Boundary Patrolling Problems

E. Kranakis
Carleton University
School of Computer Science
Ottawa, Ontario, K1S 5B6, Canada

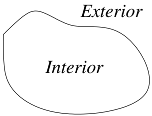
May, 2013

Outline

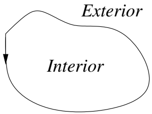
Boundary Patrolling

Motivation

- **Patrolling problems in computer games**
  - Safeguard a given region/domain/territory from enemy invasions.

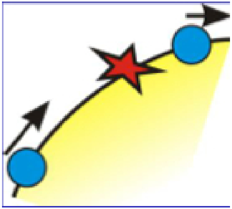


- **Patrolling problems in robotics**
  - Patrolling is defined as the perpetual process of walking around an area in order to protect or supervise it.



Problem

- A set of k mobile agents are placed on the boundary of a terrain.
- An intruder attempts to penetrate to the interior of the terrain through a point of the boundary, unknown to and unseen by the agents.
- The intrusion requires some period of time t.



- The agents are required to protect the boundary, arriving before the intrusion is complete.

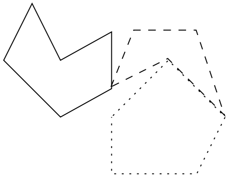
Setting

- Each agent i has its own predefined maximal speed v_i, for $1, 2, \ldots, k$.
- Agents are deployed on the boundary and programmed to move around the boundary, without exceeding their maximum speed.
- **Question:**
  for given speeds $\{v_1, v_2, \ldots, v_k\}$ and time τ, does there exist a deployment of agents which protects the boundary from any intruder with intrusion time not exceeding τ?

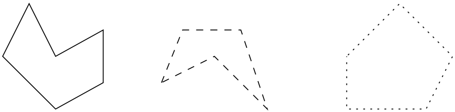
Efficiency

- How do you optimize the frequency of visits to the points of the environment?
- *Idleness (or refresh time:)* is the time elapsed since the last visit of the node.
    - Idleness can be average, worst-case, experimentally verified, etc,...
- In a way, given the input parameters you want to know what is the best effort result you can accomplish!

- The graph (or environment) to be patrolled is usually approximated by a set of subgraphs forming a (*skeletonization*).

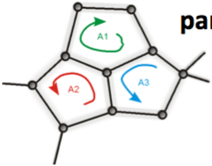


- A skeleton of the environment is defined over which patrolling is being conducted by the robots.

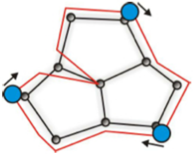


Related Work: Mostly Heuristics

- **Heuristics:**
  - Based on variants of TSP for agents with limited resources [survey - Almeida et al. 2004]
- **No coordination:** reasonable only for very simple agents
  - Many random walks have smaller refresh time than one [Alon et al. 2008, Elsaesser-Sauerwald 2009]
- **Centralized coordination:** two main types of heuristics [Chevaleyre 2004]



partition-based strategies

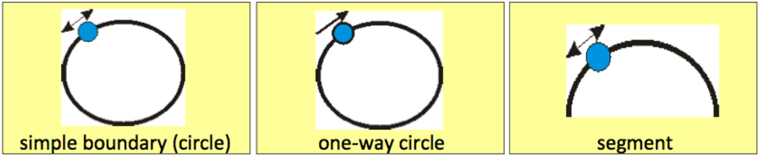


cyclic strategies

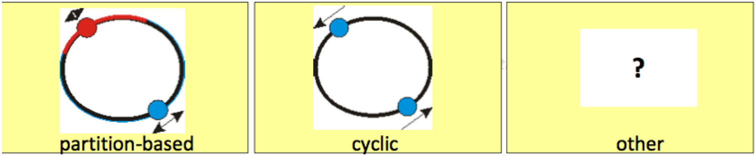
- **Distributed coordination:** using local information exchange
  - tokens, pebbles, white boards, ant/swarm algorithms [Yanovski et al. 2001, Elor-Bruckstein 2010]

Results

- **Goal:** minimize maximal idle time for a set of boundary patrolling robots with distinct maximal speeds $(v_1, v_2, \ldots, v_k)$
- **Studied Environments**



- **Studied Strategies**



Traversal Algorithm

- 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

$$dist(a_i(t), a_i(t+\epsilon)) \leq v_i \cdot \epsilon \quad (1)$$

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 (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 $i = 1, 2, \ldots, k$.

Idle Time

Definition (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 $\mathcal{A}$.

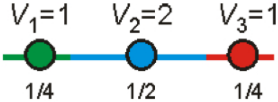
Approach and Methodology

- Domain being traversed by the robots.
  - partition
  - decomposition
- Visualizing the movement of the robots
  - Using the classical concept of Distance Line Graphs: E. J. Marey. La méthode graphique. 1878.
  - 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).
- Proofs often elaborate.

Three Algorithms

- Algorithm 1. Proportional Partition
  - Suitable for Segments
- Algorithm 2. Uniform-Cyclic
  - Suitable for Cycles
- Algorithm 3: Hybrid
  - Combination of the above

- **Algorithm 1. Proportional Partition**
  for k agents with maximal speeds $(v_1, v_2, \ldots, v_k)$
  1. Partition the unit segment 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, place the i-th agent at any point of segment s_i.
  3. For each i, the i-th agent moves perpetually at maximal speed, alternately visiting both endpoints of s_i.

Idle time of algorithm Proportional Partition

- On unit-length segment or circle, algorithm achieves idle time:

$$I = \frac{2}{v_1 + v_2 + \cdots + v_k}.$$

- The idle time of any strategy on the unit-length segment or circle is bounded from below by:

$$I_{OPT} \geq \frac{1}{v_1 + v_2 + \cdots + v_k}.$$

Performance

- In general,
  Proportional-Partition is a 2-approx strategy.
- **On the circle**
  there are some configurations for which the approximation ratio of 2 is tight.
- **On the segment**
  - **Theorem.** Proportional-Partition Strategy is optimal for 2 agents, and for any max speeds v_1, v_2.
  - **Conjecture.** Proportional-Partition Strategy is optimal for any number of agents.

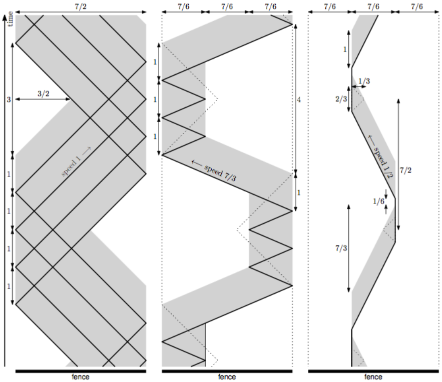
Status of the Conjecture on a Segment

1. Recall the conjecture that the maximum length of the fence that can be patrolled is $(v_1 + \cdots + v_k)/2$, which is achieved by the simple strategy where each agent i moves back and forth in a segment of length $v_i/2$.
2. Proportional-Partition Strategy is optimal for any k and any speeds s.t. $v_1 = v_2 = \cdots = v_k$.
3. Akitoshi Kawamura and Yusuke Kobayashi [1] prove
   - Proportional-Partition Strategy is optimal for $k = 3$ robots.
   - Proportional-Partition Strategy is not optimal for $k = 6$ robots.
4. Nothing known for $k = 4, 5$ and $k > 6$.

[1] Fence patrolling by mobile agents with distinct speeds, ISAAC 2012

Proportional-Partition not optimal for $k = 6$ robots

- Six robots with speeds $1, 1, 1, 1, 7/3, 1/2$ patroling a fence of length $7/2$ with idle time $T = 1$.



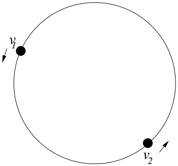
- Proportional Partition can only patrol the length $\frac{1+1+1+1+7/3+1/2}{2} = 41/12 < 7/2$ in idle time $T = 1$.

Cyclic Strategies

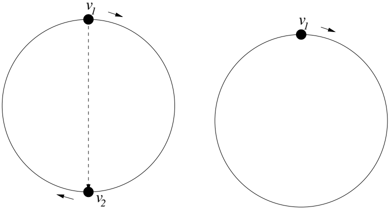
- **Goal:** deploy (some of) the robots, all moving around the circle at the same speed, with equal spacing.
- **Algorithm 2. Uniform-Cyclic**
  for k agents with maximal speeds $(v_1, v_2, \ldots, v_k)$ on the circle
- Let $v_1 \geq v_2 \geq \cdots \geq v_k$.
  1. Choose r from the range $1..k$, so as to maximize: rv_r
  2. Place agents $1, 2, \ldots, r$ at equal distances of $1/r$ around the circle.
  3. Agents $1, 2, \ldots, r$ move perpetually counterclockwise around the circle at speed v_r.
  4. Agents $r+1, r+2, \ldots, k$ are not used by the algorithm.
- **NB** Can employ this algorithm in either uni-directional or bi-directional circle.

Uniform Cyclic for $k = 2$

- Consider two robots with arbitrary max speeds $v_1 \geq v_2$



- In an optimal algorithm either 1) both robots patrol (left: $v_1 < 2v_2$) with the speed of the slower or 2) only the faster patrols (right: $v_1 \geq 2v_2$) while the slower robot is obsolete.



- Algorithm Uniform-Cyclic achieves idle time: $I = \frac{1}{\max_r rv r}$.

Idle time of algorithm Uniform Cyclic

- Algorithm Uniform-Cyclic achieves idle time: $I = \frac{1}{\max_r rvr}$.
- Since $I_{OPT} \geq \frac{1}{v_1+v_2+\cdots+v_k}$, in general, Uniform-Cyclic is a $(\ln k + 1)$-approximation strategy.
- On the **bi-directional** circle...
  - **Theorem.** Uniform-Cyclic is optimal for $k = 2$ agents, for any v_1, v_2.
  - **Note:** Uniform-Cyclic is sometimes not optimal for $k \geq 3$.

On the **uni-directional** circle...

- **Theorem.** Uniform-Cyclic Strategy is optimal for $k \leq 4$ agents, for any set of max. speeds.
- **Conjecture.** Uniform-Cyclic Strategy is optimal for any number of agents.

Proof Outline for $k = 3$ (1/2)

- **Theorem.** On the uni-directional circle, algorithm Uniform-Cyclic is optimal for $k \leq 4$ agents, for any set of max. speeds.
- **Proof for** $k = 3$
- Let $v_1 \geq v_2 \geq v_3$
- Fix an arbitrary point x of the circle.
- Consider the infinite sequence of visits to point x by different agents.
- Define patterns as substrings of this sequence, e.g.:
  - $[1, 3, 1]$ point x is visited by agent 1, next by agent 3, next by agent 1 again.
  - $[2, (13)]$ point x is visited by agent 2, next by agents 1 and 3 (meeting at x).

Proof Outline for $k = 3$ (2/2)

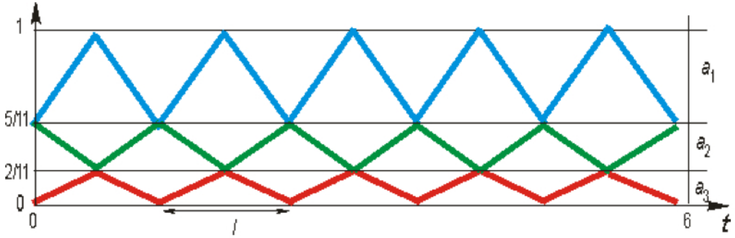
- Consider forbidden patterns: patterns which cannot appear in any strategy which has smaller idle time than algorithm Uniform-Cyclic.
  - E.g.: $[3, 1, 2, 3]$ is a forbidden pattern, since one of the time periods between visits of two successive agents is of duration at least $\frac{1}{3v_3}$.
- All sequences containing the meeting of agents (12) include a forbidden pattern: $[(12), 1][(12), 2][1, (12)][2, (12)][3, (12), 3]$
- Thus, agents 1 and 2 can never meet in a better strategy.
- Likewise, we show that no pair of agents can ever meet, and in the limit, the idle time of Uniform-Cyclic cannot be beaten.

Three Case Studies

- Can the ability of agents to change directions improve the idle time?
  - We have shown that this is not the case for any setting involving $k = 2$ agents.
  - However, there are settings already for k = 3 agents, when using negative speeds (i.e., reversing direction) by the participating agents leads to a better idle time.
- Three Case Studies
  - Proportional Partition Algorithm
  - Uniform Cyclic Algorithm
  - Hybrid Algorithm

Case Study: Proportional Partition

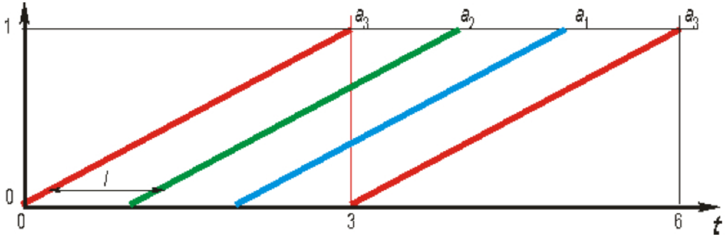
- Horizontal axis represents time; vertical axis refers to the position of the corresponding agent on the circle (with 0 and 1 representing the same point)
- Two-directional circle, $k = 3, v_1 = 1, v_2 = 1/2, v_3 = 1/3$



- Proportional-Partition Algorithm:

$$l = \frac{1}{1 + 1/2 + 1/3} = \frac{12}{11}$$

- Horizontal axis represents time; vertical axis refers to the position of the corresponding agent on the circle (with 0 and 1 representing the same point)
- Two-directional circle, $k = 3, v_1 = 1, v_2 = 1/2, v_3 = 1/3$

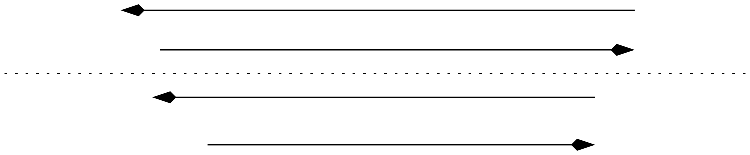


- Uniform-Cyclic:

$$l = 1$$

Is There a Better Strategy?

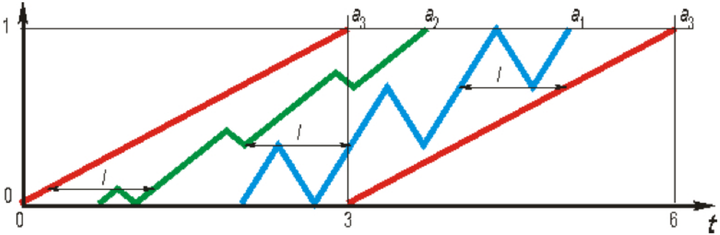
- A "partition" strategy may not necessarily be optimal.
- Instead robots are allocated "overlapping" subdomains dynamically.
- In the picture below



by reversing direction, a faster robot can help a slower moving robot reduce the idle time.

Case Study: Hybrid

- Horizontal axis represents time; vertical axis refers to the position of the corresponding agent on the circle (with 0 and 1 representing the same point)
- There exist settings such that in order to achieve the optimal idle time, some agents need to move in both directions.
- two-directional circle, $k = 3, v_1 = 1, v_2 = 1/2, v_3 = 1/3$

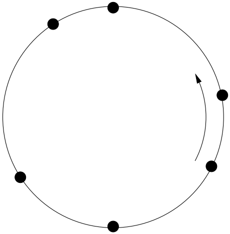


- A hybrid strategy:

$$I = 35/36 < 1$$

Lonely Runners

- **Problem** There are k runners on a unit circle, running perpetually around with **constant** speeds $\{v_1, v_2, \ldots, v_k\}$.
- They all start together from the same starting spot and continue running at their own speed forever.
- They never vary their speeds, stop, or interfere with each other.



- **Question:** Is it always true that for every runner, at some moment in time, their distance from the nearest runner will be at least $1/k$?

Progress?

- 40 years of incremental progress.
- Solved for $k \leq 6$.

k	Year Proved	Proved by
3	1972	Betke and Wills; Cusick
4	1984	Cusick and Pomerance; Bienia et al.
5	2001	Bohman, Holzman, Kleitman; Renault
6	2008	Barajas and Serra

- Problem difficult even for specific speeds.
- Some very recent progress using dynamic systems theory.
- Problem related to Diophantine approximation theory (Littlewood's conjecture) and number theory (Goldbach's and Polignac's conjectures).

Example: Runners from the Origin

- k runners start at 0, running at speeds $1, 2, \ldots, k$.
- **Question:** Will there be a time when each runner will be distance at least $\frac{2}{k}$ from the start?
- **Claim:** With positive probability there will be no runners in the interval $I = [-a, a]$, for some a.
  - Let E_i be the event that the i-th runner is in the interval I.
  - Since cycle has length one and a runner with speed i performs i laps in a unit of time, $\Pr[E_i] = 2a$, for each i.
  - Therefore $\Pr[\exists \text{ (runner in the interval } I)] \leq \sum_{i=1}^{k} 2a = 2ka$.
  - For $a < \frac{1}{2k}$ we derive that

$$\Pr[\text{no runner is in the interval } I] \geq 1 - 2ka > 0.$$

- Recall I has length $2a$, which is $\approx \frac{1}{k}$, when $a \approx \frac{1}{2k}$.

Some Open Questions

- One of the main difficulties is that robots get to choose their speeds (up to a max value).
- What is the optimal solution to the boundary patrolling problem on the circle for agents with known maximal speeds?
  - solved for $k \leq 2$.
- Is a Proportional-Partition strategy optimal on the segment?
  - proved for $k \leq 2$.
- Is Uniform-Cyclic strategy optimal on uni-directional cycle?
  - proved for $k \leq 4$.
- Which strategies will work best for patroling problems in geometric scenarios (area patrolling) and in graphs?
- How about fragmented domains?

References

- J. Czyzowicz, L. Gasieniec. A. Kosowski, E. Kranakis. Boundary Patrolling by Mobile Agents with Distinct Maximal Speeds. In proceedings of 19th European Symposium on Algorithms. ESA 2011.
- A. Collins, J. Czyzowicz, L. Gasieniec, A. Kosowski, E. Kranakis, D. Krizanc, R. Martin, O. Morales Ponce. Optimal Patrolling of Fragmented Boundaries. SPAA 2013.