Optimization Problems in Infrastructure Security

Evangelos Kranakis
Carleton University
School of Computer Science
Ottawa, ON, Canada

Outline

- Infrastructure Security
  - SCADA
- Optimization Problems
  - Robot Patrolling.
  - Sensor Coverage and Interference.
  - Robot Evacuation.
  - Domain Protection and Blocking.
- Conclusion

What is Infrastructure Security?

- Someone may steal from it at night!
  - So, a night watchman position was created!
- How can the watchman work with no instructions?
  - So, a planning department was created!
- How we know the watchman is doing the tasks correctly?
  - So, a Quality Control Department was created!
- How are these people going to get paid?
  - So, ...
- How ...?
  - So, ...

Infrastructure Security

Infrastructure Security

- Infrastructure security is concerned with securing physical assets so as to
  - withstand, and
  - rapidly recover

  from potential threats that may affect critical resources located or enclosed within a given bounded region.
- This is a very "broad statement".

Diversity of Infrastructure Security

- The diversity of such systems makes potential threats difficult to grasp and the required rigorous security analysis almost impossible to pursue.

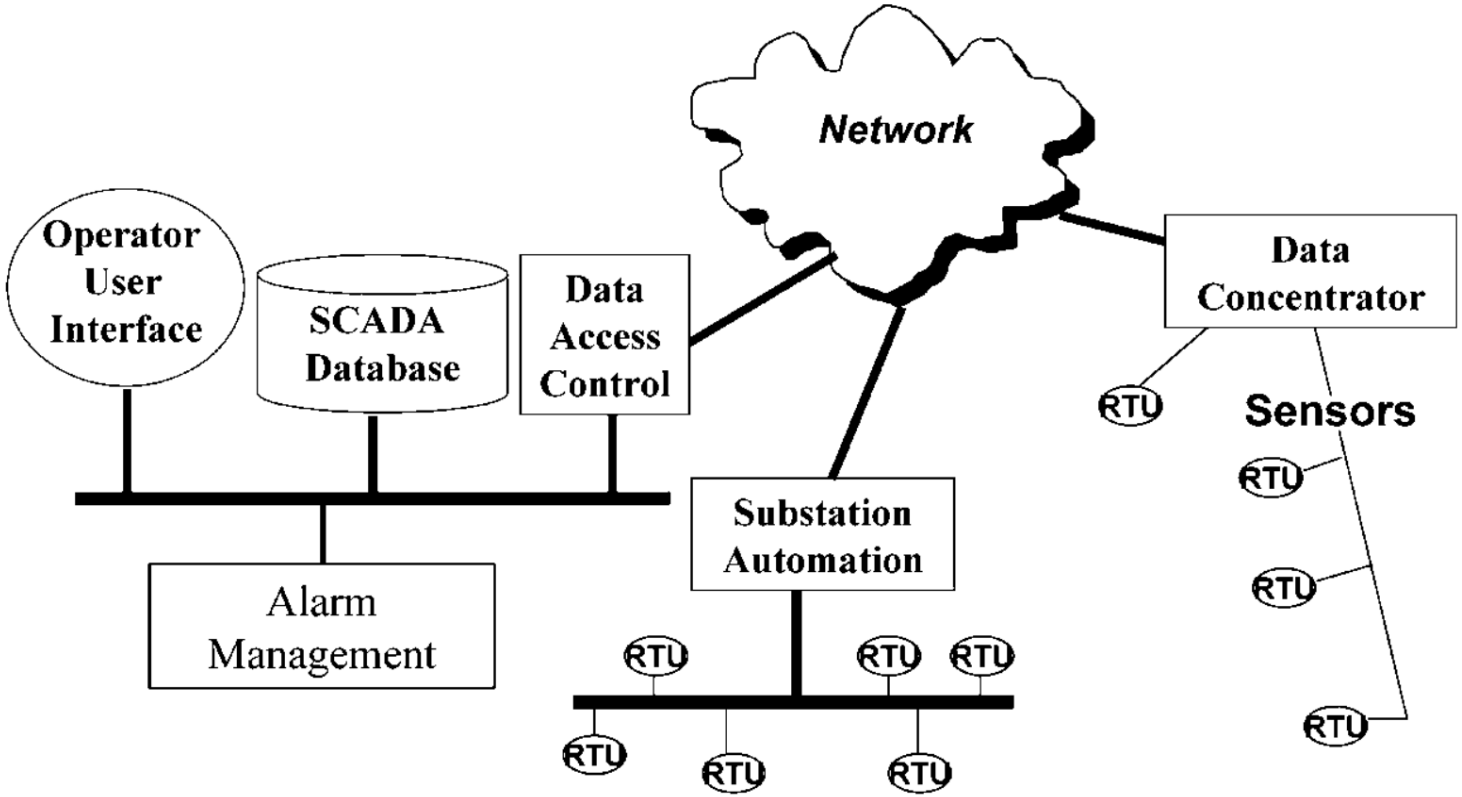
Infrastructure Sectors

- Buildings and roads,
- Border systems,
- Economic structures and materials,
- Energy and water supply systems,
- Internet and telecommunication systems.

What is SCADA?

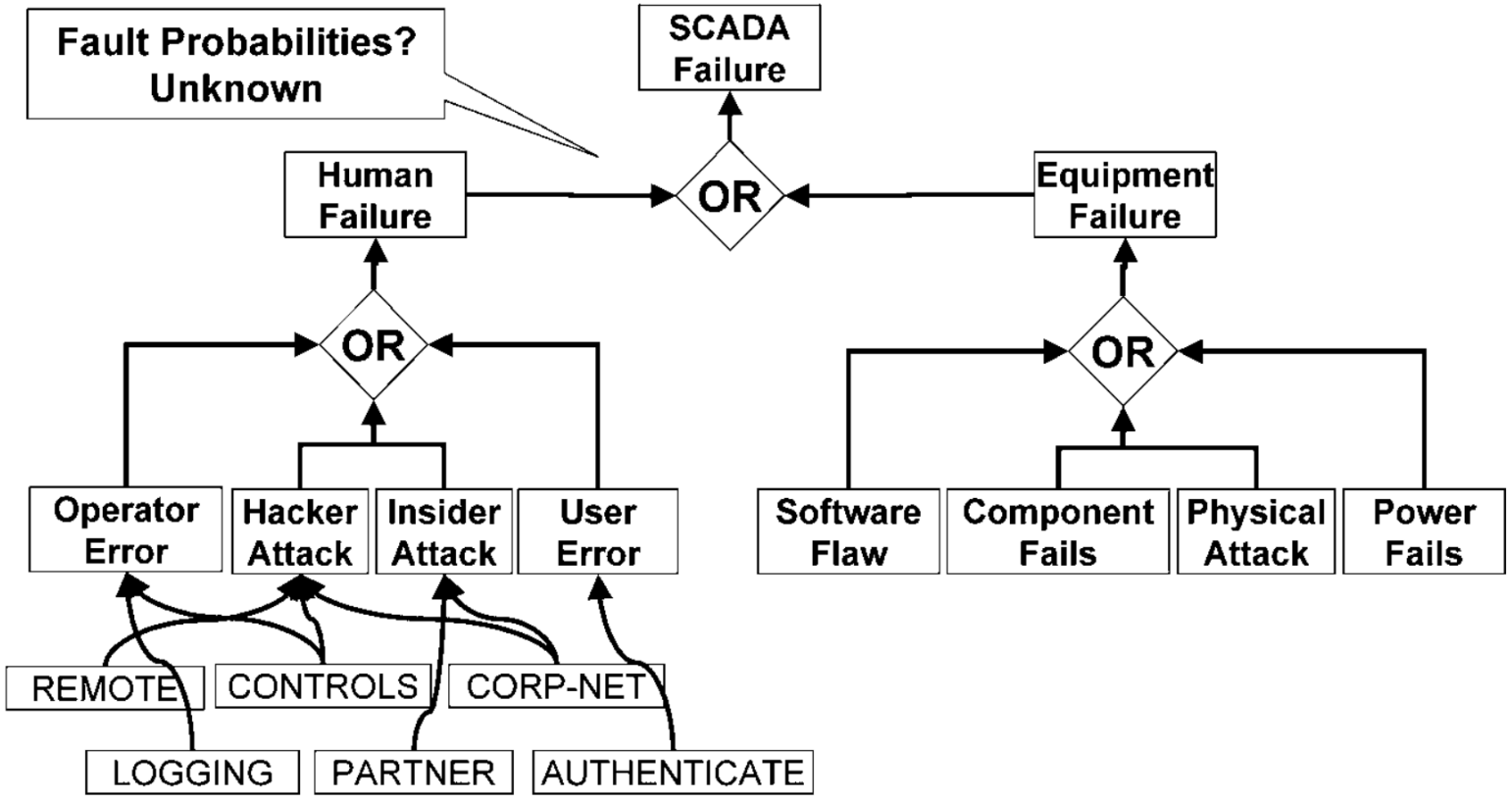
- Supervisory Control And Data Acquisition (SCADA).
  - Large scale computer based industrial control system for monitoring and controlling industrial facility based processes
- Includes various general buildings, transport systems, heating and ventilation systems, energy production and consumption.
- SCADA architectures:
  - originally primitive in design and conception
  - evolving systems; distributed and networked control augmented with sensor systems based on IoT.
- Network Infrastructure: system concepts and details of system components, control system for HCI by supervisory station(s), various types of communication methods.

SCADA[a]



[a]T. Lewis, Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation, 2006

General fault tree of possible vulnerabilities[a]



[a]T. Lewis, Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation, 2006

Smart Cities

- Intelligent Operations Center (IOC) for monitoring city services

  - water systems,
  - public safety,
  - transportation,
  - hospitals,
  - electricity grids, and
  - buildings, ...

Optimization Problems

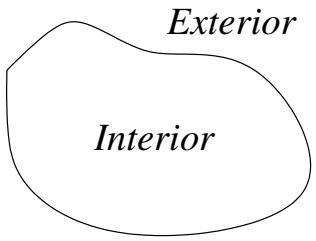
Optimization Problems

- Why optimization?
  - Optimizing the solution of a problem affects reaction time.
- Which optimization problems?
  - There are so many!
- Will discuss some which are relevant to security.
  - Robot Patrolling.
  - Sensor Coverage and Interference.
  - Robot Evacuation.
  - Domain Protection and Blocking.

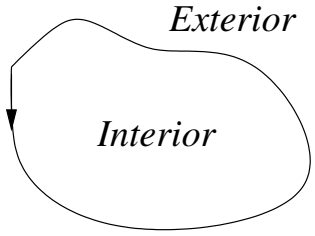
Patrolling

Motivation

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

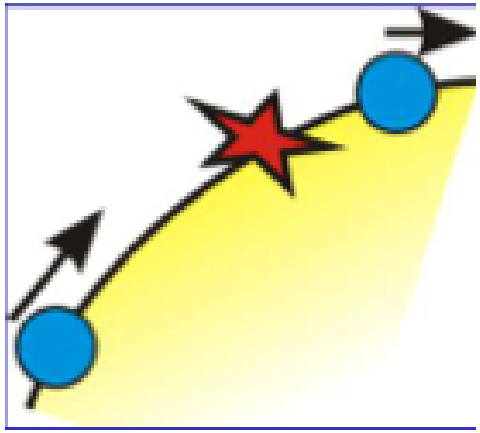


- **Patrolling problems in robotics**
  - Patrolling is defined as the perpetual process of walking around an area in order to protect or supervise it.



Problem

- 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

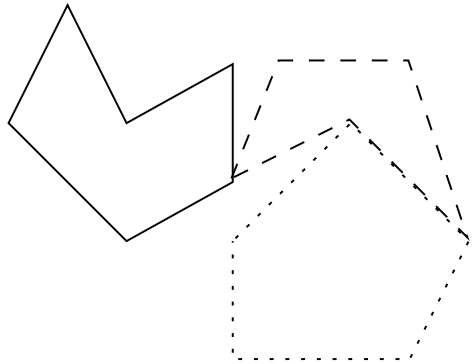
- Each agent i has its own predefined maximal speed v_i, for $1, 2, \ldots, k$.
- Agents are deployed on the boundary and programmed to move around the boundary, without exceeding their maximum speed.
- **Question:**
  for given speeds $\{v_1, v_2, \ldots, v_k\}$ and time τ, does there exist a deployment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

Efficiency

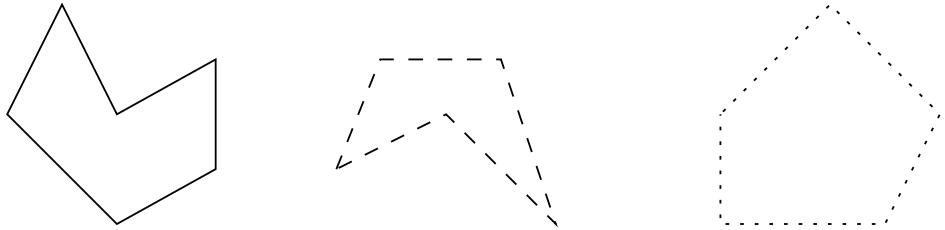
- How do you optimize the frequency of visits to the points of the environment?
- *Idleness (or refresh time:)* is the time elapsed since the last visit of the node.
  - Idleness can be average, worst-case, experimentally verified, etc,...
- In a way, given the input parameters you want to know what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

Patrolling Strategies

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

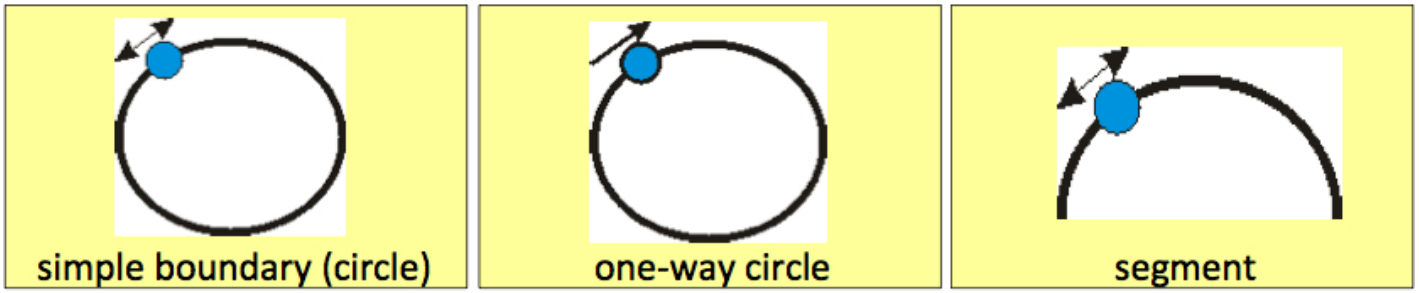


- A skeleton of the environment is defined over which patrolling is being conducted by the robots.

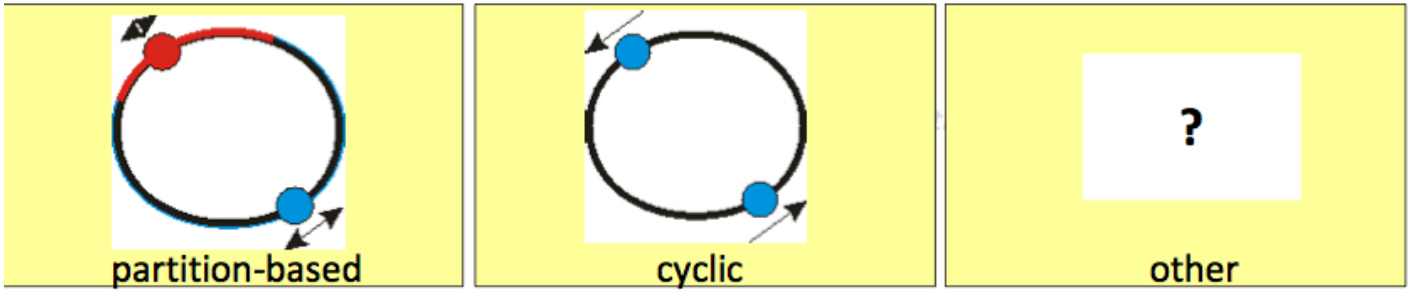


Goal

- 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 Algorithms

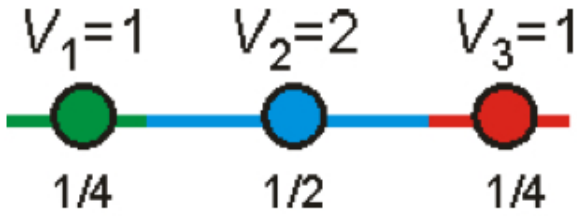
- The position of agent a_i at time $t \in [0, \infty)$ is described by the continuous function $a_i(t)$.
- Hence respecting the maximal speed v_i of agent a_i means that for each real value $t \geq 0$ and $\epsilon > 0$, s.t., $\epsilon v_i < 1/2$, the following condition is true

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

  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 $t+\epsilon$.
- 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 $i = 1, 2, \ldots, k$.

Proportional Partition

- **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.

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

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

Cyclic

- **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.

Conclusion & Further Results

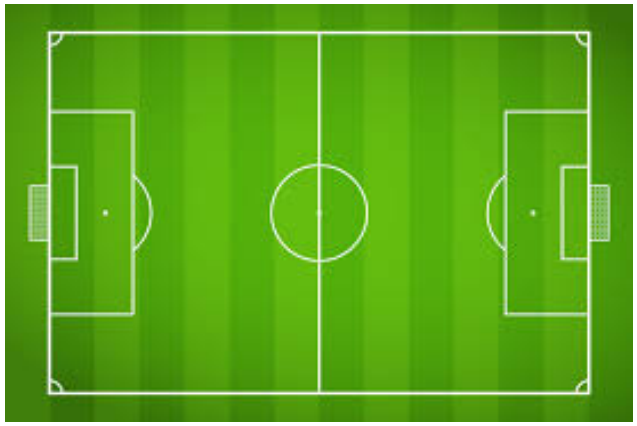
- Faulty robots.
- General Graph & Geometric Environments.
- Distributed vs Centralized Control.
- Many open and very difficult problems.

Coverage & Interference

Why Monitoring

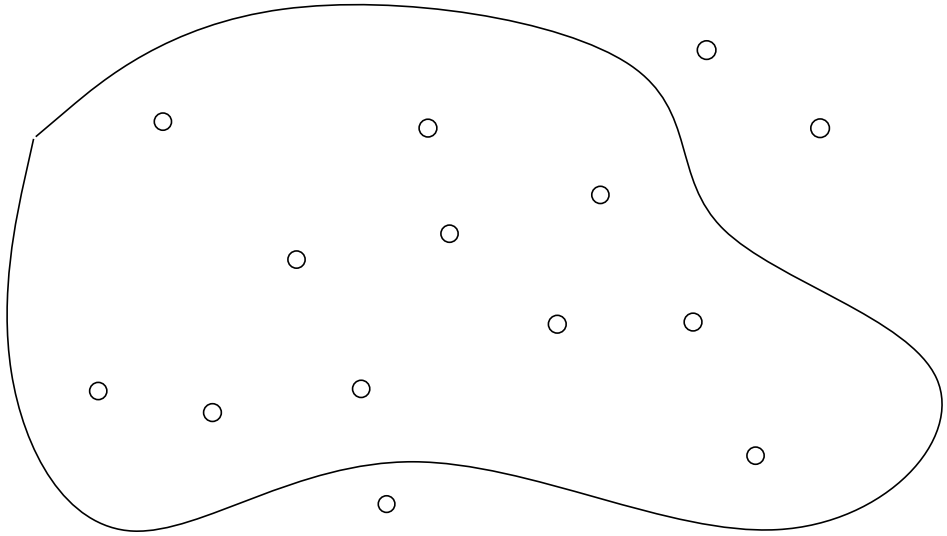
- Making Canadian "Ice Wine".

- The Beautiful game!



Sensor (Barrier) Coverage

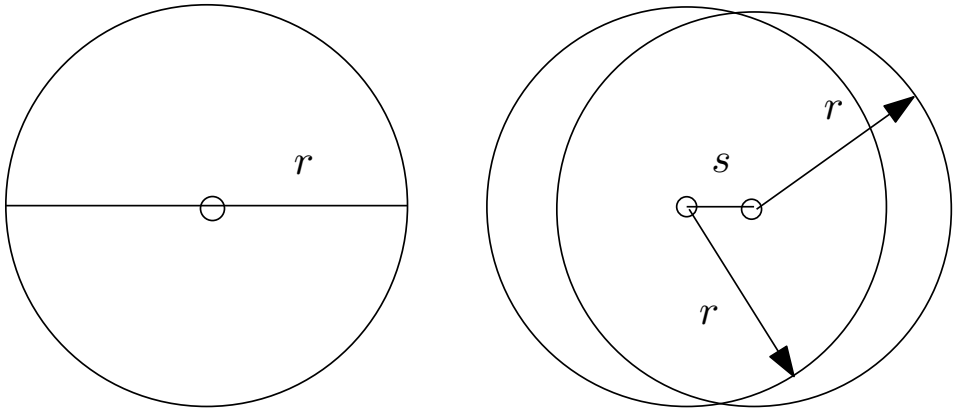
- A geometric domain and n sensors in(out)side the domain.



- Sensors may not cover the (barrier of the) domain!
- **Problem:** We want to move the sensors so as to cover the (barrier of the) domain in the sense that every point in the (barrier of the) domain is within the range of a sensor. Optimize the movement!

Sensor Coverage & Interference

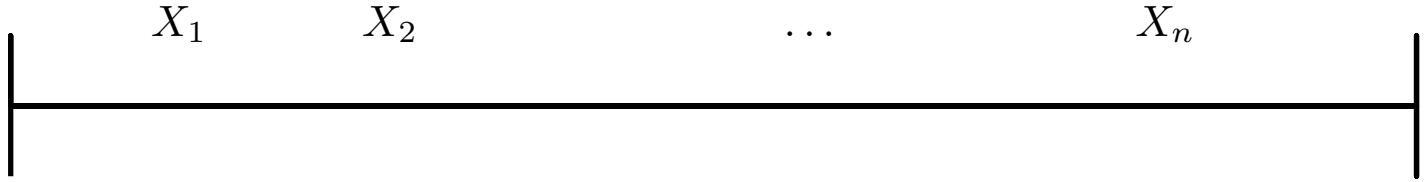
- Move the sensors from start positions so as to cover the domain.
- A critical value r specifies the coverage range.



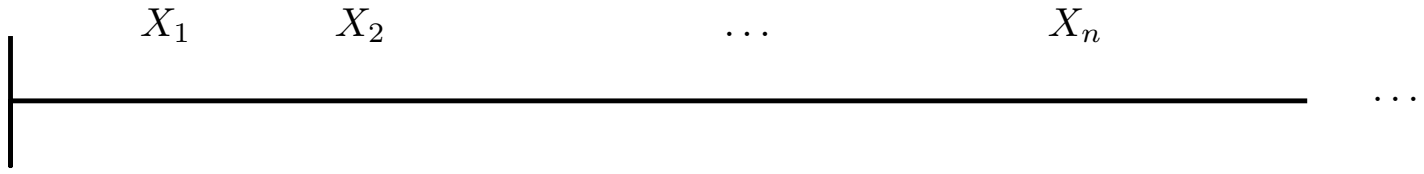
- A critical value, say $s > 0$ specifies that sensors be kept a distance of at least s apart.
  - *Signals interfere during communication if distance is* $< s$.

Random Model for Coverage & Interference

- Sensors are thrown randomly and independently with the uniform distribution in the unit interval.
- $X_1, X_2, \ldots, X_n$ represent sensor positions.
- Coverage Problem in the unit interval $[0, 1]$:



- Interference Problem in the half-line $[0, +\infty)$:



- X_i is the i-th arrival in a Poisson process.
- How much movement is needed to accomplish the task?

Coverage: Motivation (1/3)

- Throw n sensors of radius $r := \frac{1}{2n}$ at random in a unit interval.
- To ensure coverage of the interval they must be moved to anchors $a_i = \frac{i}{n} + \frac{1}{2n}$, for $i = 0, 1, \ldots, n-1$.
  - This is the worst-case total movement!
  - The cost is roughly $\sqrt{n}$.
  - Why?
- Do a simulation!

Coverage: Motivation (2/3)

- Keep increasing the sensor radius.
  - The bigger the radius the less the movement! Why?
- For n sensors of radius $\Theta(\frac{\ln n}{n})$, w.h.p. no sensor needs to move!
- Why?
  - The probability that no sensor drops inside a subinterval of length x is $(1-x)^n$.

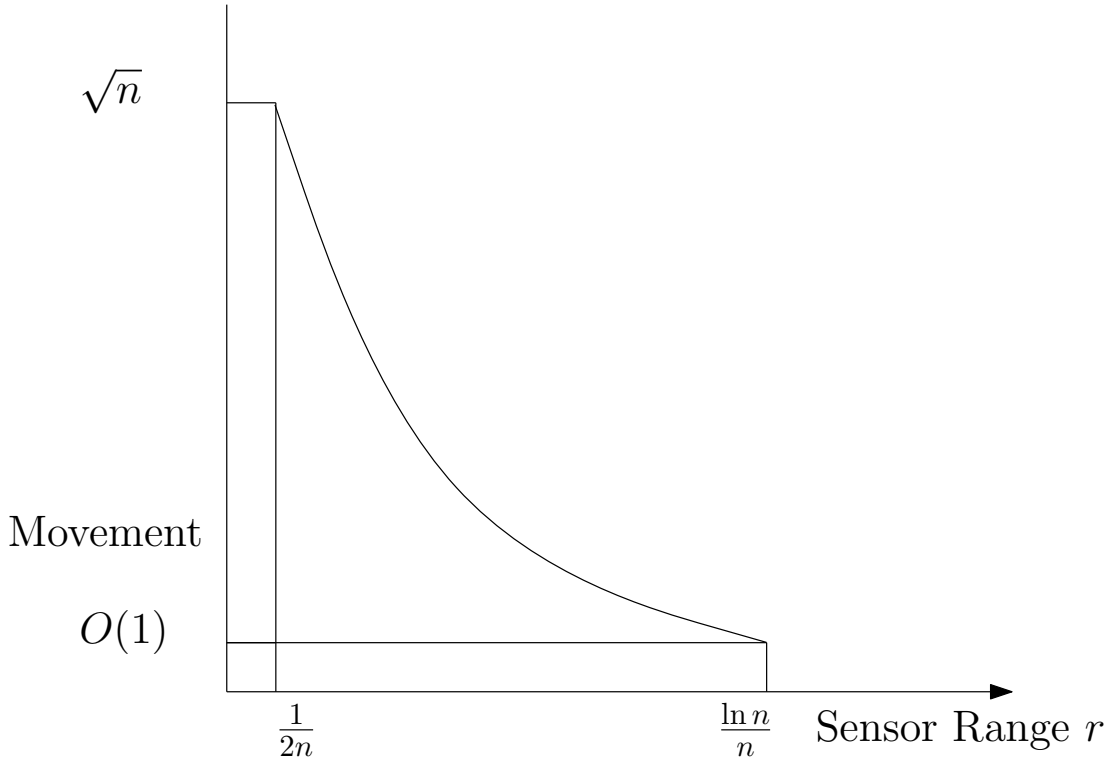
  - However,

$$(1-x)^n = \left(1 - \frac{xn}{n}\right)^n \approx e^{-xn} = \frac{1}{n^c},$$

for $x = \frac{c \ln n}{n}$, where $c > 0$.

Coverage: Prediction (3/3)

- Sensor movement as a function of the sensor range.



- The bigger the radius (range) the smaller the movement.

Interference: Motivation (1/2)

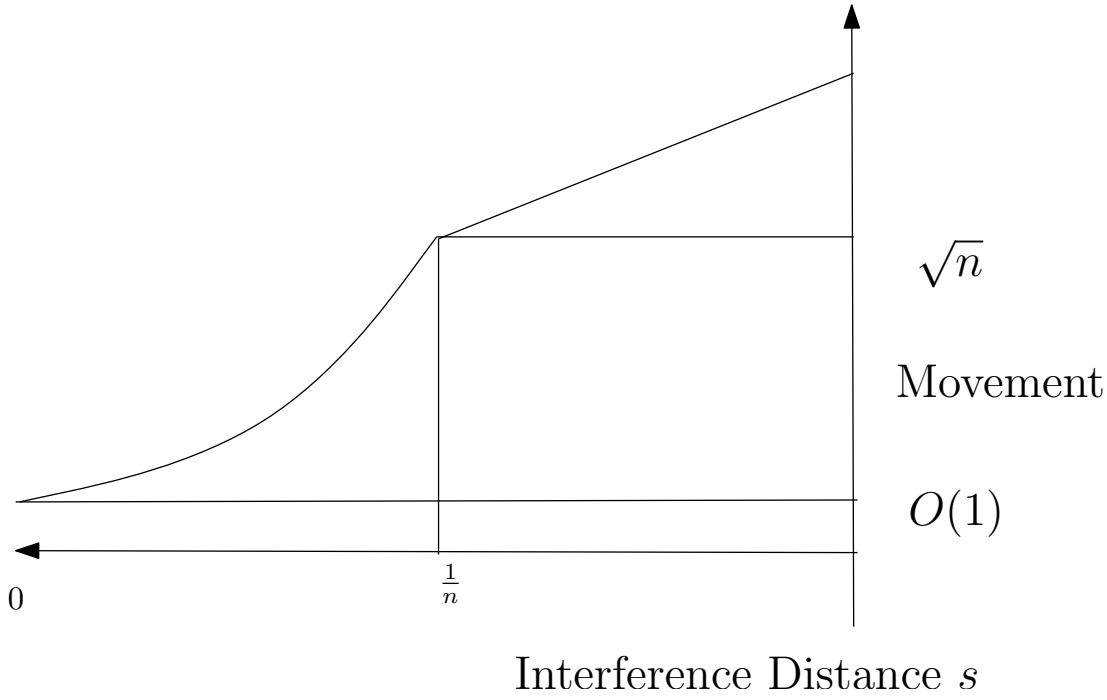
- Throw n sensors at random in a unit interval. We want to ensure no two sensors are at distance $< s$.
  - To ensure no two sensors are at distance $< \frac{1}{2n}$ they must all be placed to anchors $a_i = \frac{i}{n} + \frac{1}{2n}$, for $i = 0, 1, \ldots, n-1$. This is the worst-case total movement! Why?
- Keep decreasing the interference distance s.
  - The smaller the interference distance s the less the movement! Why?
- In general,

  Arrival Time of $i+1$st sensor − Arrival Time of ith sensor

  are the interarrival times of the Poisson process.

Interference: Prediction (2/2)

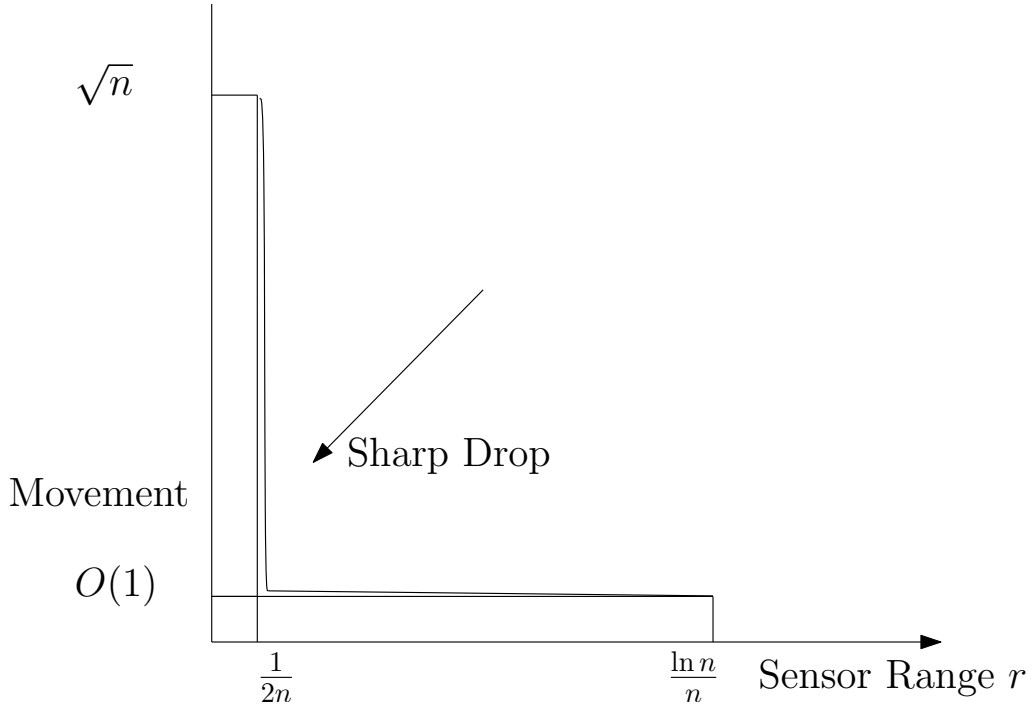
- Sensor movement as a function of the sensor distance.



- The smaller the interference distance the smaller the movement.

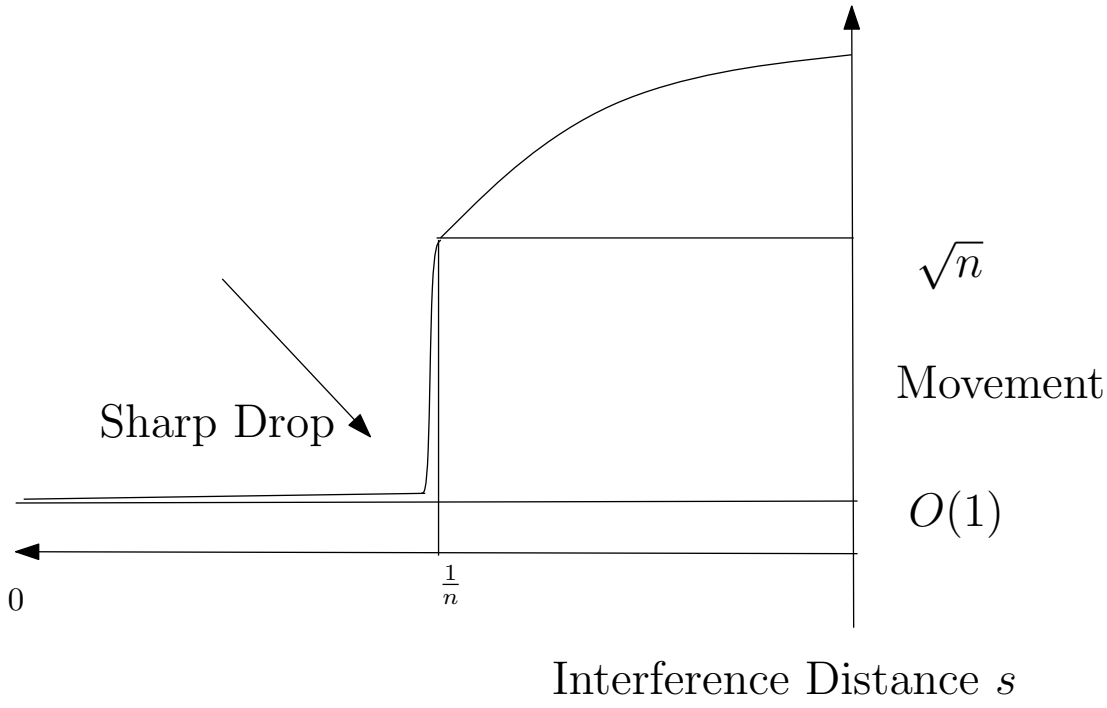
Critical Regime for Coverage

Sensor Range r	Total Displacement $E(r)$
$\frac{1}{2n}$	$\Theta(\sqrt{n})$
$\frac{1+\epsilon}{2n}$	$O(1)$



Critical Regime for Interference

- On a line there is **critical threshold** around $\frac{1}{n}$,
  1. for s below $\frac{1}{n} - \frac{1}{n^{3/2}}$, $E(s)$ is a constant $O(1)$,
  2. for $s \in \left[\frac{1}{n} - \frac{1}{n^{3/2}}, \frac{1}{n} + \frac{1}{n^{3/2}}\right]$, $E(s)$ is in $\Theta(\sqrt{n})$,
  3. for s above $\frac{1}{n} + \frac{1}{n^{3/2}}$, $E(s)$ is above $\Theta(\sqrt{n})$.
- Sensor movement as a function of the sensor distance.



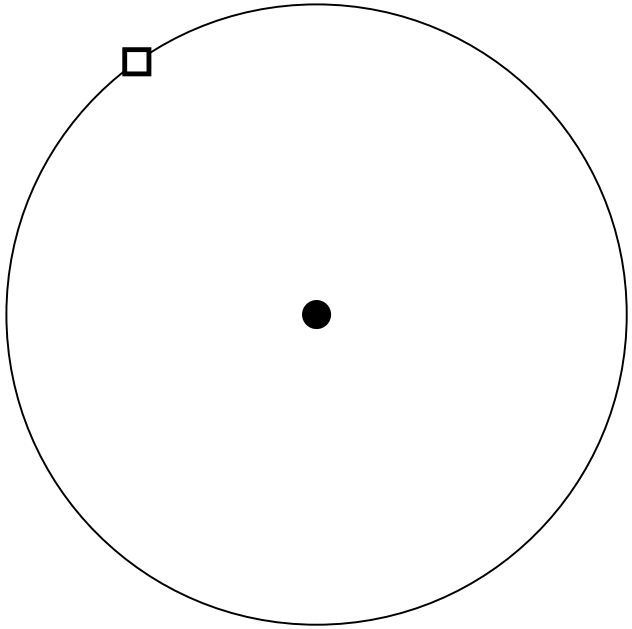
Conclusion and Further Results

- Several algorithms known in 1D
- Problem is harder in 2D
- Several metrics have been considered.
- Many interesting (difficult) questions for general domans.

Evacuation

Searching for an Exit (1/3)

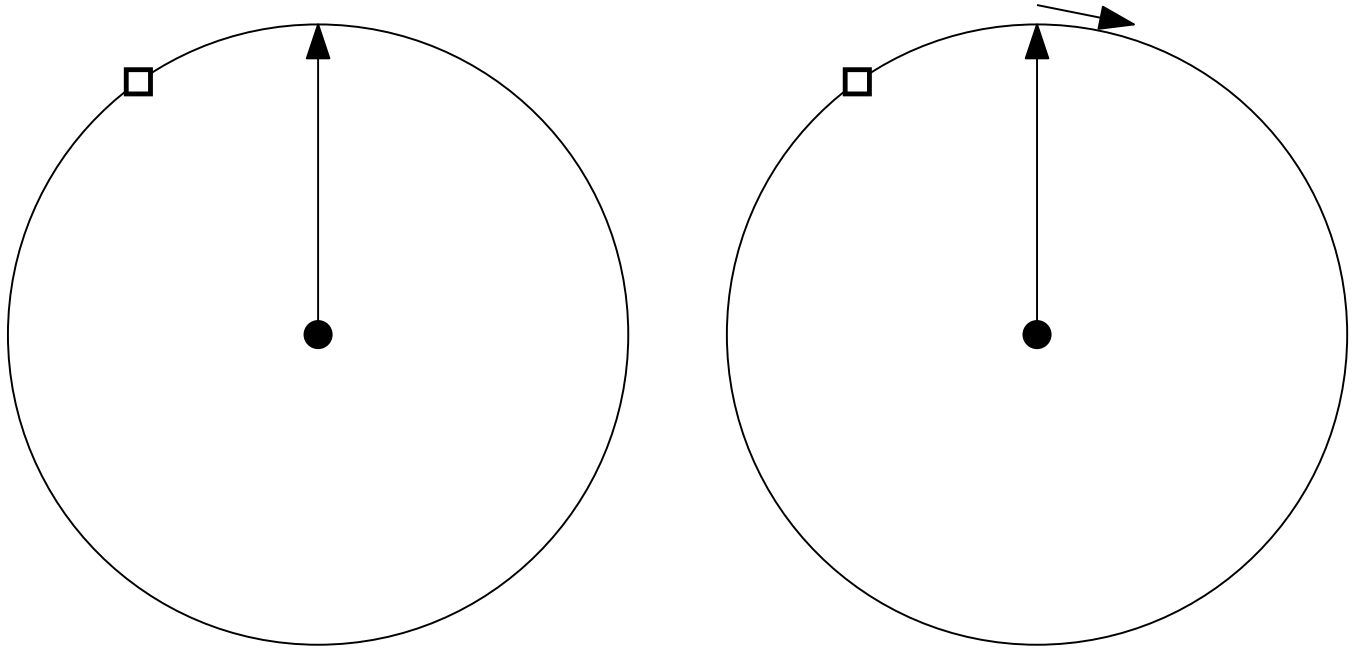
- You are located at some point.



- You are told an exit is at distance 1 from you.
- Your max speed is 1.
- What is the best (time optimal) algorithm to find the exit?

Searching for an Exit (2/3)

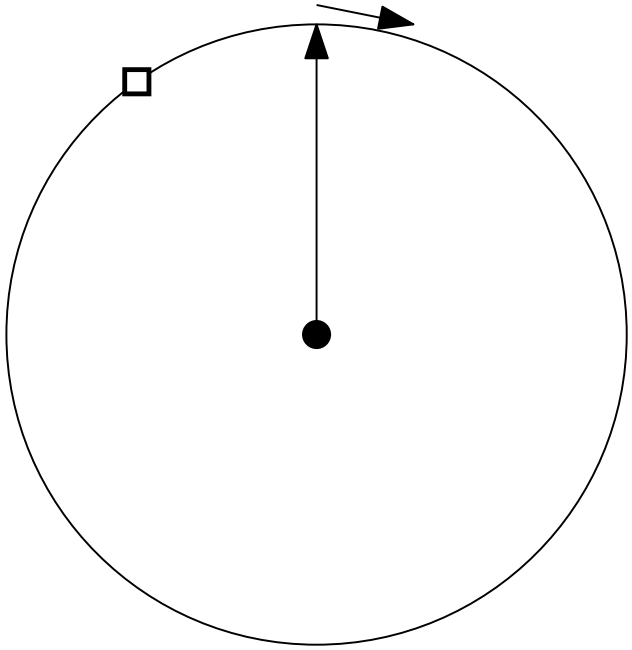
- Go for distance 1 (to the perimeter).



- On the perimeter choose a direction (CW or CCW).
- How long does it take you (in the worst case) to find the exit?

Searching for an Exit (3/3)

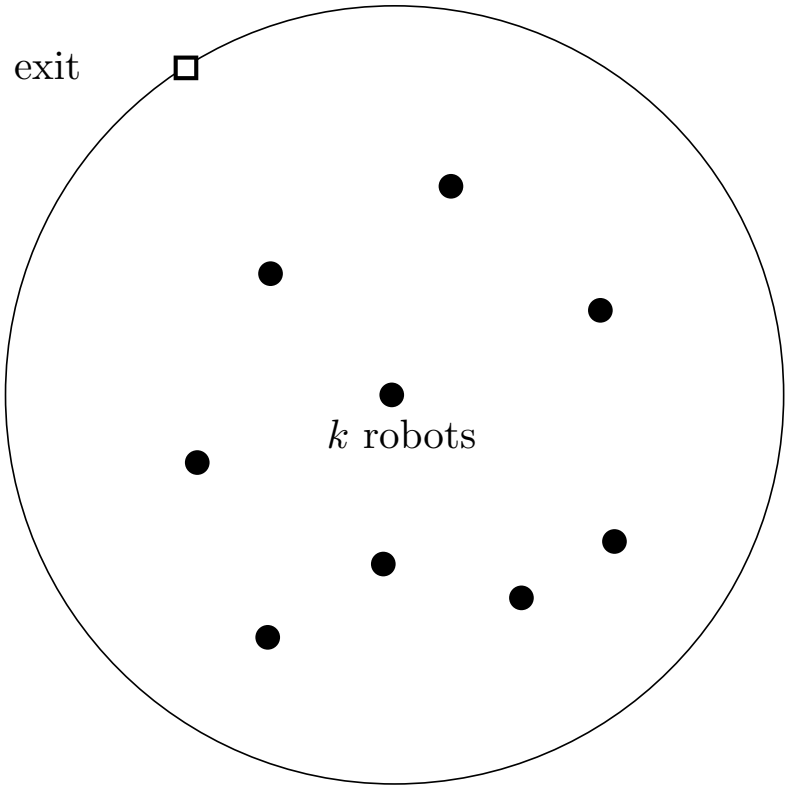
- In the worst case, this algorithm takes time $1 + 2\pi \sim 7.28$.



- Can you do better than $1 + 2\pi$?

Evacuation from a Circle

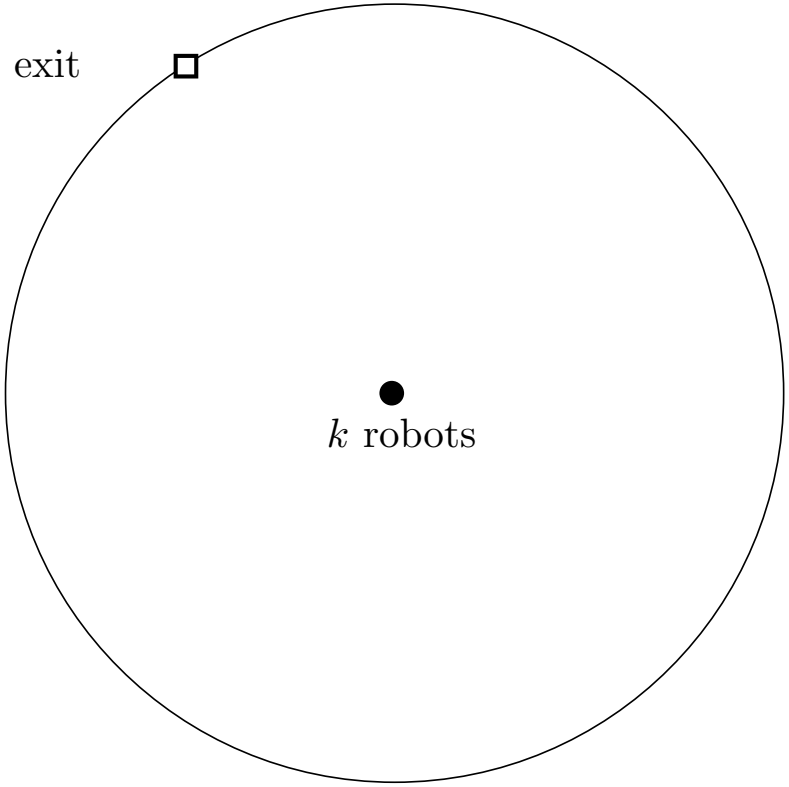
- $k \geq 2$ robots start from anywhere inside a disk.



- Exit is located on the perimeter.

Lets simplify the geometry!

- k robots start from the center of a disk.



- Exit is located on the perimeter.

Evacuation Problem

- Consider k mobile robots inside a circular disk of unit radius.
- The robots are required to evacuate the disk through an unknown *exit*: a point situated on its boundary.
- We assume all robots having the same (unit) maximal speed and starting at the centre of the disk.
- The robots may communicate in order to inform themselves about the presence (and its position) or the absence of an exit.
- The goal is for all the robots to evacuate through the exit in minimum time.

Communication Models

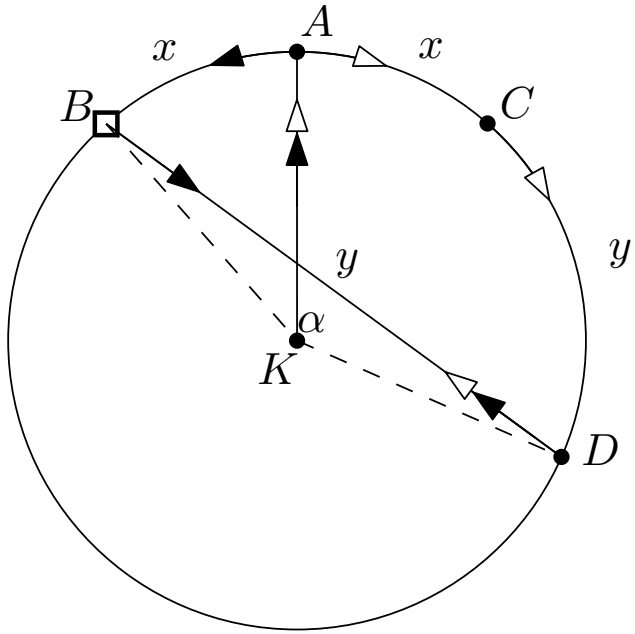
- How do the robots communicate?
- We consider two models of communication between the robots:
  - *face-to-face* (or *local*) *communication* model: robots exchange information only when simultaneously located at the same point, and
  - *wireless communication* model: robots can communicate with each other at any time.

Evacuation Time

- Robots do not necessarily evacuate at the same time.
  - Robots can try to find the exit on their own!
  - As soon as a robot finds the exit it tries to inform the rest!
- Measuring the complexity of an algorithm.
  - Minimum time required so that all robots evacuate.

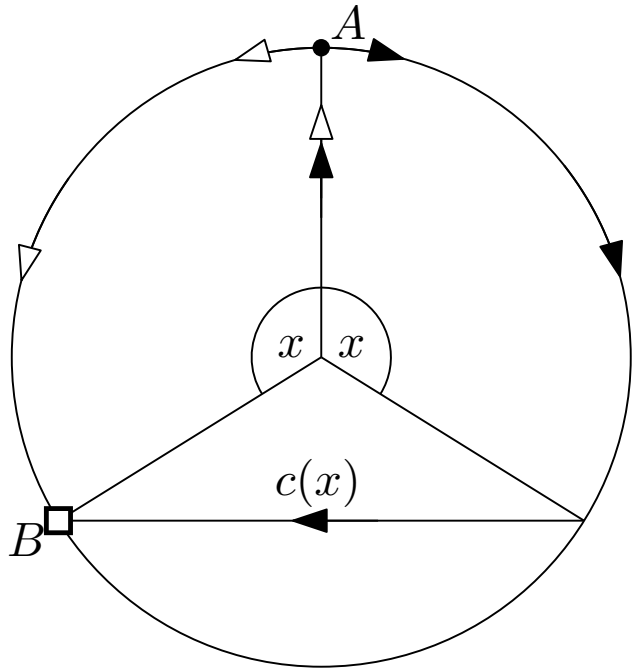
Face-to-Face Model: Evacuation Algorithm for 2 robots

- **Theorem 1** *There is an algorithm for evacuating the robots from an unknown exit located on the perimeter of the disk which takes time* $1 + \alpha/2 + 3\sin(\alpha/2)$ *where the angle* α *satisfies the equation* $\cos(\alpha/2) = -1/3$. *It follows that the evacuation algorithm takes time* ~ 5.74.
- Evacuation Algorithm



Wireless Communication Model

- **Theorem 2** *There is an algorithm for evacuating two robots from an unknown exit located on the perimeter of the disk which takes time at most* $1 + \frac{2\pi}{3} + \sqrt{3} \approx 4.826$.
- Evacuation Algorithm



What Else is Known about Evacuation

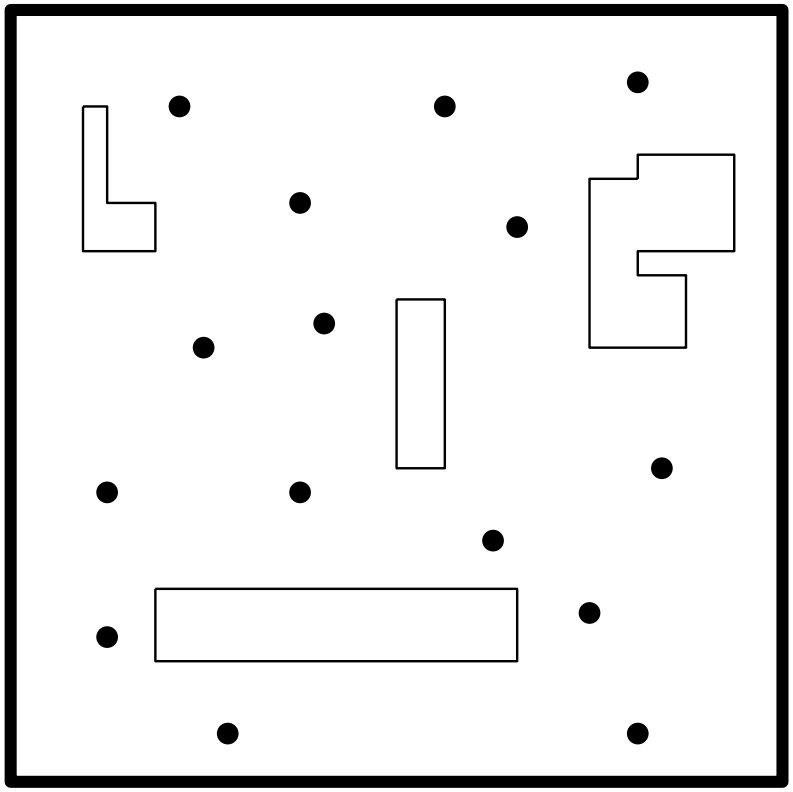
Table 1. Upper and Lower bounds for $k \geq 2$ robots

Model	*Bound*	$k = 2$	$k = 3$	$k \geq 4$
Non-wireless	Upper	~5.74 (Th 1)	~5.09 (Th 8)	$3 + \frac{2\pi}{k} < 4.58$ (Th 8)
	Lower	~5.199 (Th 2)	~4.519 (Th 5)	$3 + \frac{2\pi}{k} - O(k^{-2})$ (Th 9)
Wireless	Upper	~4.83 (Th 3)	~4.22 (Th 6)	$3 + \frac{\pi}{k} + O(k^{-4/3})$ (Th 10)
	Lower	~4.83 (Th 4)	~4.159 (Th 7)	$3 + \frac{\pi}{k} > 3.785$ (Th 11)

Blocking

Rectangular Domain

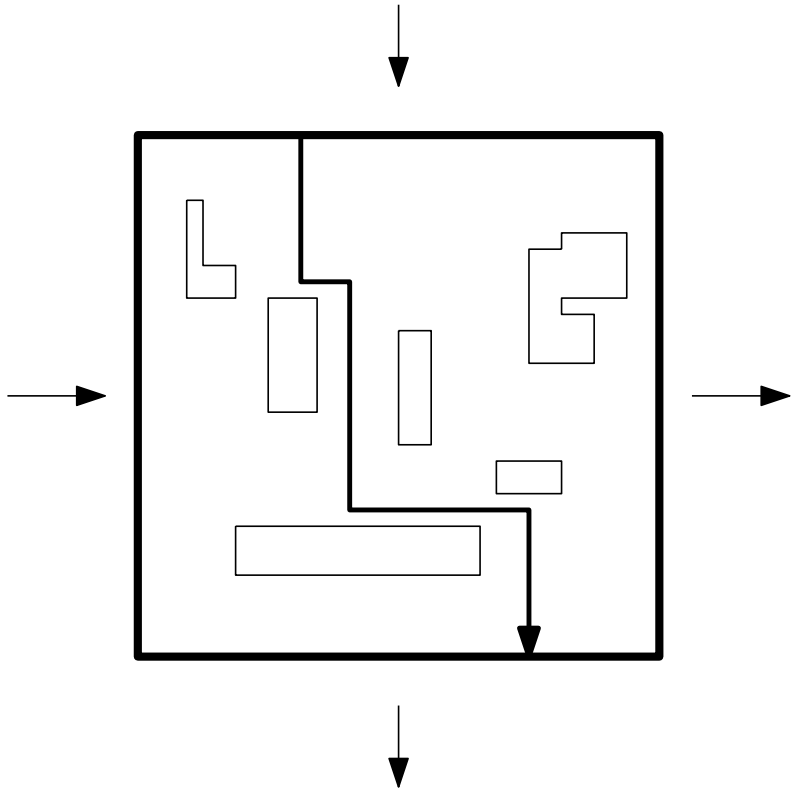
- Consider a rectangular grid domain



including buildings and sensors in specific locations.

Penetration

- Intruders attempt to penetrate walking in a rectangular domain.



- Sensors are placed at nodes of a regular spaced grid laid out over the rectangle.

Protection & Blocking

- An intruder that steps within the sensing range of a sensor will be detected.
- It is desired that we prevent potential attacks in either one dimension or two dimensions.
  - A one-dimensional attack succeeds when an intruder enters from the top (North) side and exits out the bottom (South) side of the domain without being detected.
  - Preventing attacks in two dimensions requires that we simultaneously prevent the intruder from either entering North and exiting South or entering East (left side) and exiting West (right side) undetected.

Fault Tolerant

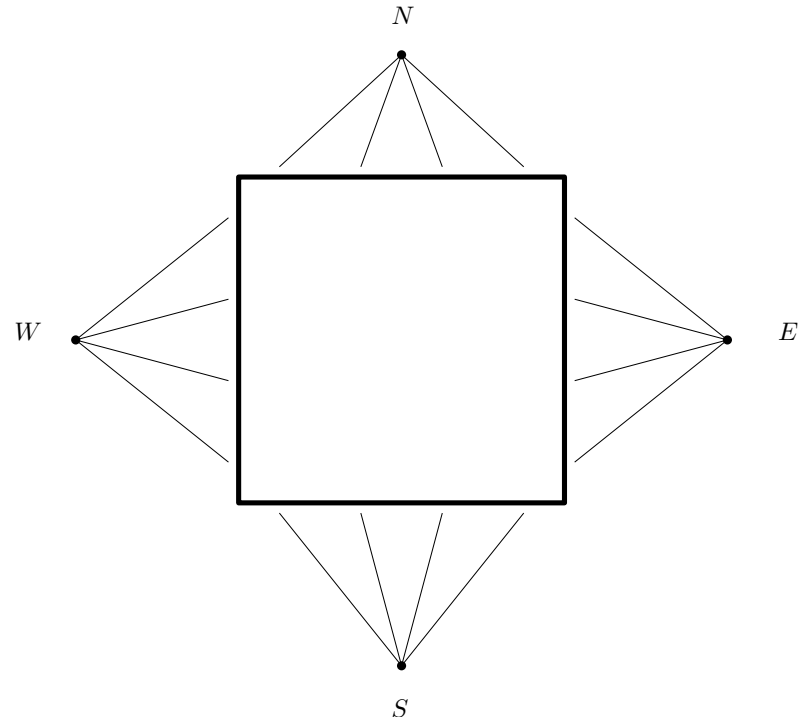
- Initially, all of the sensors are working properly and the domain is fully protected, i.e., attacks will be detected, in both dimensions
- Over time, the sensors may fail and we are left with a subset of working sensors. Under these conditions we wish to
  1. determine if one or two-dimensional attack detection still persists and
  2. if not, restore protection by adding the least number of sensors required to ensure detection in either one or two dimensions.

Blocking: k-Fault Tolerant

- Main Problem:
  1. Decide if a subset of the sensors provides protection with up to k faults and
  2. if not, find the minimum number of grid points to add sensors to in order to achieve k fault-tolerance
  3. or, if not, find the minimum number of grid points to add sensors to in order to achieve k fault-tolerance that minimizes the total or max movement.
- Also interesting for optimally restoring k-fault tolerant protection in one dimension and for restoring protection in two dimensions (optimally for $k = 0$ and approximately otherwise).

Main Idea: Steiner Points

- Add four virtual points: N, S, E, W



- Sensors (Steiner Points) are placed at nodes of a regular spaced grid laid out over the rectangle.

Results

- We show the following for an $m \times n$ grid:
  1. There exist $O(mn)$ time algorithms for solving the one- and two-dimensional k-protection decision problems.
  2. There exists a $O(kmn \log(mn))$ time algorithm for solving the one-dimensional k-protection placement problem.
  3. There exists a $O(m^2n^2)$ time algorithm for solving the two-dimensional protection placement problem.
  4. There exists a $O(kmn \log(mn))$ time 2-approximation algorithm for solving the two-dimensional k-protection placement problem.
- In all of the above we assume $k < \min\{m, n\}$ as we shall see that the problems can not be solved otherwise.

Extensions

- More general versions of these problems could be studied, including protecting:
  1. domains containing impassable regions,
  2. non-rectangular domains, and
  3. against more general attacks than just North-South or East-West.

Conclusions

- Many optimization problems are tied to infrastructure security.
- Their study, analysis and appropriate implementation could be vital to the proper functioning of infrastrucure in our society.