r} = \frac{1}{\max_{1 \le i \le k} iv_i} = \min_{1 \le i \le k} \frac{1}{iv_i}$$

We prove that algorithm $\mathcal{A}_2$ is optimal for any setting involving less than 5 mobile agents. For this purpose we introduce first the notion of a *visit pattern.*

Definition 3. *Suppose that k agents $a_1, a_2 \dots, a_k$ patrol a unit circle according to algorithm $\mathcal{A}$. We say that algorithm $\mathcal{A}$ admits the visit pattern $P = i_1 i_2 \dots i_p$, where $1 \le i_j \le k$ for $j = 1, 2, \dots, p$, if there exists time t and a point x on the circle, such that starting at time t the following, consecutive visits of point x are made (in this order) by the agents $a_{i_1}, a_{i_2} \dots a_{i_p}$.*

For example, the visit pattern 131 implies that at some time during the execution of the algorithm, a certain point x on the circle is visited by agent a_1, the next visit to x is made by agent a_3 and the subsequent visit is made again by agent a_1. The notion of the visit pattern may be also extended to the case when more than one agent visits point x at the same time.

We prove below that algorithm $\mathcal{A}_2$ is optimal for the case of $k < 5$ agents. We first make an important observation, which we will use in this proof.

Observation. In order to prove that no algorithm $\mathcal{A}'$ provides a better idle time than algorithm $\mathcal{A}_2$ it is sufficient to focus our attention on a class of algorithms $\mathcal{A}'$ in which the agents $a_1, a_2, \ldots, a_k$ have speeds restricted to, respectively, $v_1 = v, v_2 = v/2, \ldots, v_k = v/k$, for some $v > 0$.

Indeed, assume that the algorithm $\mathcal{A}'$ must run for any sequence of speeds $v_1' \geq v_2' \geq \cdots \geq v_k'$, such that $rv_r' = \max_{1 \leq i \leq k} iv_i'$. Suppose that we change the speeds of the agents to $v_1 = rv_r', v_2 = rv_r'/2, \ldots, v_k = rv_r'/k$. Observe that $v_r = v_r'$ and $v_i \geq v_i'$, for $i \neq r$. Increasing the speeds of the agents may never result in a worse idle time since the algorithm may always choose to use some smaller speeds.

We show below that if some algorithm $\mathcal{A}'$ provides a better idle time than algorithm $\mathcal{A}_2$, then between any two consecutive visits to any point x of the circle by some agent a_c, $1 \leq c \leq k$, there must be at least c visits of point x which are made by other agents.

Lemma 1. *Consider any algorithm $\mathcal{A}$ run for the agents speeds $v_1 = v, v_2 = v/2, \ldots, v_k = v/k$. If algorithm $\mathcal{A}$ admits a visit pattern $ci_1i_2 \ldots i_dc$, where $1 \leq d < c \leq k$, then algorithm $\mathcal{A}$ cannot result in a better idle time than $I = \frac{1}{\max_{1 \leq i \leq k} iv_i} = 1/v$ (i.e., the idle time of algorithm $\mathcal{A}_2$).*

Proof. Suppose that algorithm $\mathcal{A}$ admits some visit pattern $ci_1i_2 \ldots i_dc$ for some point x of the circle, where $1 \leq d < c \leq k$. Since $v_c = v/c$, the time T between the first and the last visit of point x by a_c is at least

$$T \geq \frac{1}{v_c} = \frac{c}{v}.$$

This time interval is split into $d+1$ sub-intervals by the visits of agents $a_{i_1}, a_{i_2}, \ldots, a_{i_d}$. The largest such sub-interval T' must be at least equal to their average, i.e., since $d + 1 \leq c$,

$$T' \geq \frac{T}{d+1} \geq \frac{c}{v(d+1)} \geq \frac{c}{vc} = \frac{1}{v}.$$

Hence, the idle time of algorithm $\mathcal{A}$ is not less than that of algorithm $\mathcal{A}_2$. ■

The visit patterns from the statement of Lemma 1, which can never be admitted by any algorithm supposedly offering a better idle time than $\mathcal{A}_2$, will be called *forbidden patterns.*

We now show that for any case of $k < 5$ agents the algorithm $\mathcal{A}_2$ is optimal.

Theorem 3. *Consider the case of $k < 5$ agents patrolling a unit circle having positive speeds not exceeding the maximal values $v_1 \geq v_2 \geq \ldots \geq v_k > 0$, respectively. Algorithm $\mathcal{A}_2$ achieving the idle time $I = \frac{1}{\max_{1 \leq i \leq k} iv_i}$ is optimal for this case.*

Proof idea. The proof is performed for $k = 4$, and implies the result also for $k < 4$. The idea of the proof for $k = 4$ is the following. We show first that, when some particular pairs of agents meet, a forbidden pattern is forced. As a consequence we will show that no agent can overtake agent a_4 more than once. Therefore, the number of agents' visits to any point x of the circle is at most four times the number of visits of this point by agent a_4 (plus a small constant). Hence for any $\epsilon > 0$ no algorithm can offer the idle time smaller than $\frac{1}{4v_4} - \epsilon$ and the idle time of algorithm $\mathcal{A}_2$ cannot be improved. ■

We believe that Theorem 3 extends to any number of agents, hence we propose the following conjecture:

Conjecture 2. In the case of k agents patrolling the circle, which have to use positive speeds not exceeding their respective maximal values $v_1 \geq v_2 \geq \ldots \geq v_k > 0$, the algorithm $\mathcal{A}_2$ achieving the idle time $I = \frac{1}{\max_{1 \leq i \leq k} iv_i}$ is optimal.

One can show that the idle time of $\frac{1}{\max_{1 \leq i \leq k} iv_i}$, achieved with positive speeds, is always within a multiplicative factor of $(1 + \ln k)$ away from the theoretical lower bound on idle time of $\frac{1}{\sum_{1 \leq i \leq k} v_i}$ (Proposition 2), which holds even when we allow agents moving in both directions. In this context, it is natural to ask whether using positive speeds by all agents, i.e., traversing the circle in the same direction is always the best strategy. This problem is addressed in the next section.

2.3 Allowing Movement in Both Directions

In this section we consider patrolling of a circle which may be traversed in both, clockwise and counterclockwise directions. It is important to understand whether this additional ability of agents to change directions may be sometimes useful and whether it may lead to a technique better than algorithm $\mathcal{A}_2$ from the previous section. We show that this is not the case for any setting involving $k = 2$ agents. We show, however, that there are settings already for $k = 3$ agents, when using negative speeds by the participating agents leads to a better idle time.

Theorem 4. *Consider two agents patrolling a unit circle with the possibility of movement in both directions. For any pair of maximal speeds $v_1 \geq v_2$ no algorithm $\mathcal{A}$ permitting agents' movement in both directions of the circle can achieve an idle time $I_{\mathcal{A}}$ which is better than $\min\{\frac{1}{v_1}, \frac{1}{2v_2}\}$ (i.e., the idle time provided by algorithm $\mathcal{A}_2$).*

Proof idea. We first show that a pair of agents a_1 and a_2, following algorithm $\mathcal{A}$, may never meet. Then, we prove that at some point of time, the circle is uniquely decomposed into a pair of arcs, each of which contains the set of points which were last visited by the first and by the second agent, respectively. Moreover, the location of each agent is confined to its corresponding arc. The endpoints of the arcs perform a continuous motion in time. We show that one of the endpoints of the arc of agent a_1 is never visited by this agent, whereas the other endpoint is visited regularly for arbitrarily large values of time. Without loss of generality, we will say that the *clockwise orientation* of the circle is the orientation given by traversing the arc of agent a_1 from the endpoint which this agent does not visit, to the other endpoint. We prove that the trajectory of the arc endpoint which is never visited by a_1 is necessarily a monotonous clockwise rotation around the circle (without a limit point). Within every such rotation, we show that each point of the circle is visited by agent a_1 at least twice, and that the distance traversed clockwise in between any two such visits to the same point by agent is at most $\frac{1}{4}$. Taking into account these observations, and the fact that the length of the arc corresponding to agent a_1 is always greater than $\frac{1}{2}$, we perform the main part of the proof which consists in a technical analysis of the trajectory of agent a_1. We finally show that the idle time of algorithm $\mathcal{A}$ cannot be better than $\min\{\frac{1}{v_1}, \frac{1}{2v_2}\}$. ■

The last theorem implies the following

Corollary 1. *There exists settings when the optimal algorithm solving the boundary patrolling problem does not use some of the agents.*

Indeed, from Theorem 4 it follows that algorithm $\mathcal{A}_2$ is optimal for two agents patrolling a circle. However if agent a_2 has a speed at least twice slower than a_1, patrolling by a_1 (disregarding the behavior of agent a_2), using its maximal speed, results in the optimal idle time.

The next theorem gives an example of the setting for three agents, where using both directions is sometimes necessary to achieve the optimal idle time. This would not be true for every speed setting for three agents (e.g., clearly not for three agents with equal maximal speeds).

Theorem 5. *Consider $k = 3$ agents patrolling a unit circle with the possibility of movement in both directions. There exist settings such that in order to achieve the optimal idle time, some agents need to move in both directions.*

Proof. Consider the setting with $k = 3$ agents having maximal speeds $v_1 = 1, v_2 = v_1/2, v_3 = v_1/3$. Suppose that all agents move in the same counterclockwise direction around the unit circle. By Theorem 3, $\mathcal{A}_2$ is the optimal algorithm for this case and it achieves the idle time $I_{\mathcal{A}_2} = \frac{1}{\max_{1\leq i\leq k} iv_i} = 1$. In order to prove the claim of our theorem we need to give an example of an algorithm $\mathcal{A}'$ controlling the movement of the three agents using $v_1, v_2 = v_1/2, v_3 = v_1/3$ as their maximal speeds, such that some agents move in both directions, and such that its idle time $I_{\mathcal{A}'} < 1$. Using the classical concept of distance-time graphs due to E.J. Marey [17] the movement of the agents is described at Fig. 1, where the horizontal axis represents time and the vertical axis refers to the position of the corresponding agent on the circle (with 0 and 1 representing the same point).

A detailed discussion of the construction and its analysis is omitted in this version of the paper. We show that the maximal idle time is equal to $\frac{35}{36} < 1 = I_{\mathcal{A}_2}$, proving the claim of the theorem. ■

Note that for the speed setting from the example from Figure 1 the algorithm $\mathcal{A}_1$ is not optimal as well since the partition of the circle into segments proportional

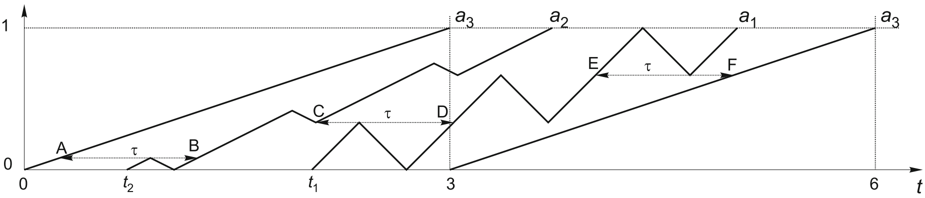

Fig. 1. Example of an algorithm achieving an idle time of $\frac{35}{36}$ for three robots with speeds $v_1 = 1$, $v_2 = 1/2$, $v_3 = 1/3$

to the agents' speeds would result in the idle time of $\frac{2}{1+\frac{1}{2}+\frac{1}{3}} > 1$. Therefore for this speed setting, neither the partition strategy of algorithm $\mathcal{A}_1$, nor the cyclic strategy of algorithm $\mathcal{A}_2$ is the best.

3 Conclusion and Open Problems

The problem of boundary patrolling has been lately intensively studied by the robotics research community. Optimality measures related to idleness — the minimization of the time when a boundary point remained unvisited — are applied in the vast majority of work in the field. This is also the measure of algorithm efficiency adopted in our paper. We have shown that for agents with distinct maximum speeds, in some settings of the problem the decisions made by the optimal algorithms are to some extent counter-intuitive. For example, we showed that it is sometimes advantageous not to make use of all of the agents. We showed that the partition strategy, represented by algorithm $\mathcal{A}_1$, and the cyclic strategy, performed by algorithm $\mathcal{A}_2$ are indeed optimal in certain cases. However, as follows from Theorem 5, in some settings the optimal idle time cannot be obtained by either of these two strategies.

Several problems remain open. The fundamental open problem is to design the optimal strategy for any configuration of speed settings of k agents patrolling a circle with the unit circumference. Two other problems, stated as Conjecture 1 and Conjecture 2, concern the extension of our results to a larger number of agents. Some other important questions include the following. Is it possible that in some setting the optimal algorithm needs agents to overtake (pass) one another? What is the solution to the problem in the case when agents have some radius of visibility (i.e., a point is considered visited when an agent is at some ε neighborhood of the point), potentially different for different agents? Finally, it is interesting to study local coordination scenarios which would allow variable-speed agents to stabilize to an efficient patrolling scheme in a distributed manner.

References

1. Agmon, N., Hazon, N., Kaminka, G.A.: The giving tree: constructing trees for efficient offline and online multi-robot coverage. Ann. Math. Artif. Intell. 52(2-4), 143–168 (2008)