Multi-Agent Adversarial Games in Partially Observable Euclidean Space

Eric Raboin and Dana Nau
Dept of Computer Science, University of Maryland

Ugur Kuter
Smart Information Flow Technologies (SIFT)

Satyandra K. Gupta and Petr Svec
Dept of Mechanical Engineering, University of Maryland

Two Related Problem Domains

- Two problems that involve adversarial groups of agents maneuvering in Euclidean space
  - » Neither of them is *exactly* moving-target defense
  - » But both of them are closely related to it

1. Tracking and evasion

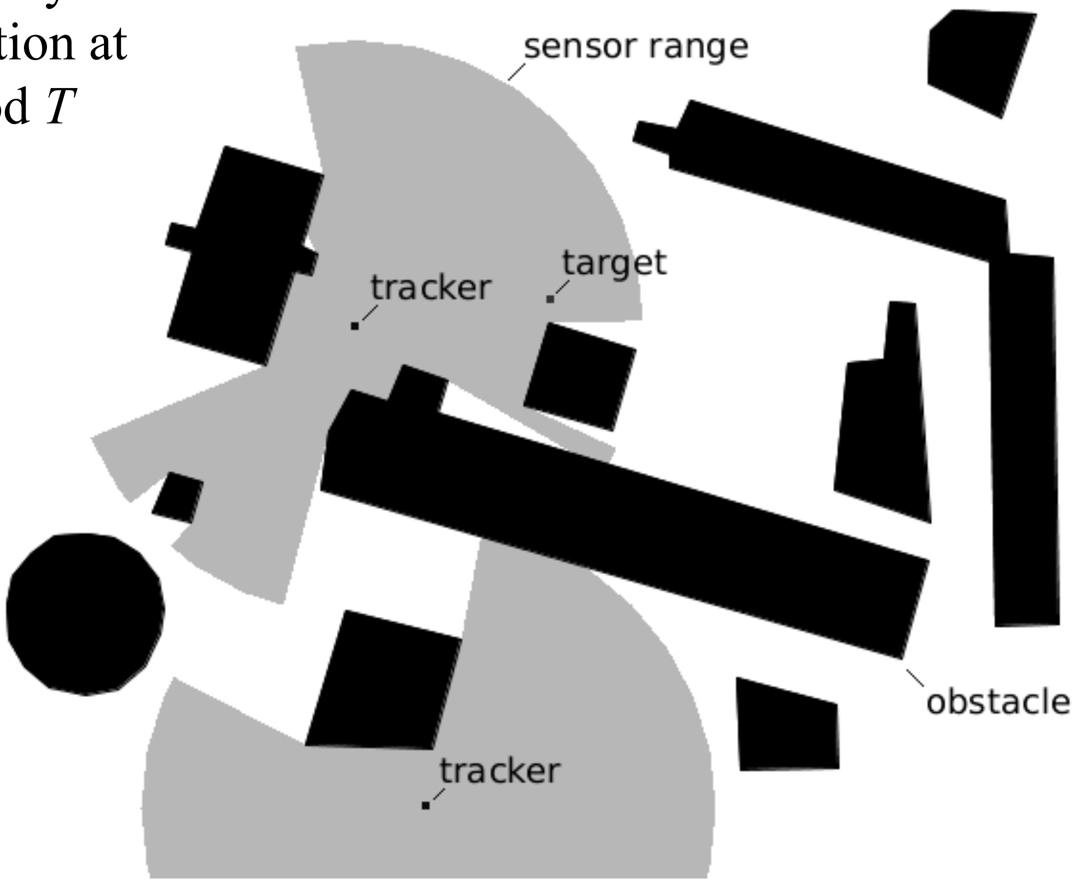
   Team of *tracker* agents want to track the location
   of an intelligent *target* agent
   that wants to evade their surveillance

2. Naval asset protection

   Team of patrolling/blocking agents want to protect
   assets against intelligent intruders

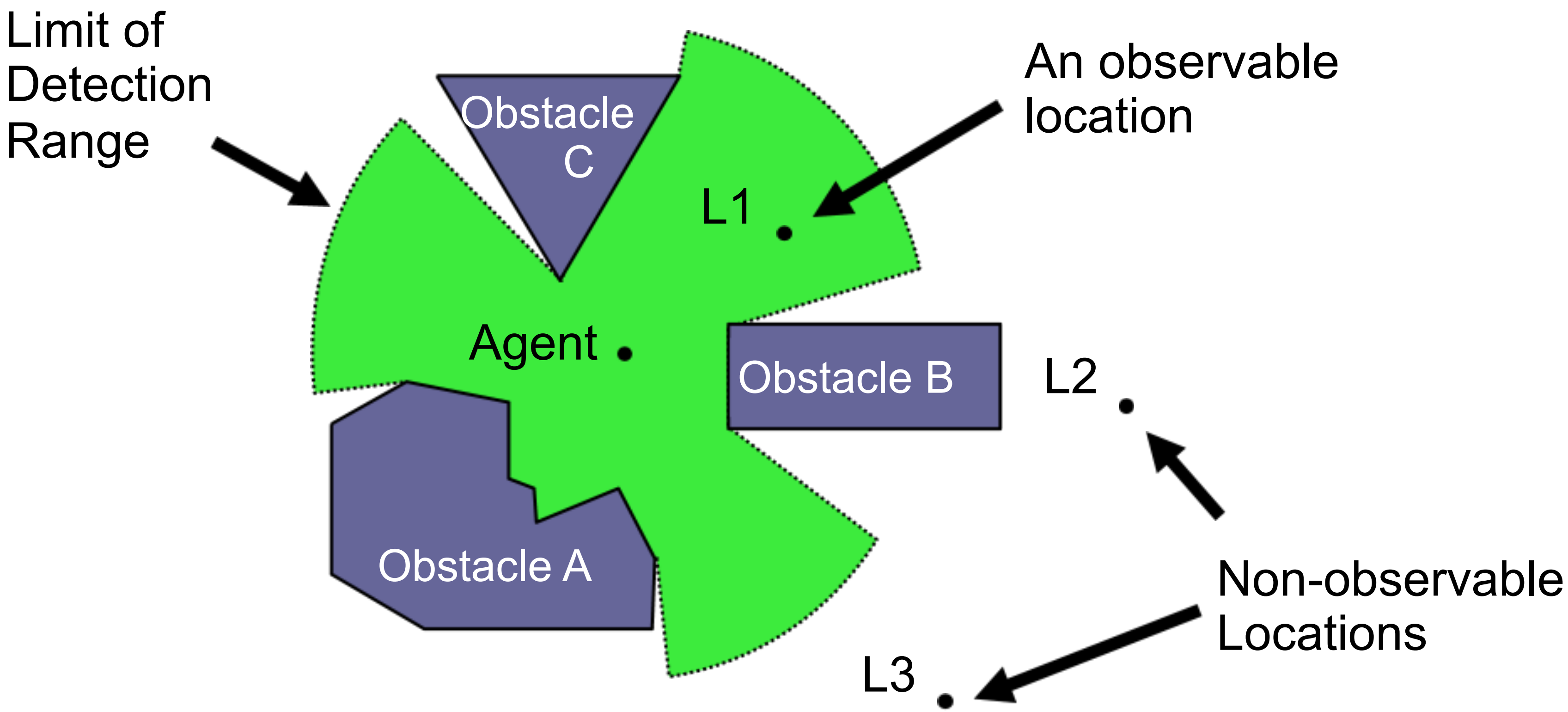
Part 1. Tracking and Evasion

- Team of cooperating tracker agents
  - Want to minimize uncertainty about a target agent's location at the end of some time period *T*
- Target agent
  - Wants the opposite
- Continuous Euclidean space
  - Arbitrarily shaped polygonal obstacles
- Partially observable
  - …



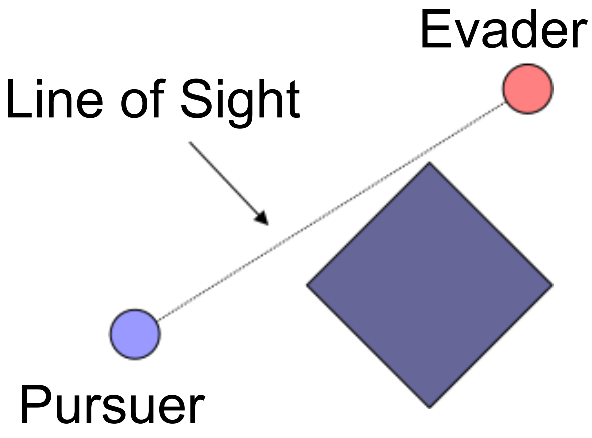
Partial Observability

- Agents' observation capabilities are limited by
  - sensor range
  - obstacles

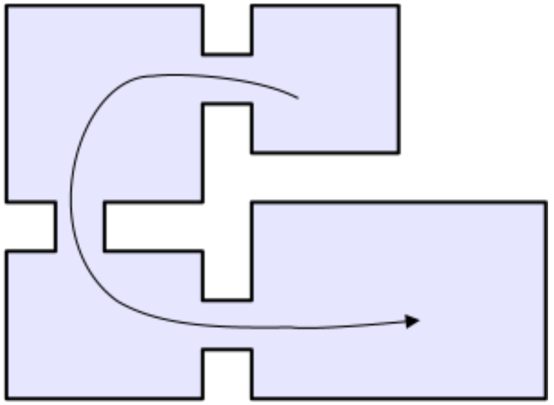


Related Work

- Maintain line-of-sight on the target
  - » Game ends when visibility is lost
  - » Perfect-information game
  - » Differential game theory



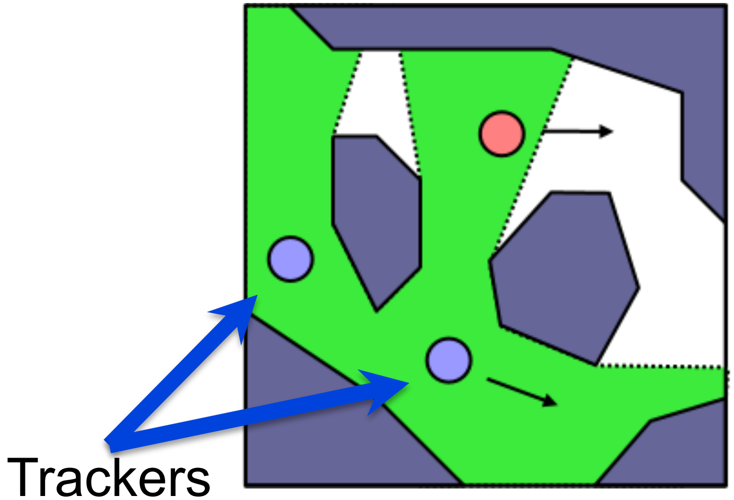
- Find an unseen intruder (hider-seeker)
  - » Game ends when target is discovered
  - » Combinatorial search to generate patrol strategies
  - » Computed offline
  - » No pursuit strategy



Need for New Techniques

- Can we just combine existing techniques?
  - Not effectively, for two reasons

- Our objective is different
  - Minimize uncertainty about the target's location, even when the target isn't visible
  - Sometimes the most effective strategy is to to choose actions that sacrifice visibility,
    - in order to reduce uncertainty later on

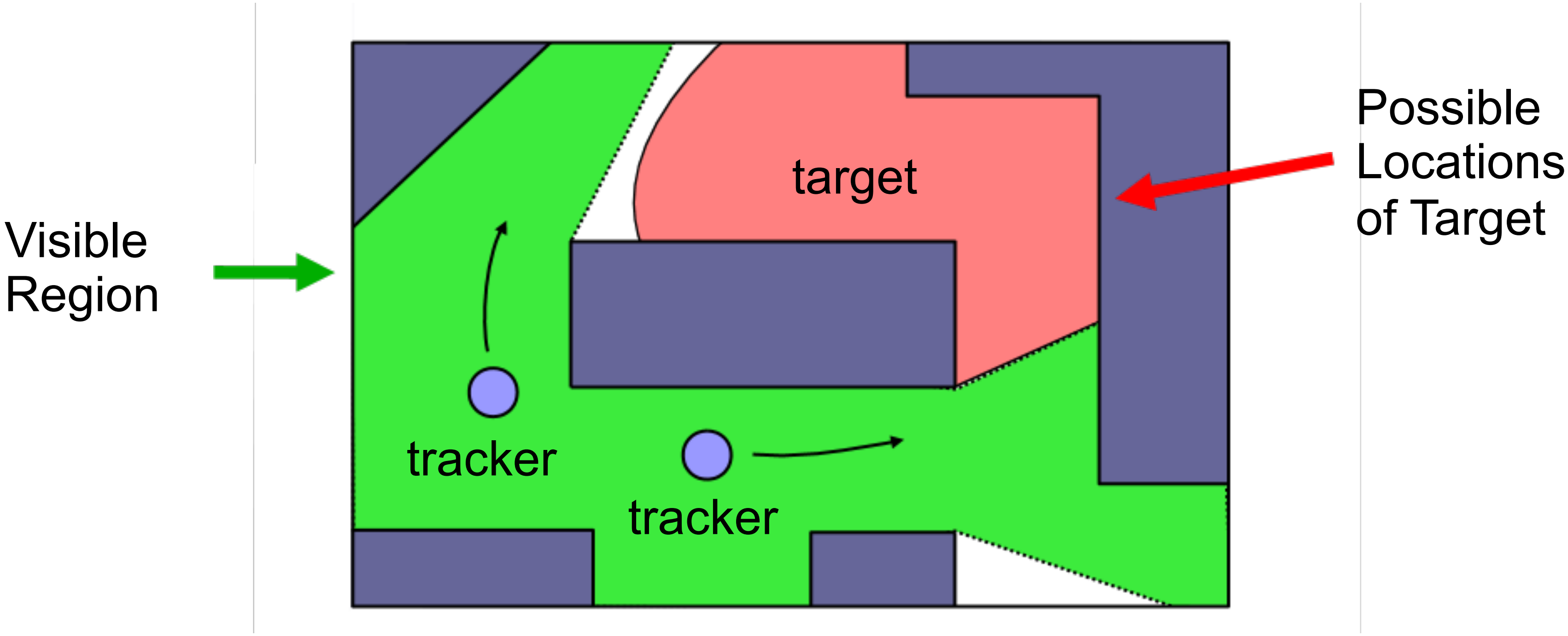


Move out of sight of the target, in order to regain visibility later

- Need to generate strategies quickly, in response to target's movement
  - Rules out many techniques that are based on deep combinatorial search

Minimizing Uncertainty

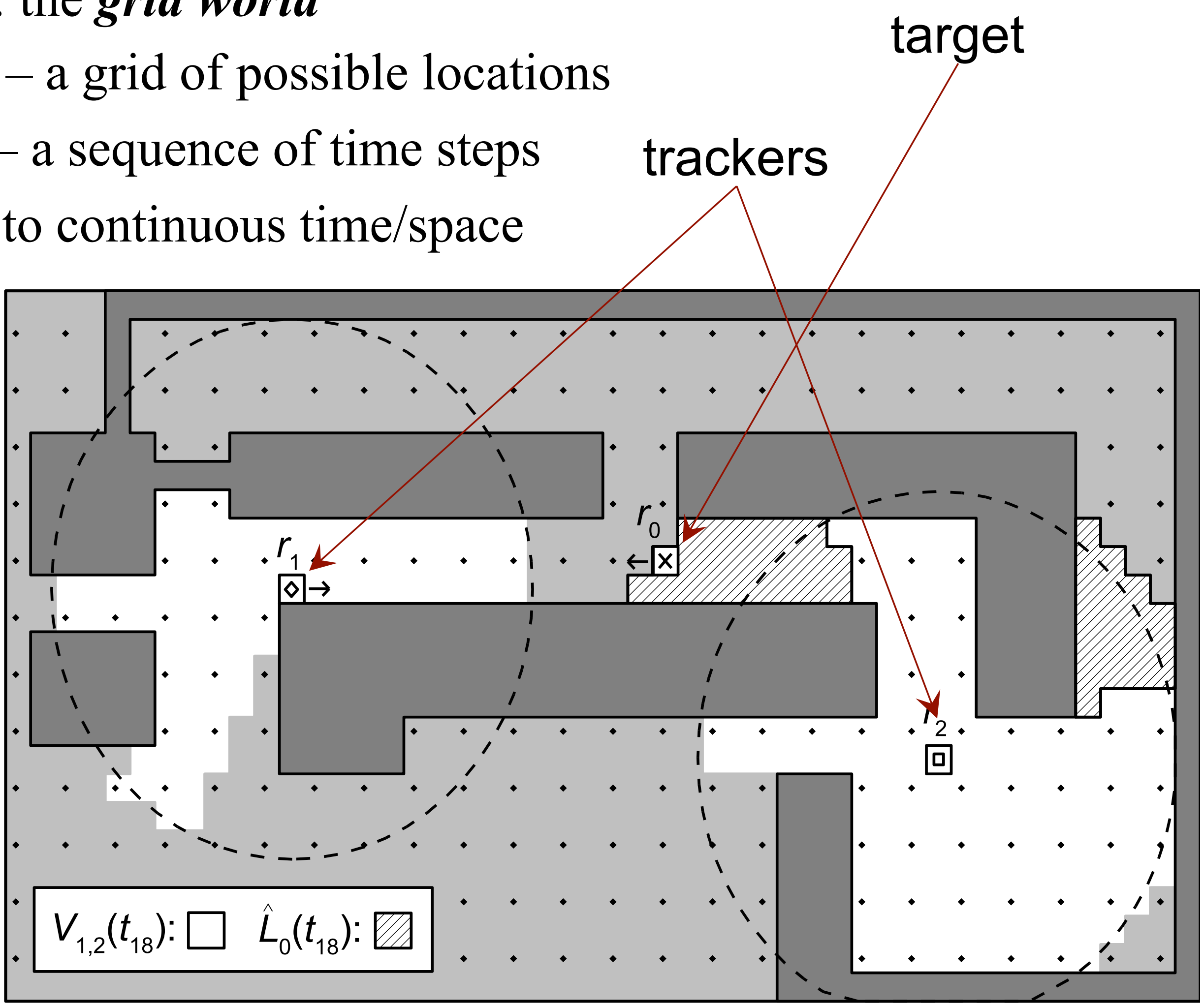
- Loss of visibility of the target may be inevitable
- But with the right strategy, the trackers may be able to
  - Guarantee that the target is within some small region
  - Recover visibility later on



Part 1(a). Earlier Work

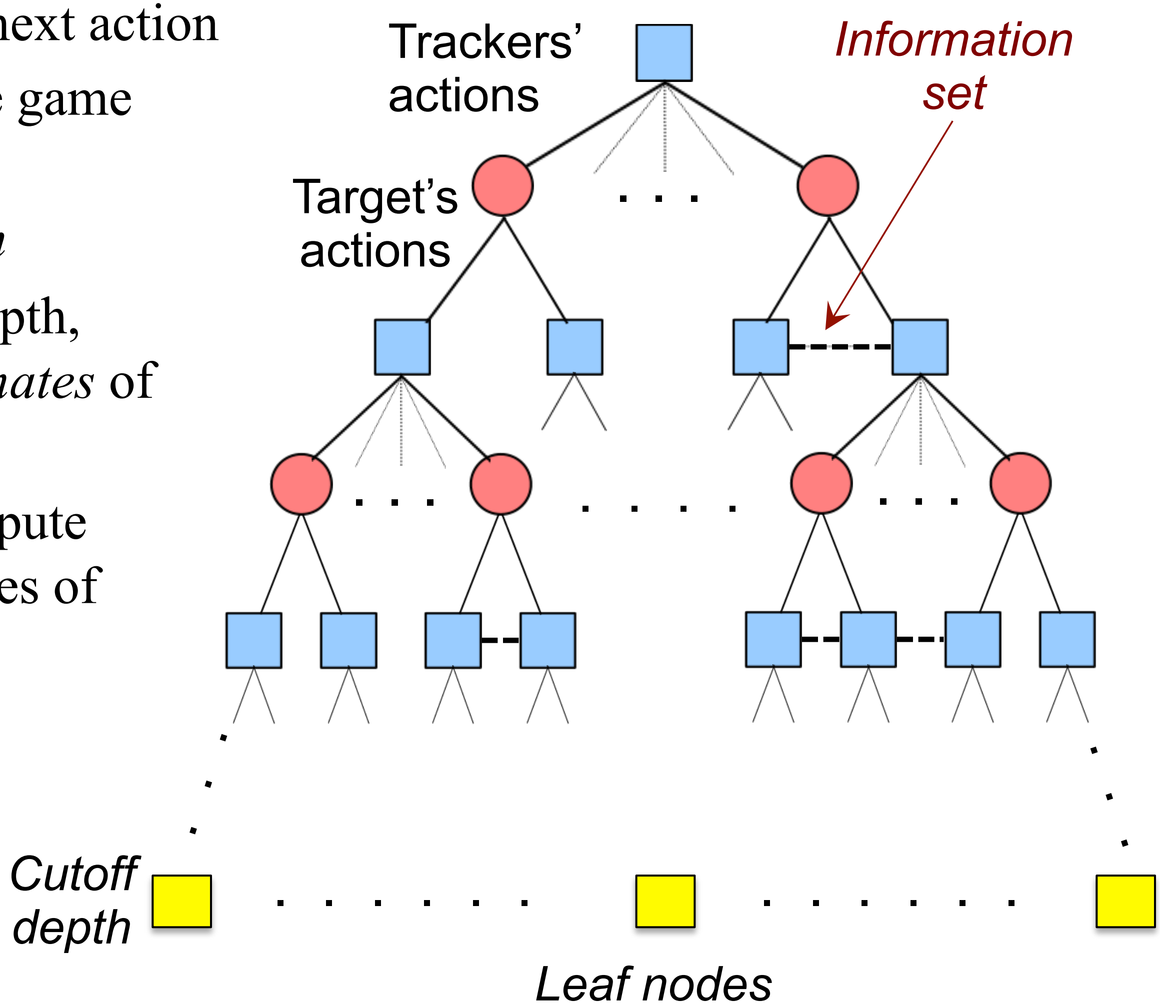
E. Raboin, D. Nau, U. Kuter, S. K. Gupta, and P. Švec. Strategy generation in multi-agent imperfect-information pursuit games. In *Ninth Internat. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, 2010. pdf.

- Imperfect-information zero-sum extensive-form game
- Simplifying assumption: the ***grid world***
  - Space is discrete – a grid of possible locations
  - Time is discrete – a sequence of time steps
    - » Later I'll generalize to continuous time/space
- Agent's possible actions at each time step:
  - » Move to any adjacent grid point that isn't occupied by an obstacle



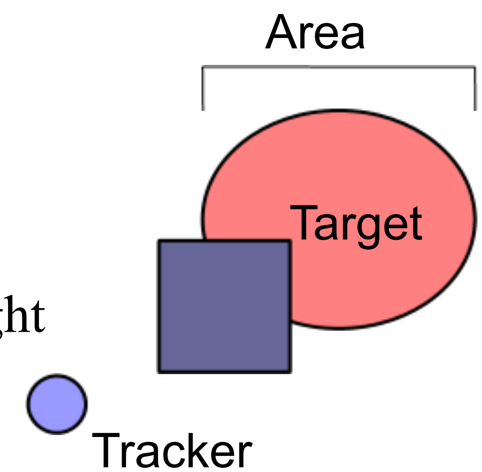
Grid-World Game Tree Search

- Imperfect-information game tree
- Search algorithm to select next action
  - Run it repeatedly as the game progresses
- Search to some *cutoff depth*
  - For the nodes at that depth, compute *heuristic estimates* of their utility values
- Use those estimates to compute estimates of the utility values of the nodes higher in the tree
  - At the top level, choose the action that looks best

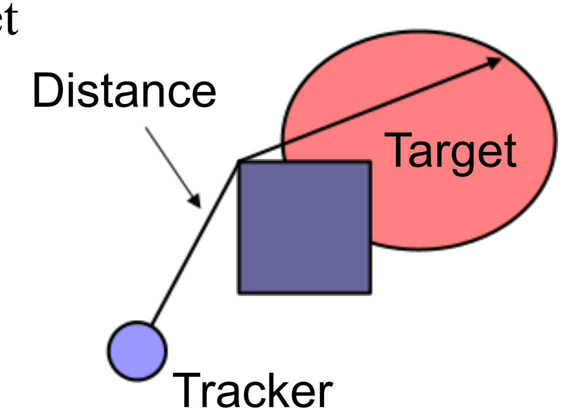


Evaluation Functions

- A ***heuristic evaluation function*** is what's used to estimate the utility values of the leaf nodes



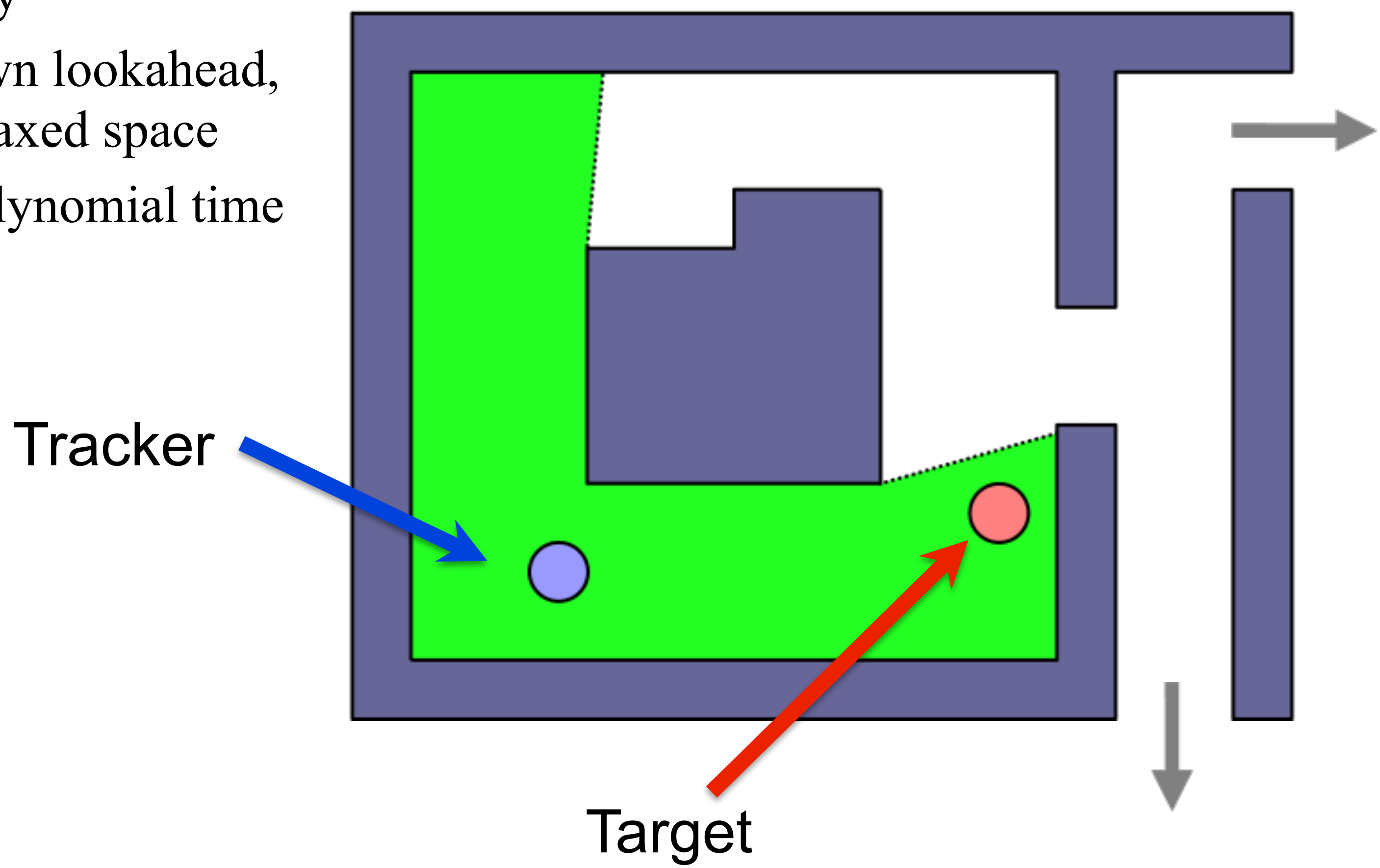
- Some simple ***local heuristics***:
  - » **Region Size (RS)**:
    - size of the region where the target might be located
  - » **Max Distance (MD)**:
    - maximum possible distance to the target



- But we can do better …

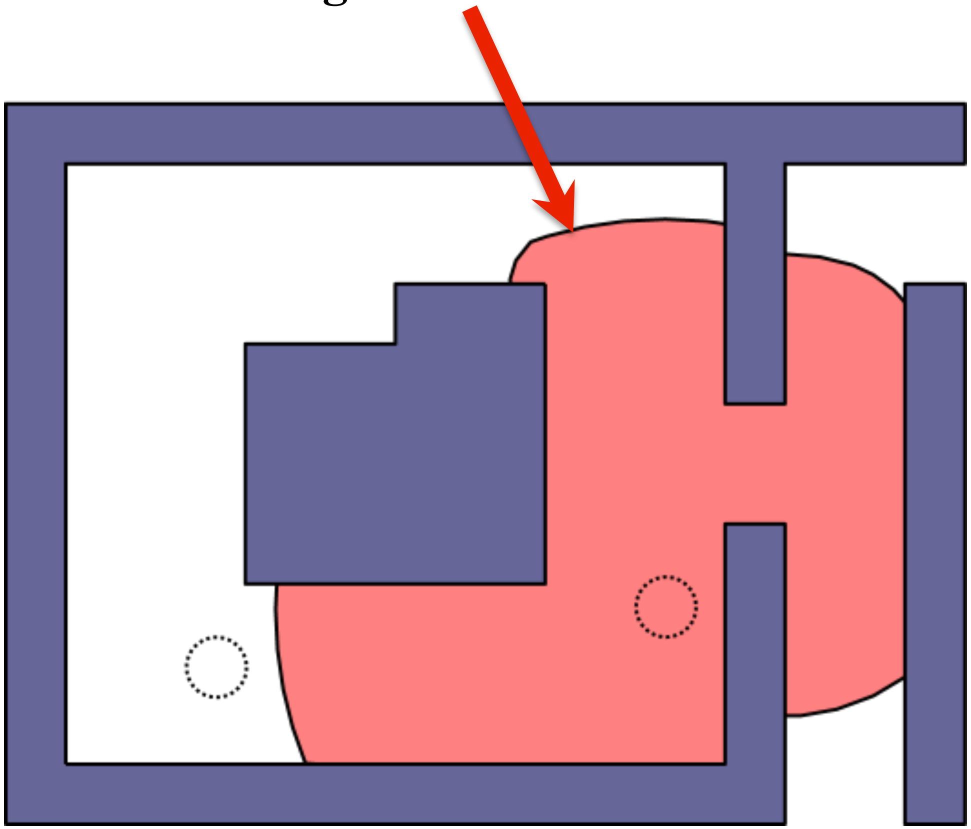
Predicting Loss of Visibility

- Heuristic algorithm to predict loss of visibility
  - » Does its own lookahead, but in a relaxed space
  - » Runs in polynomial time



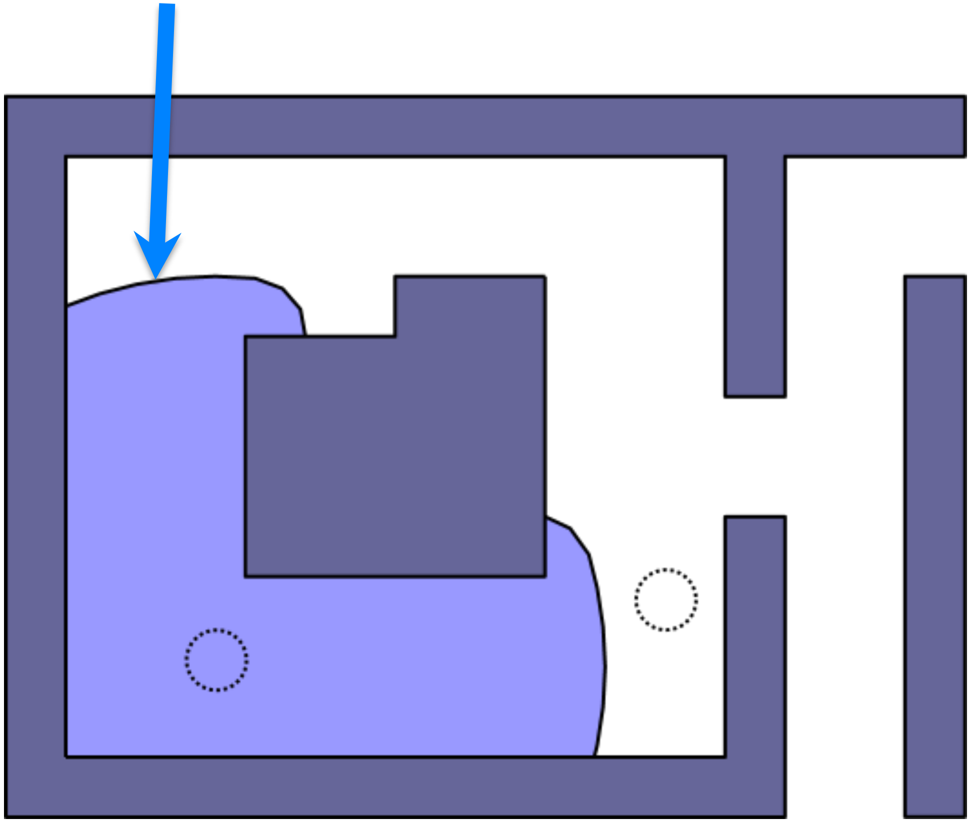
Predicting Loss of Visibility

- Set of locations that it's possible for the **target** to reach in time *t*



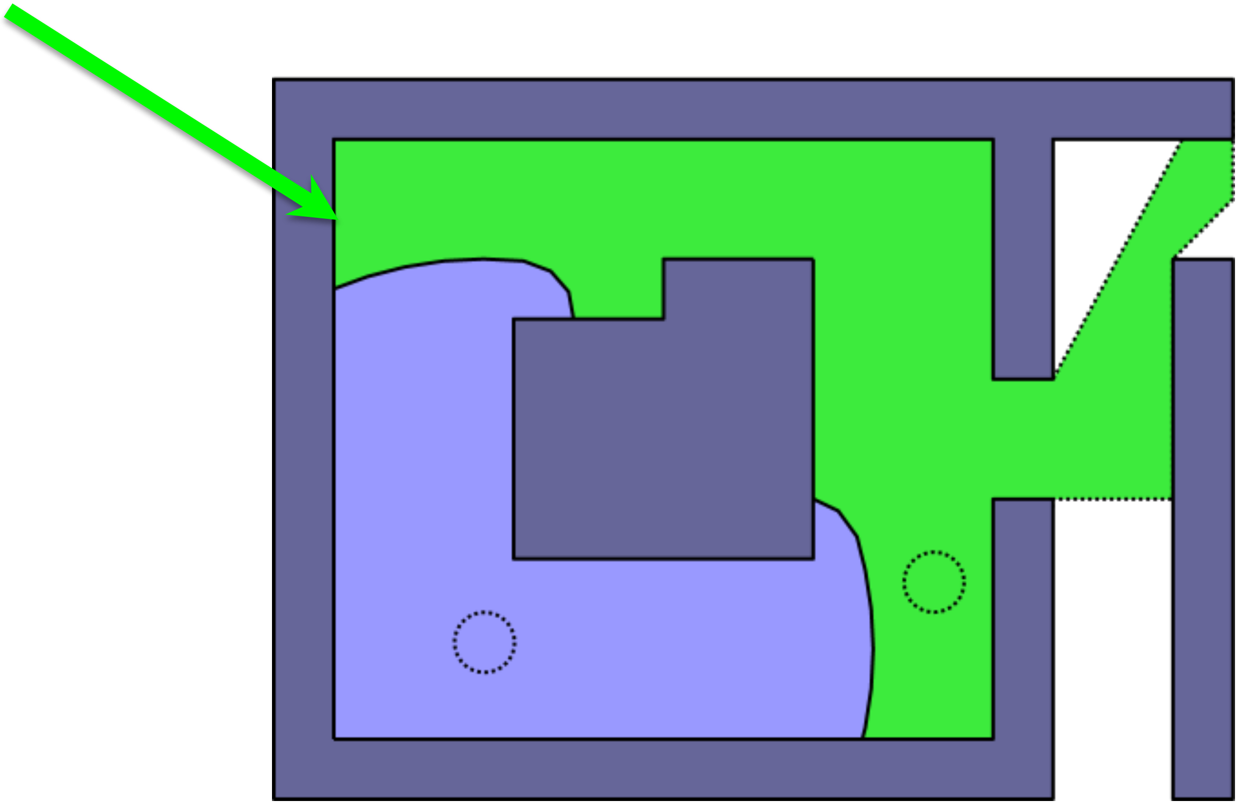
Predicting Loss of Visibility

- Set of locations the **tracker** can reach in time *t*



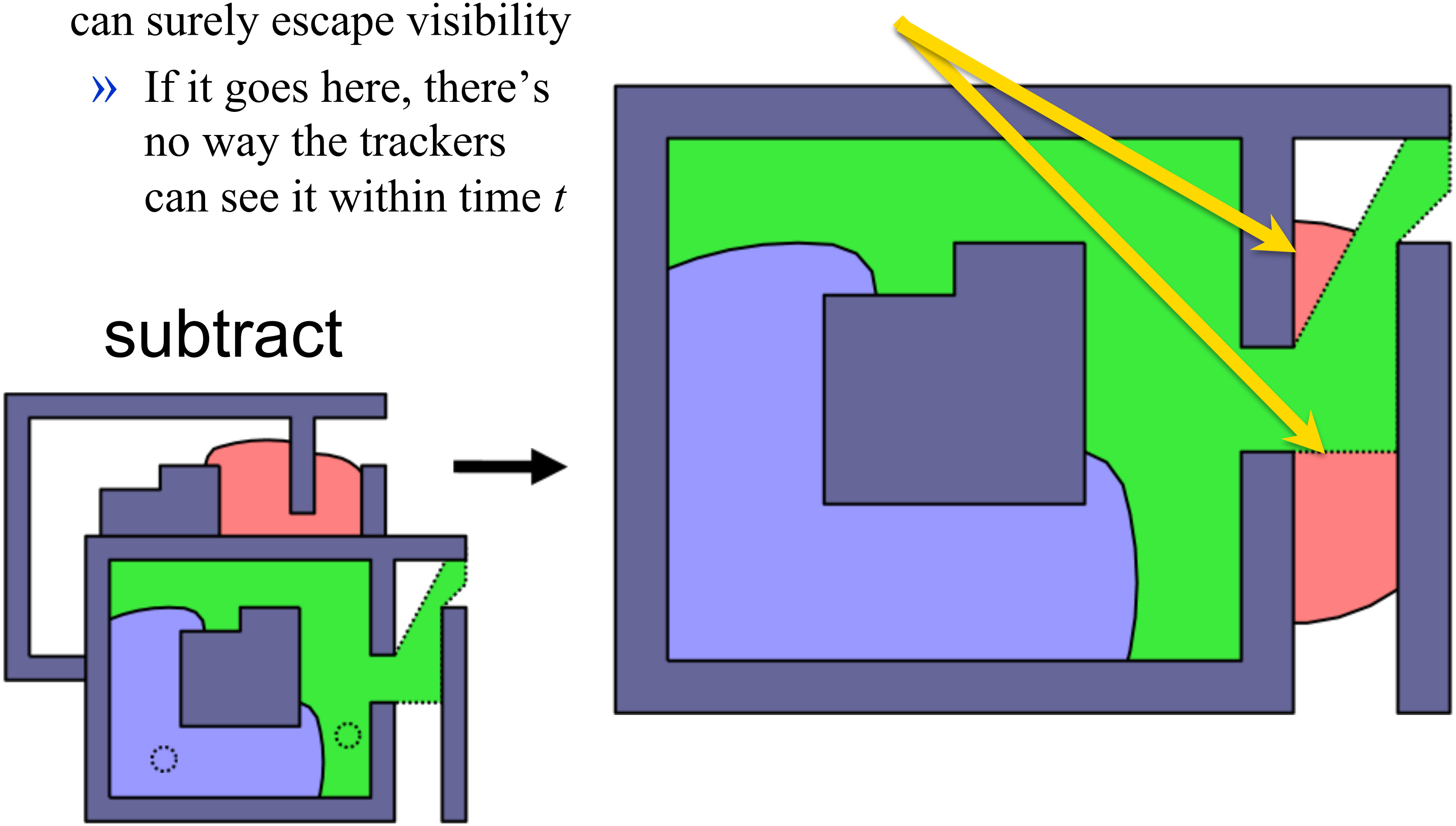
Predicting Loss of Visibility

- Locations that are **visible** from locations the tracker can reach by time *t*



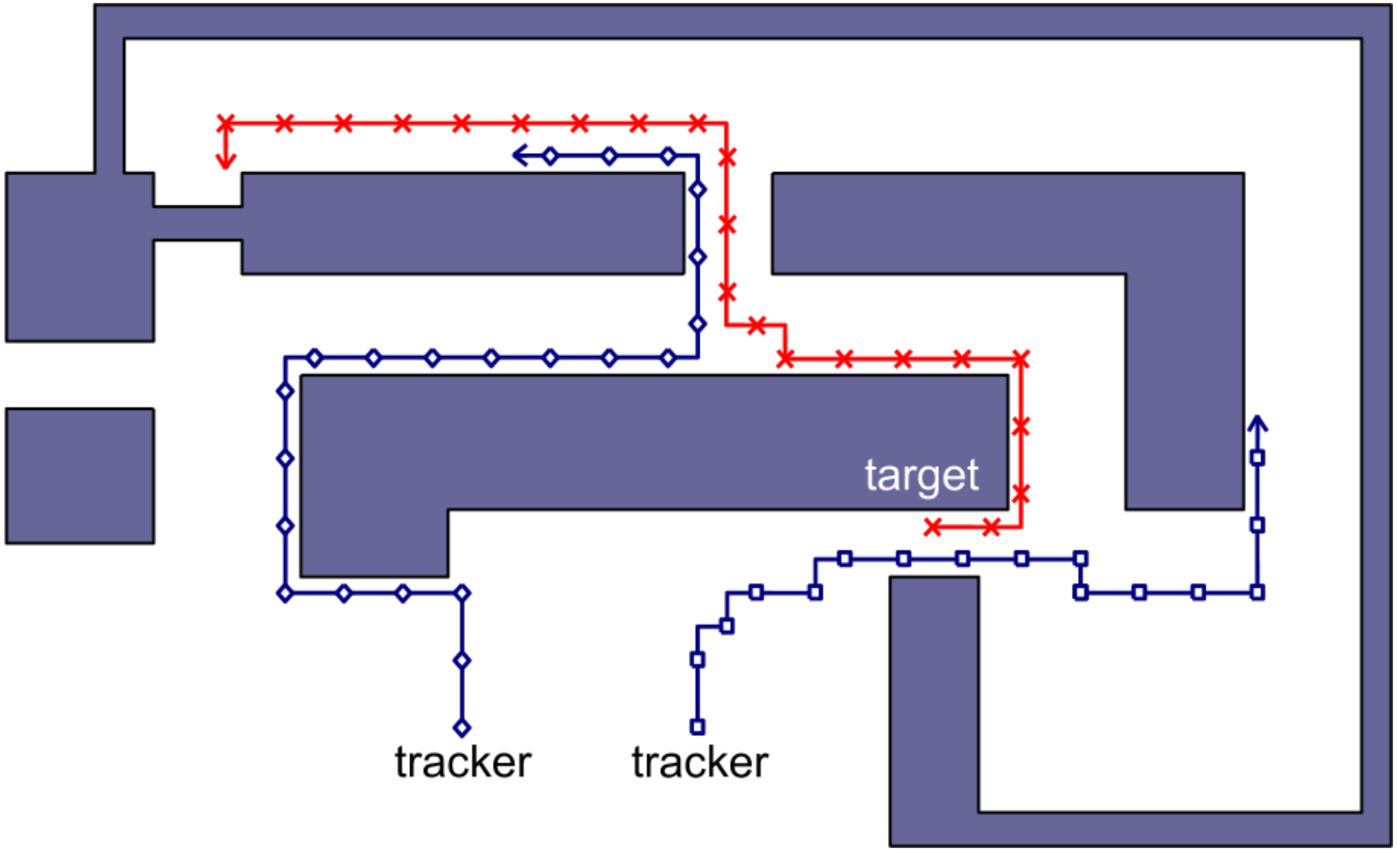
Predicting Loss of Visibility

- **Relaxed Lookahead (RLA)** heuristic: size of the region where the target can surely escape visibility
  - If it goes here, there's no way the trackers can see it within time *t*



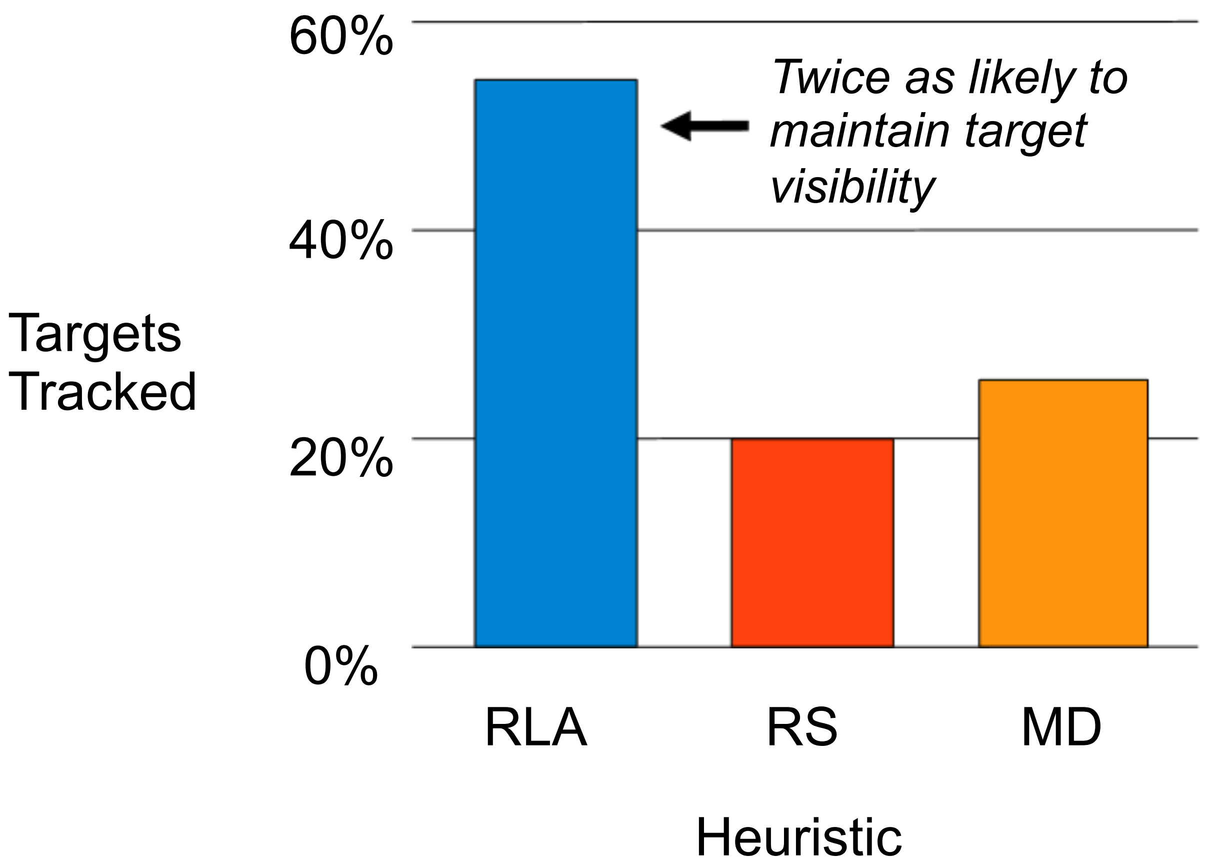
Performance

- RLA can be computed very quickly
- It produces significantly better strategies than the local heuristics



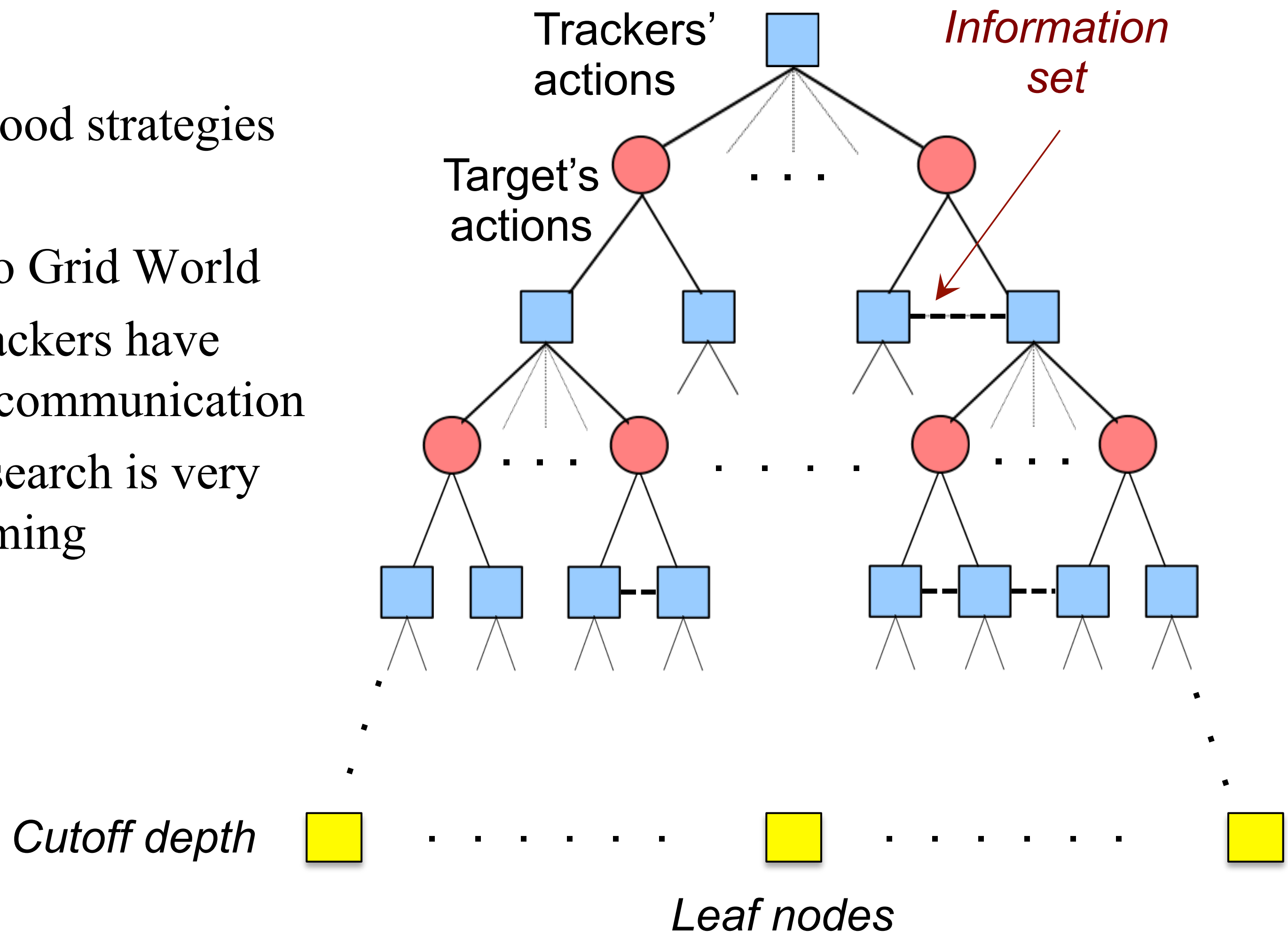
Experimental Results

- Two tracker agents, one target agent
- Three heuristics: RLA, Max Distance, Region Size
- 500 randomly generated trials



Advantages and Disadvantages

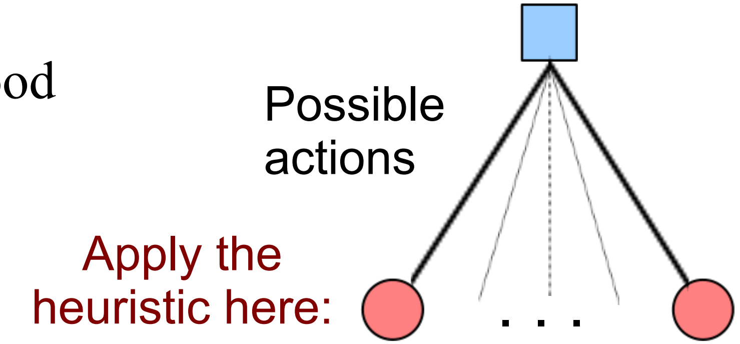
- Advantage:
  - Generates good strategies
- Disadvantages:
  - Restricted to Grid World
  - Assumes trackers have continuous communication
  - Game-tree search is very time-consuming



Part 1(b). Later Work

E. Raboin, D. Nau, and U. Kuter. Generating strategies for multi-agent pursuit-evasion games in partially observable Euclidean space. In *Autonomous Robots and Multirobot Systems (ARMS) Workshop*, 2012. pdf.

- Observations
  - The game-tree search is more-or-less superfluous
  - If we apply RLA directly to the available actions
    - i.e., depth-1 search
    - We get strategies that are nearly as good
    - We get them **much** more quickly

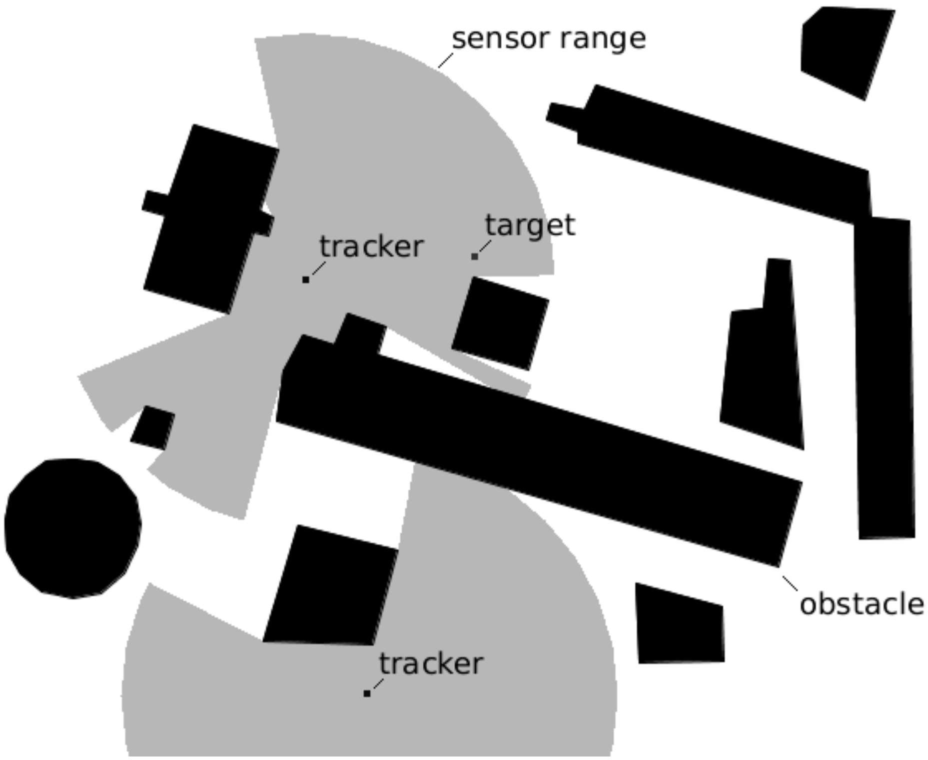


- Thus:
  - Develop a heuristic similar to RLA, but for continuous Euclidean space
  - Design it to be tolerant of interruptions in communication
  - Use it directly, rather than doing a game-tree search

The LEL Heuristic

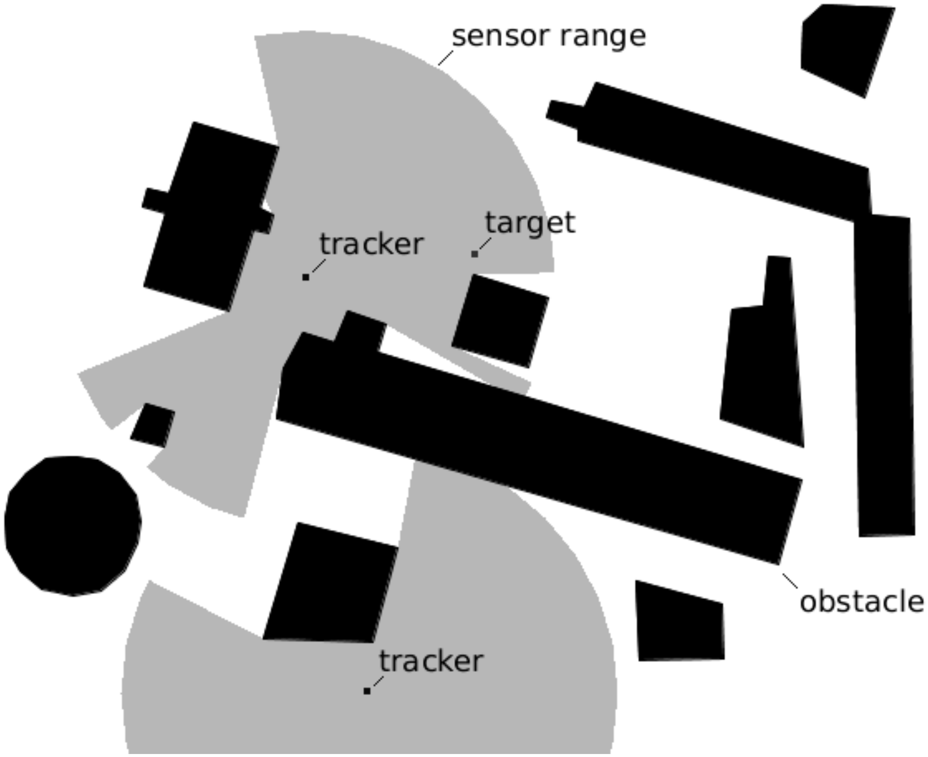
- **LEL**: **L**imited-communication **E**uclidean-space **L**ookahead

- Notation:
  - » Agents $a_0, a_1, \ldots, a_n$,
    - a_0 is the target
    - $a_1, \ldots, a_n$ are the trackers
  - » A location l is **hidden** at time t if no agent will be able to reach, by time t, a location from which l can be observed



- In principle, we want this:
  - » LEL = {area of the set of hidden locations that a_0 can reach by time t}
    averaged over some time interval}

Imperfect Information

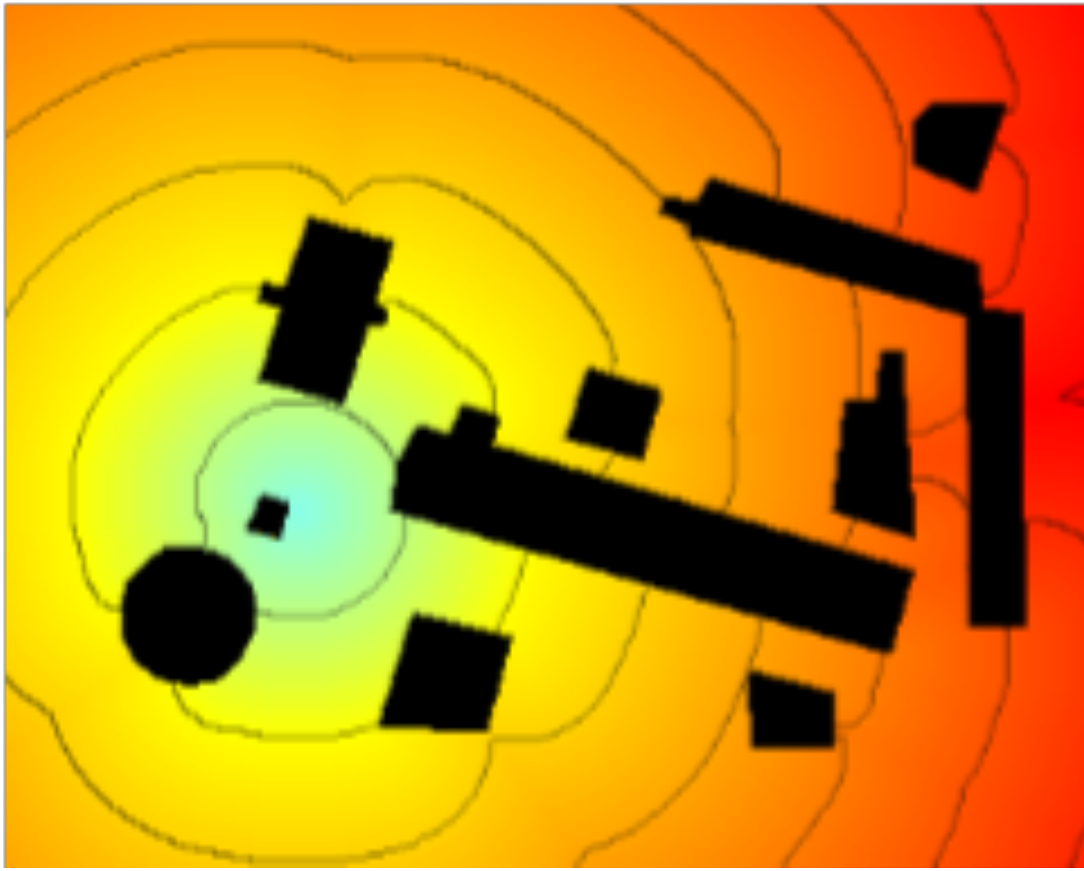


- We need to deal with **imperfect information**
  - » Limits on the sensor data that a_i can acquire directly
  - » Limits on the information that the other trackers can communicate to a_i
- For each tracker agent a_i, let

  O_i = {all observations available to a_i}
  = {observations a_i has made}
  ∪{all observations the other trackers have been able to send to a_i}
- Calculate values conditioned on O_i

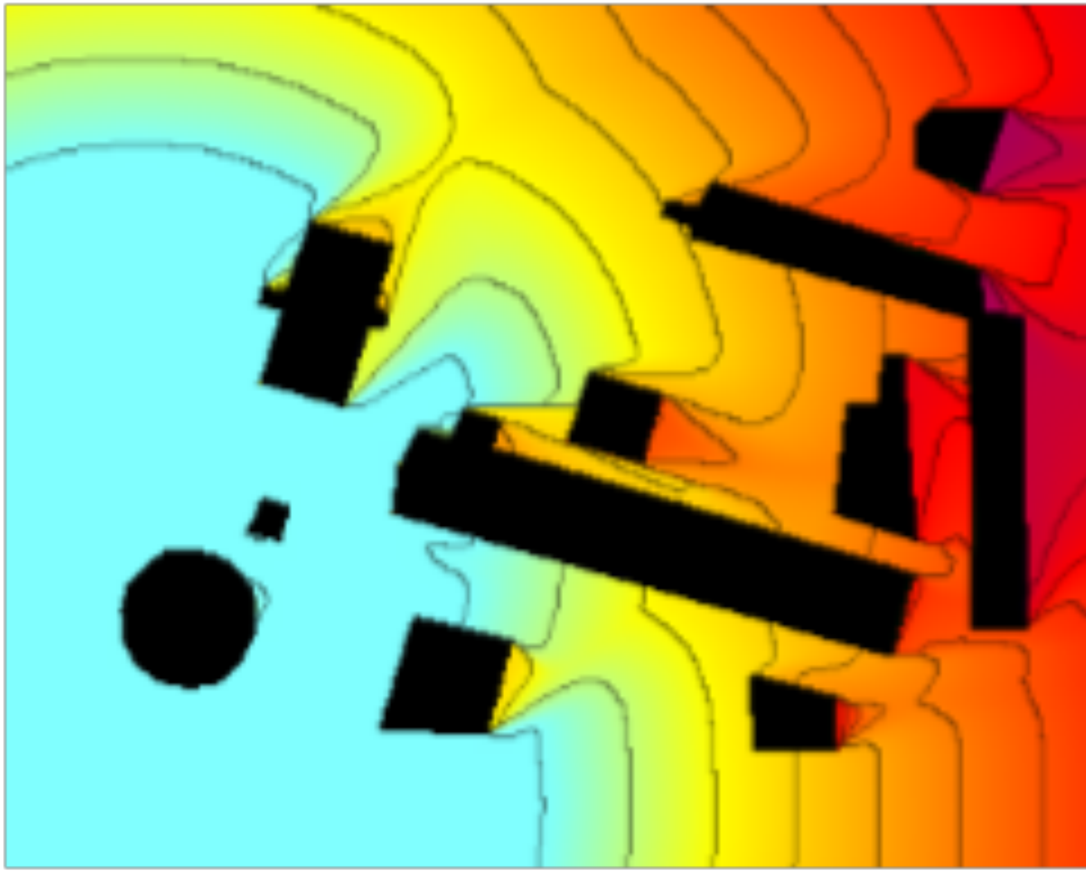
Locations the Target Might Reach

Reachable$(t \mid O_i)$
= {all locations that the **target** might be able to reach by time t, given O_i}



Heat graph:
color shows *Reachable*$(t \mid O_i)$
for increasing t

Locations the Trackers Might See



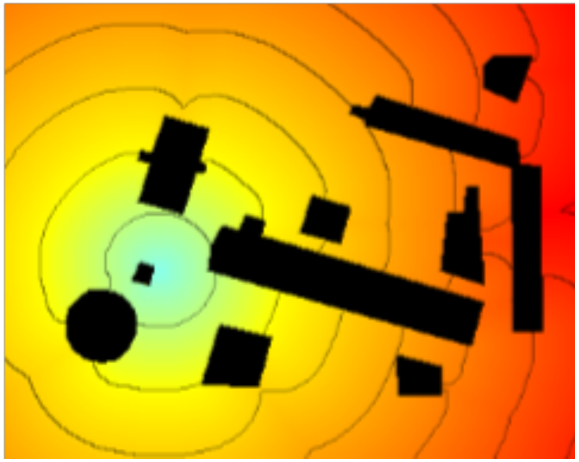
$Viewable(t \mid O_i)$

= {all locations that the **trackers** might be able to observe at time t, given O_i}

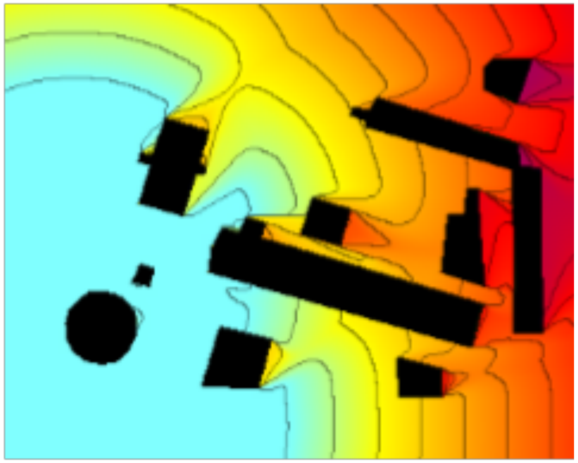
Heat graph:
color shows $Viewable(t \mid O_i)$
for increasing t

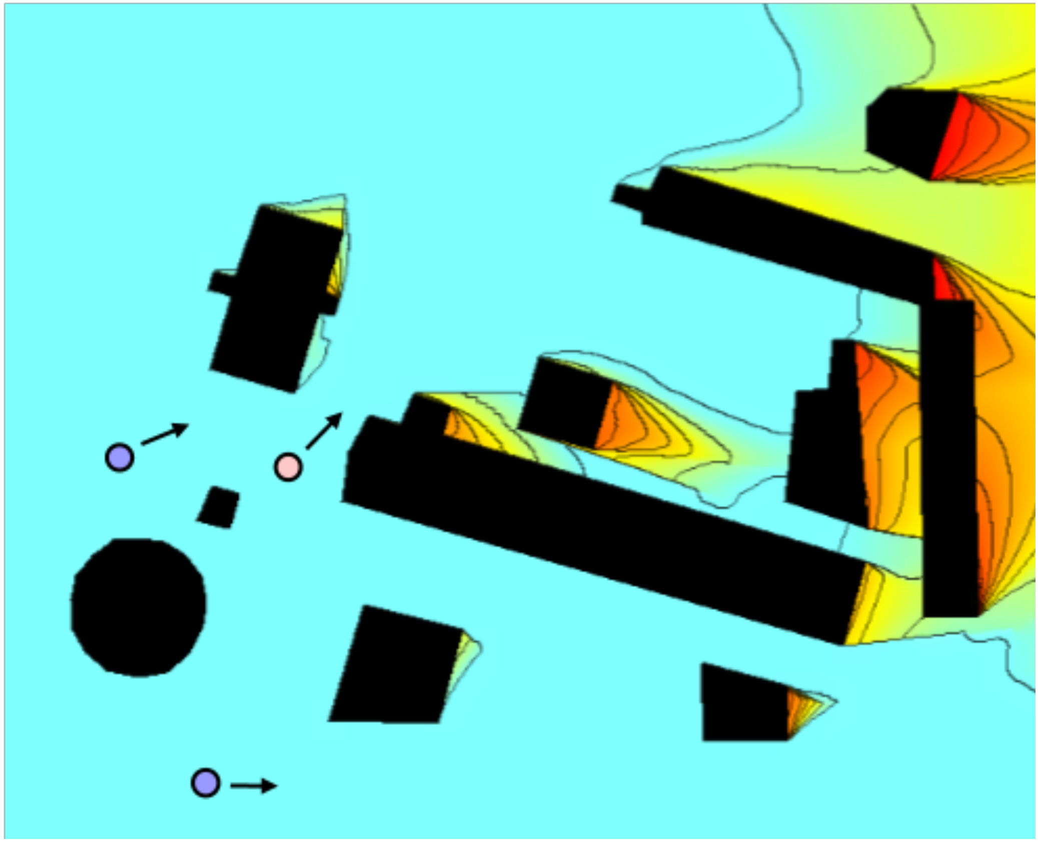
The LEL Heuristic

Reachable$(t \mid O_i)$



Viewable$(t \mid O_i)$





Heat graph:

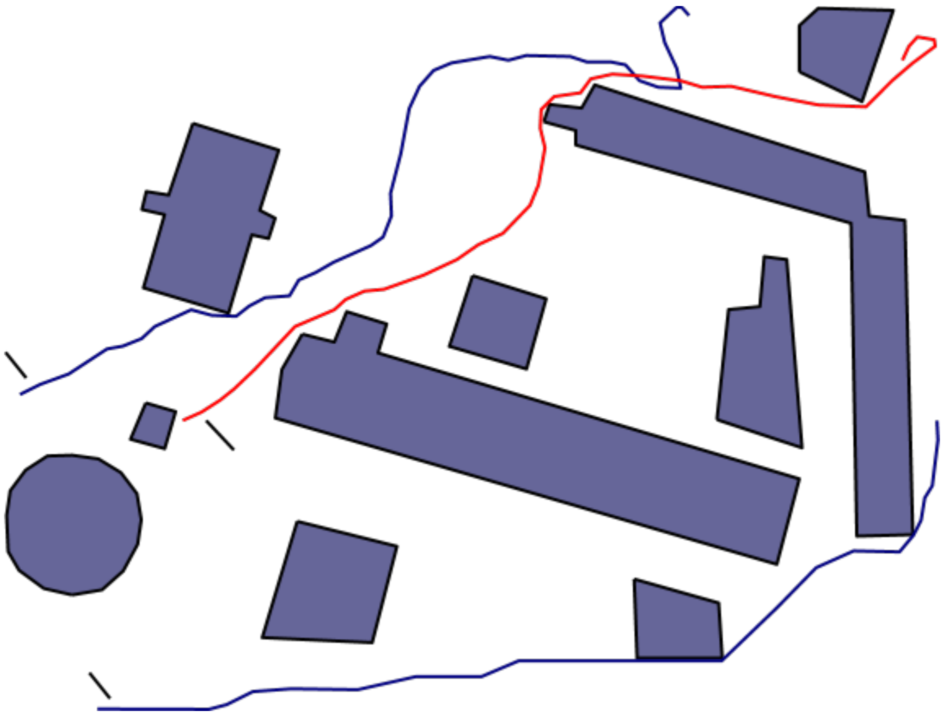
color shows *Reachable*$(t \mid O_i)$ – *Viewable*$(t \mid O_i)$ for increasing t

LEL =

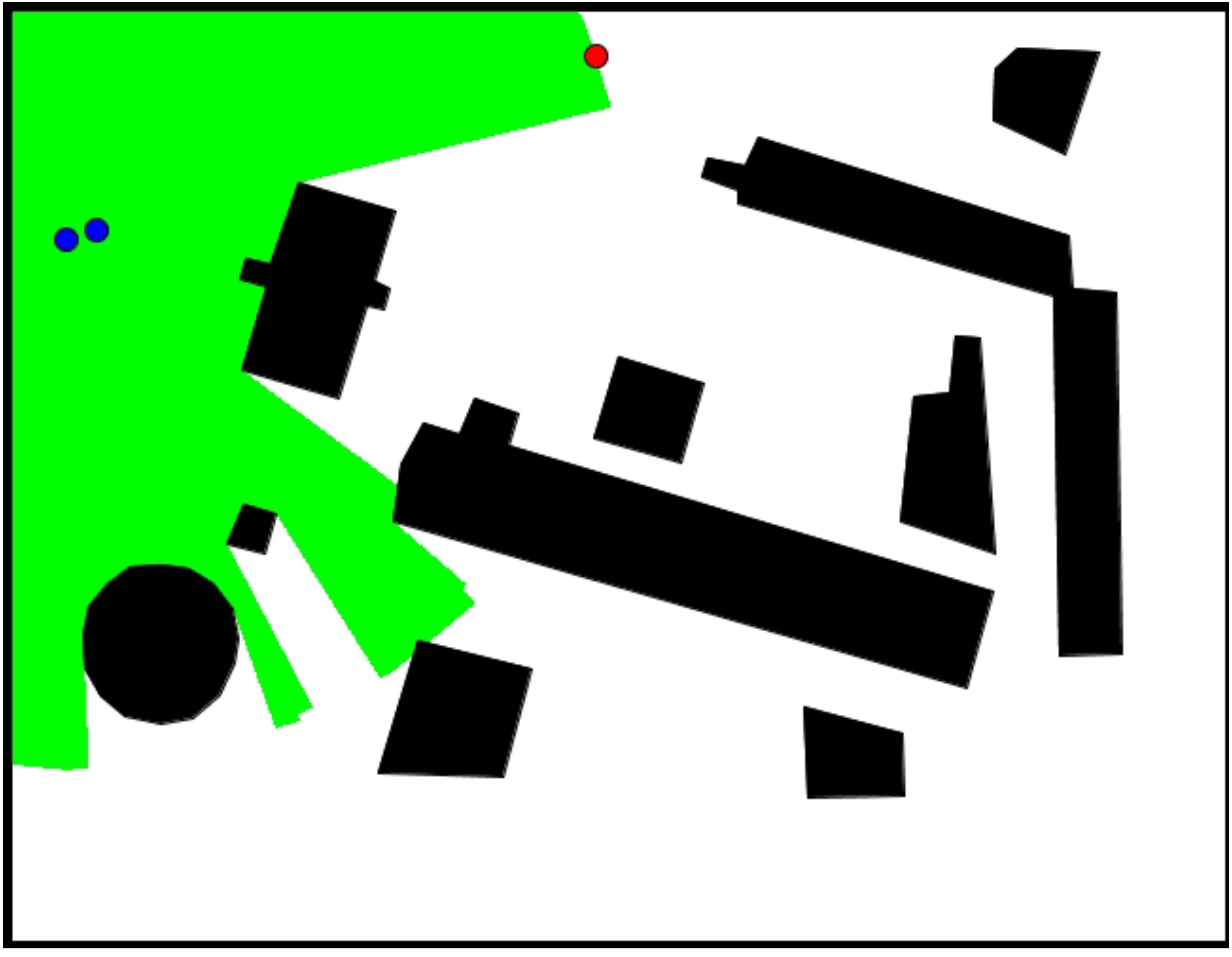
area of *Reachable*$(t \mid O_i)$ – *Viewable*$(t \mid O_i)$, averaged over some time interval

Cooperative Team Behavior

- Trajectories generated using LEL
  - two tracker agents (blue)
  - one target agent (red)



Video



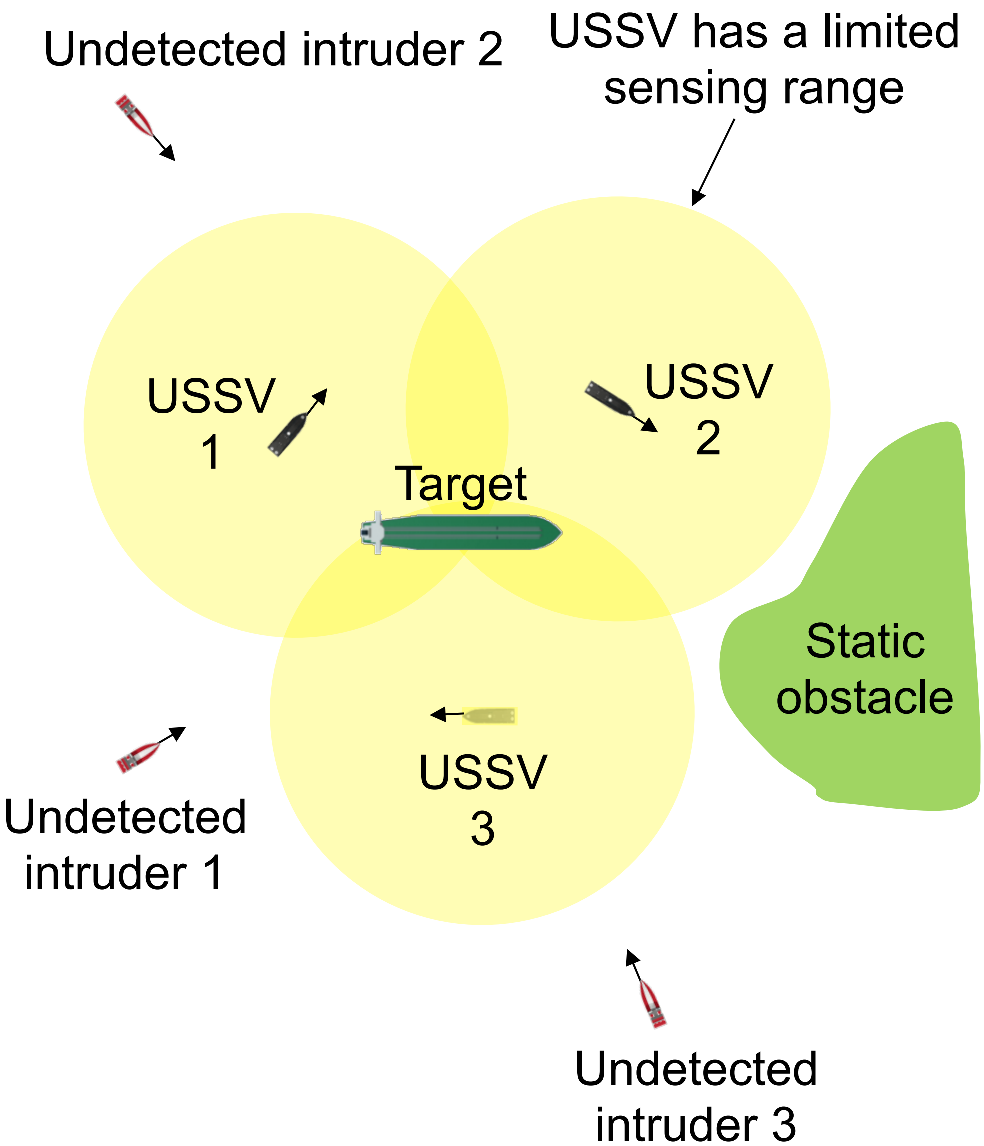
Part 2. Naval Asset Protection

- Generation of behaviors to support multi-USSV blocking and patrolling in the presence of intelligent adversaries

E. Raboin, P. Švec, D. S. Nau, and S. K. Gupta. Model-predictive target defense by team of unmanned surface vehicles operating in uncertain environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013. pdf.

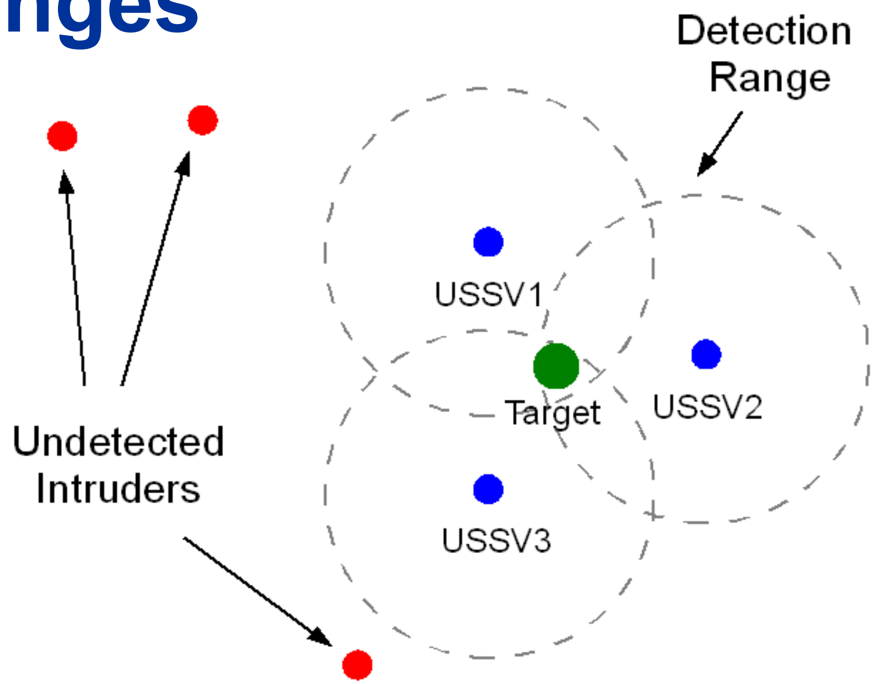
Motivation and Goals

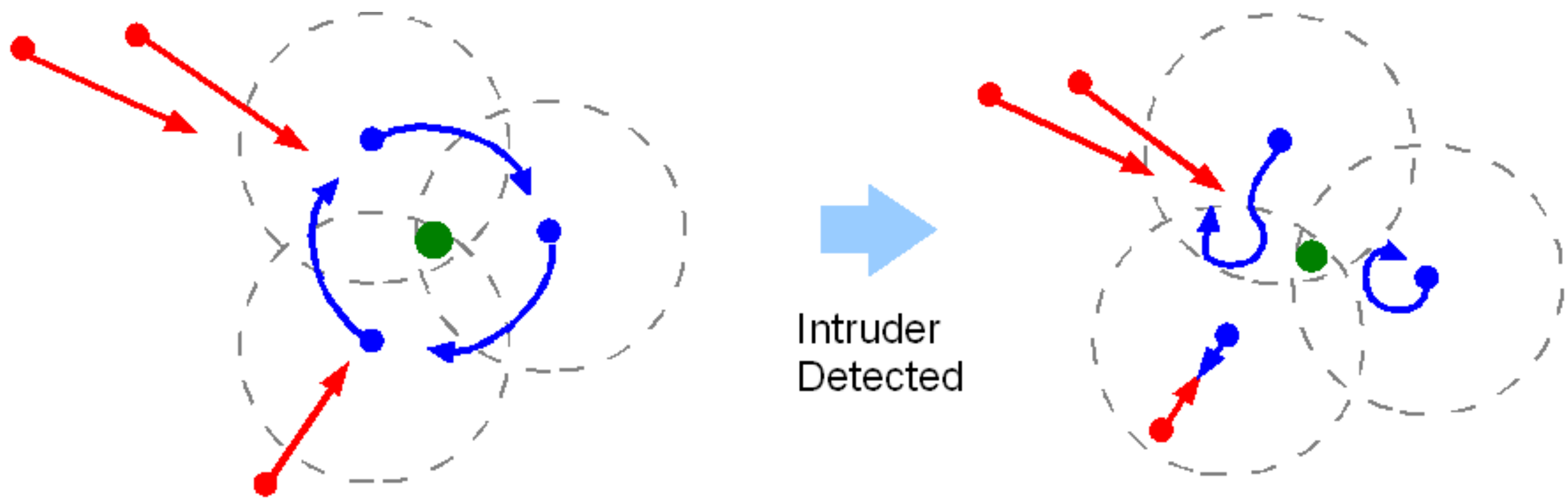
- Group of semi-autonomous USSVs required to patrol an area around an asset to protect it against intruder boats
  - USSVs can have different physical properties
- Goals:
  - Generate multi-USSV, decentralized, and adaptable behavior for patrolling and blocking multiple intruders



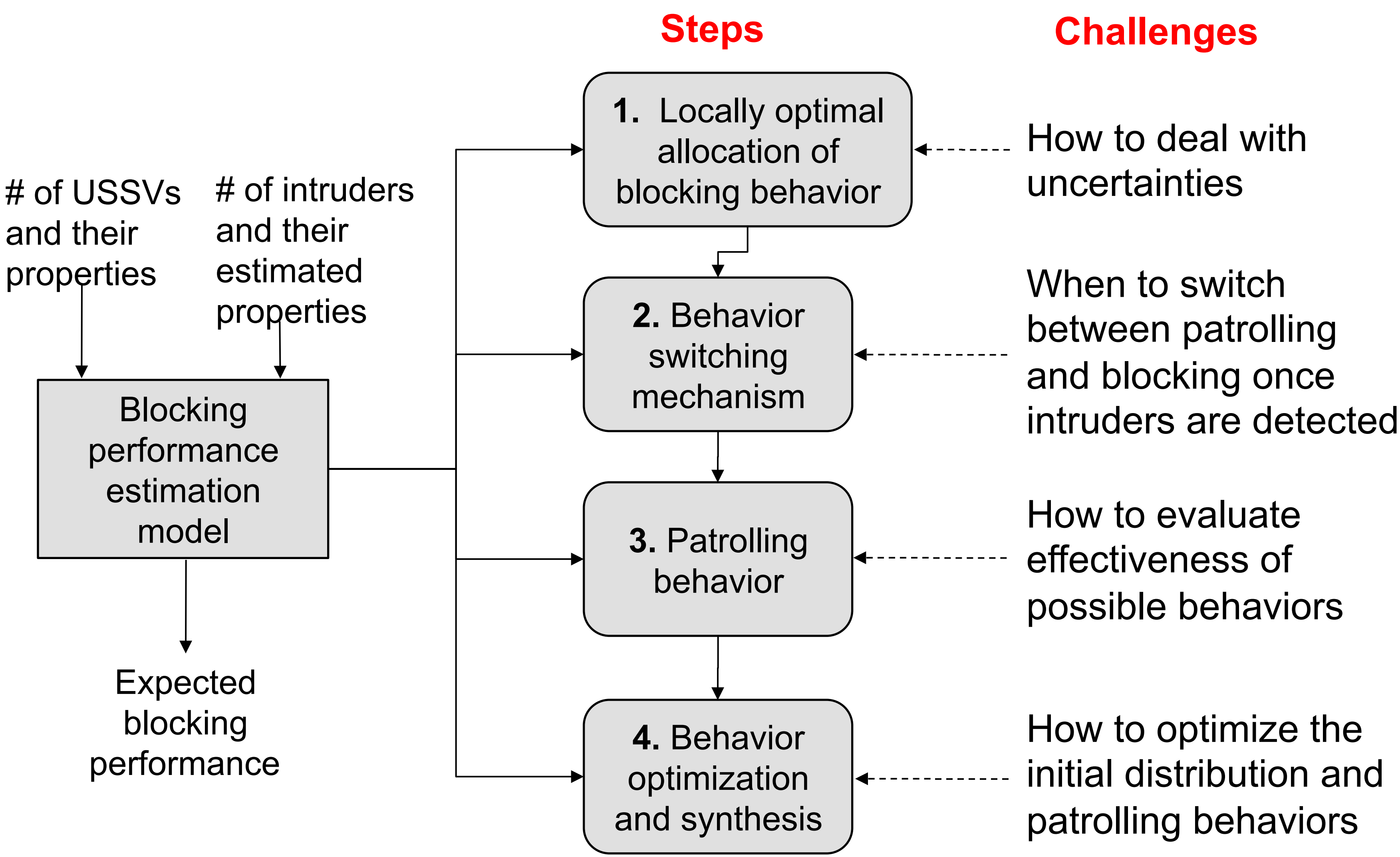
Challenges

- Initial distribution and patrolling behaviors of USSVs
- Patrolling behavior adaptation
  - When to switch between patrolling and blocking behaviors once intruders are detected
  - How to allocate blocking behaviors among USSVs

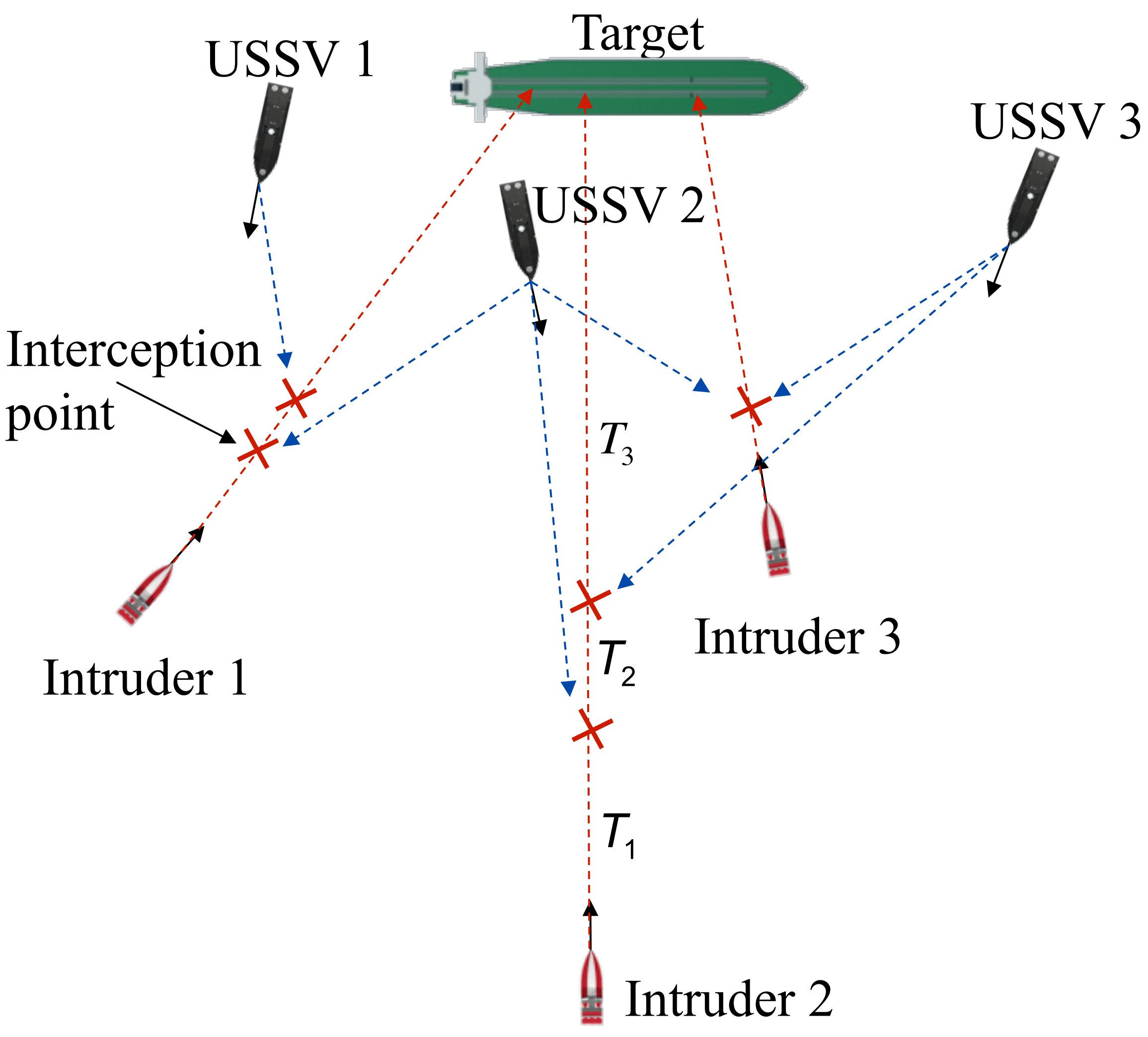




Overall Approach

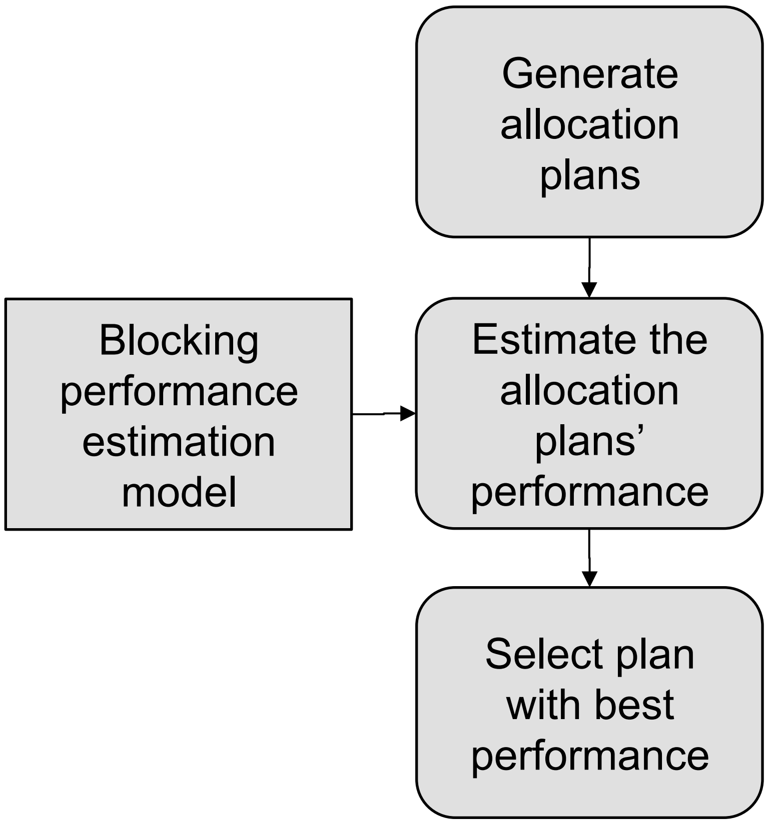


(1) Locally Optimal Blocking Behavior Allocation



(1) Locally Optimal Blocking Behavior Allocation

- Decentralized allocation of blocking behaviors
  - Search for locally best-performing assignment
  - Use knowledge about the defenders' blocking capabilities



$P_1 = (U_1 - I_1, U_2 - I_2, U_3 - I_3)$ **Arr. time 1**

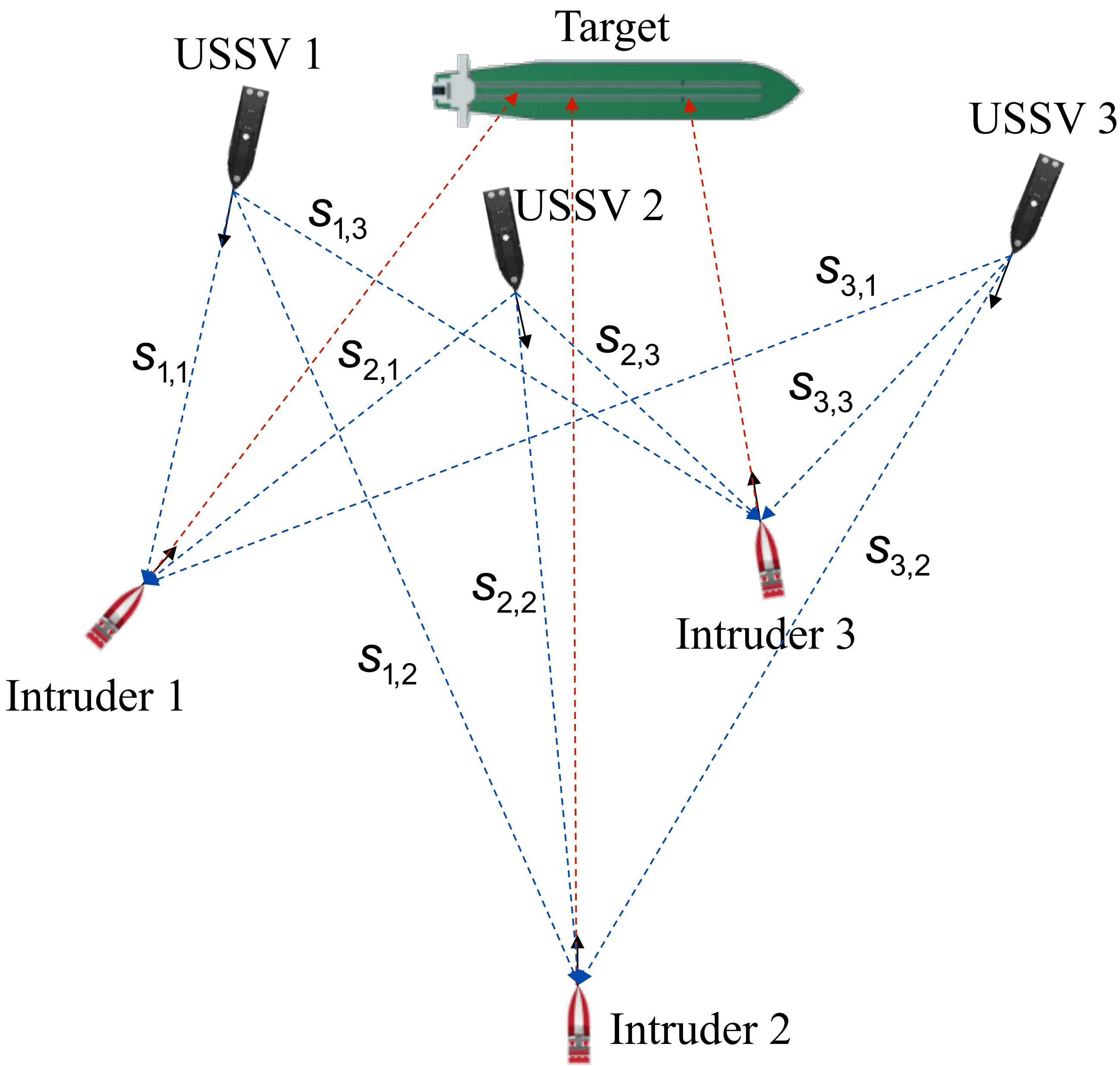
$P_2 = (U_1 - I_1, U_2 - I_3, U_3 - I_2)$ **Arr. time 2**

$P_3 = (U_1 - I_2, U_2 - I_2, U_3 - I_3)$ **Arr. time 3**

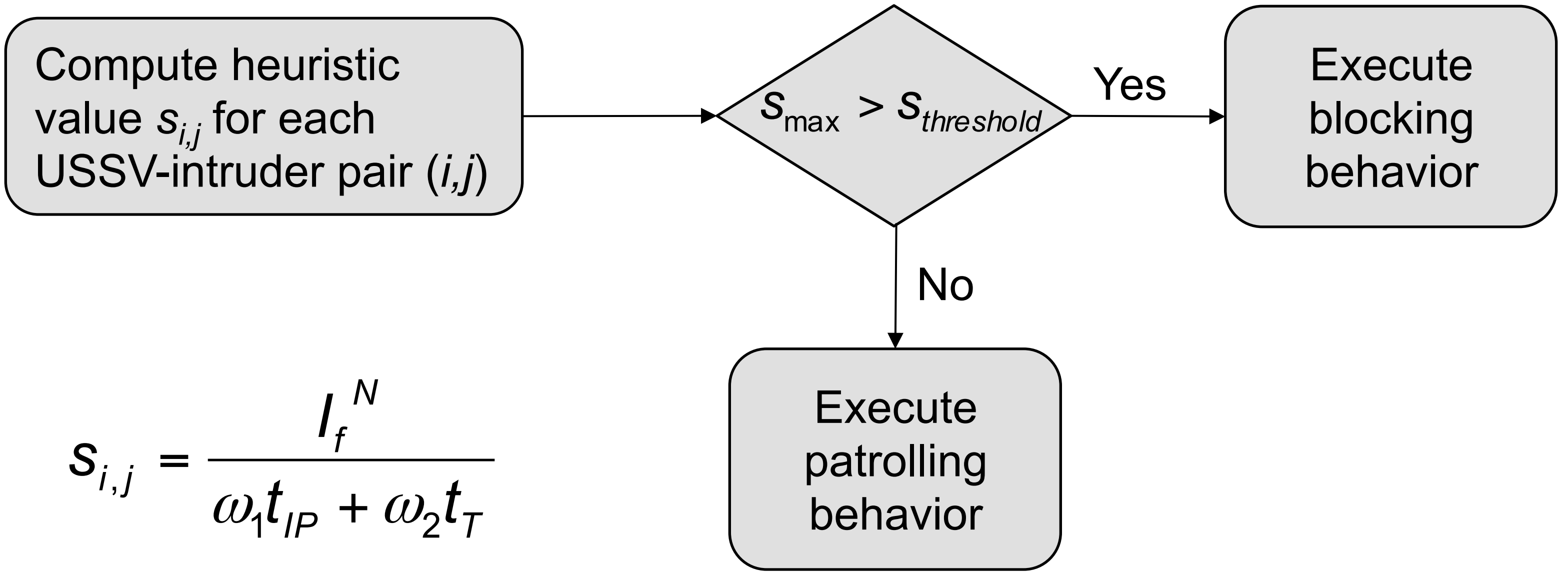
$\dots$

$P_1 = (U_1 - I_1, U_2 - I_2, U_3 - I_3)$

(2) Behavior Switching Mechanism



(2) Behavior Switching Mechanism



$$s_{i,j} = \frac{I_f^{\,N}}{\omega_1 t_{IP} + \omega_2 t_T}$$

I_f – Intruder supress value parameter

N – Number of USSVs following / blocking the intruder j

t_{IP} – Arrival time to the intersection point of the intruder j

t_T – Arrival time to the target

ω_1, ω_2 – Weight parameters

(3) Patrolling Behavior

The quality of a patrol route can be evaluated given a set of hypothetical intruders. For each intruder *i*, compute:

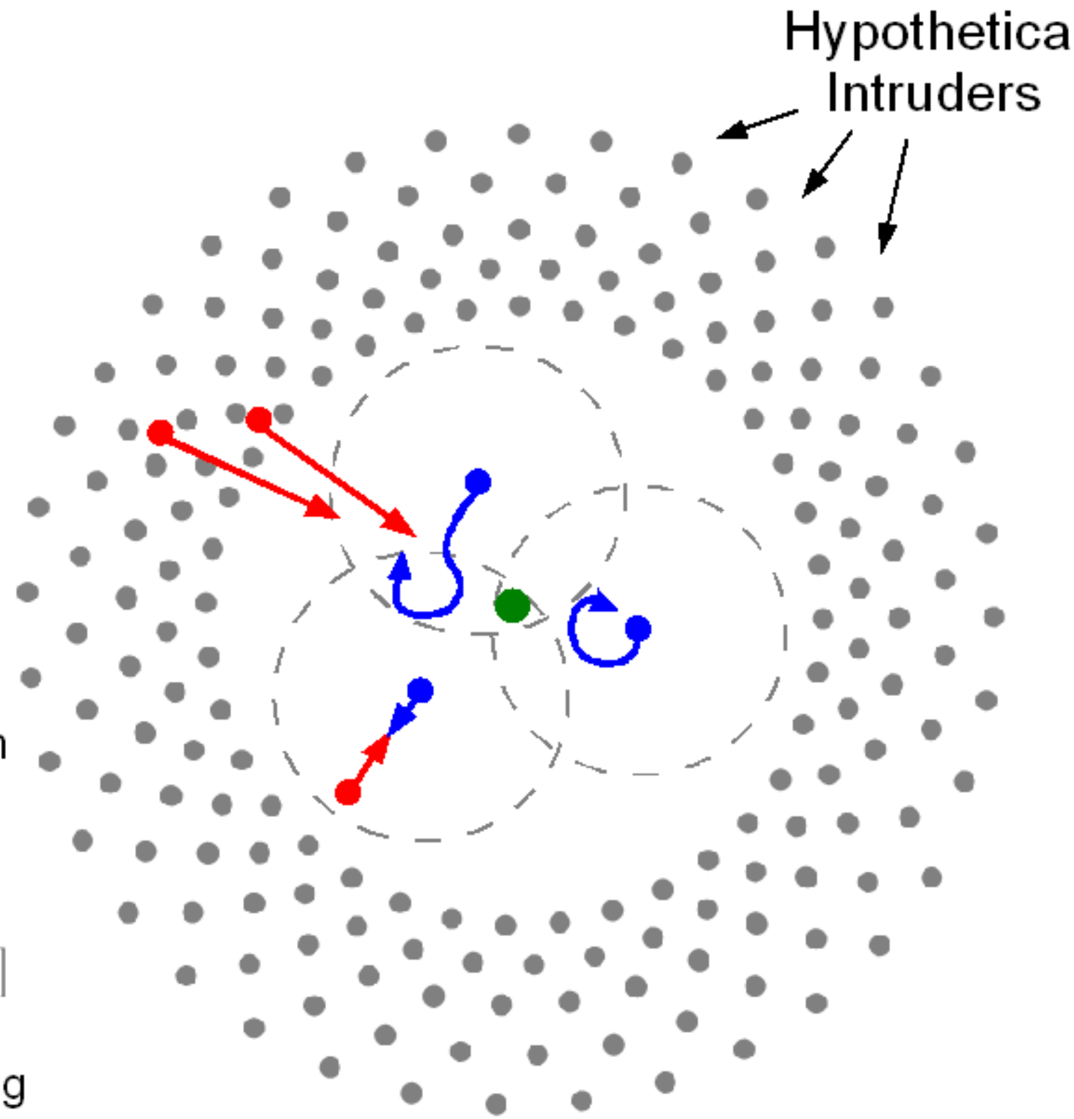
$P_i(t_d | r_j)$: probability that intruder *i* will be detected at time t_d by a USSV following route r_j

$t_i(t_d, r_j)$: time when intruder *i* will collide with the target if detected a time t_d and immediately intercepted by a USSV following route r_j

Given set of routes, $r_1, r_2, \ldots r_n$ for a team of *n* USSVs, we estimate the expected collision time using the formula:

$$\frac{1}{mk} \Sigma_{i=1}^{m} \Sigma_{t=t_0}^{t_k} max_j [P_i(t|r_j) t_i(t, r_j)]$$

This can be computed efficiently, allowing for thousands of candidate routes to be evaluated in under a second.

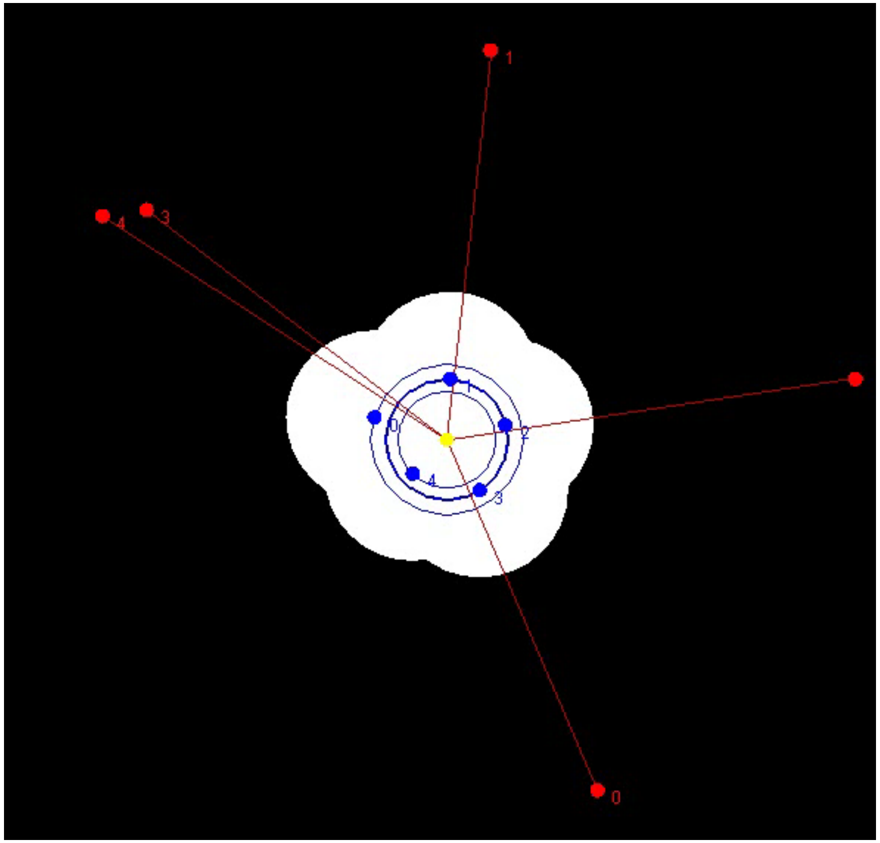


(4) Policy Optimization

- Genetic Algorithm (GA) to optimize:
  - (d_i, ϕ_i) – Initial locations of USSVs around the target
  - $s_{threshold}$ – Heuristic value threshold
  - I_f – Intruder suppress value parameter
  - ω_1, ω_2 – Shame equation weight parameters

- Several different types of defending policies
  - Baseline
  - Motivation-based
  - Purely patrolling
  - Hybrid (combination of Motivation-based & Patrolling)

Multi-USSV Patrolling Behaviors: Results



Simple defender policy

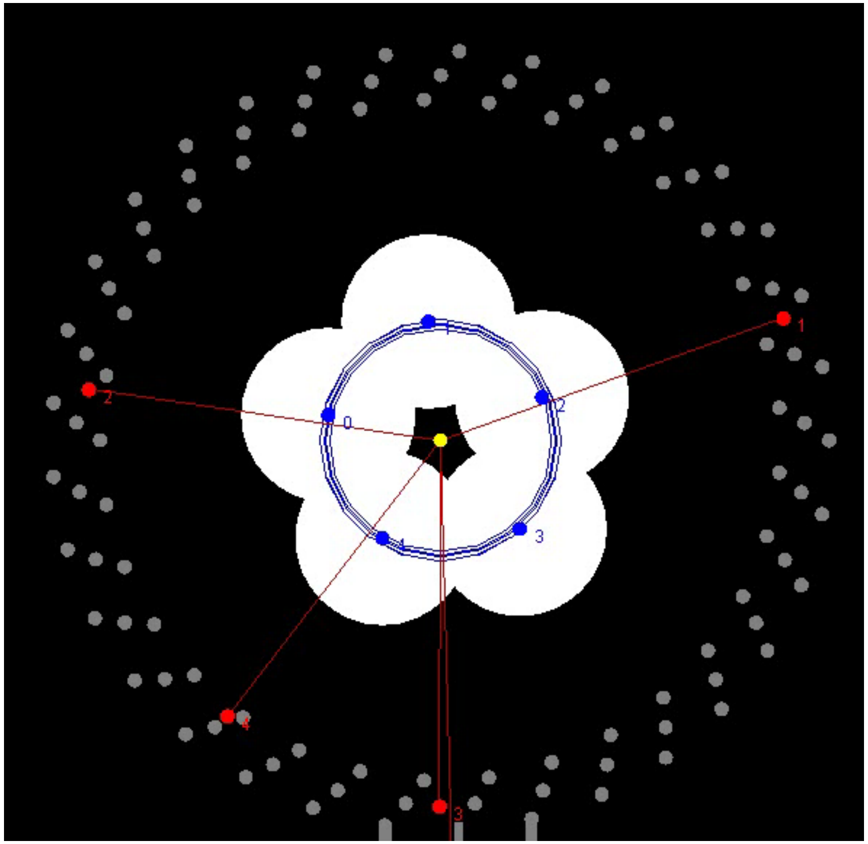
Max. velocity of USSVs: 10 m/s

Max. velocity of intruders: 9 m/s

Visibility radius of vehicles: 60 m

Range of initial distances of intruders from the target: 250-300 m

Average optimized initial distance of USSVs from the target: 42 m



Purely patrolling defender policy

Max. velocity of USSVs: 10 m/s

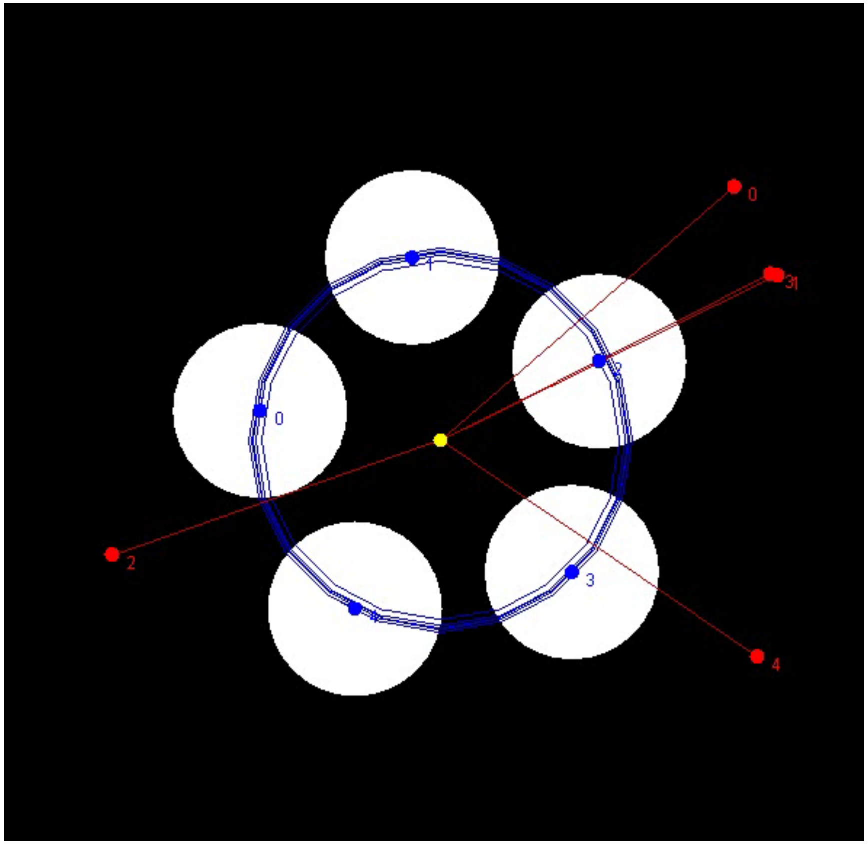
Max. velocity of intruders: 9 m/s

Visibility radius of vehicles: 60 m

Range of initial distances of intruders from the target: 250-300 m

Average optimized initial distance of USSVs from the target: 80 m

Multi-USSV Patrolling Behaviors: Results (continued)



Motivation based defender policy

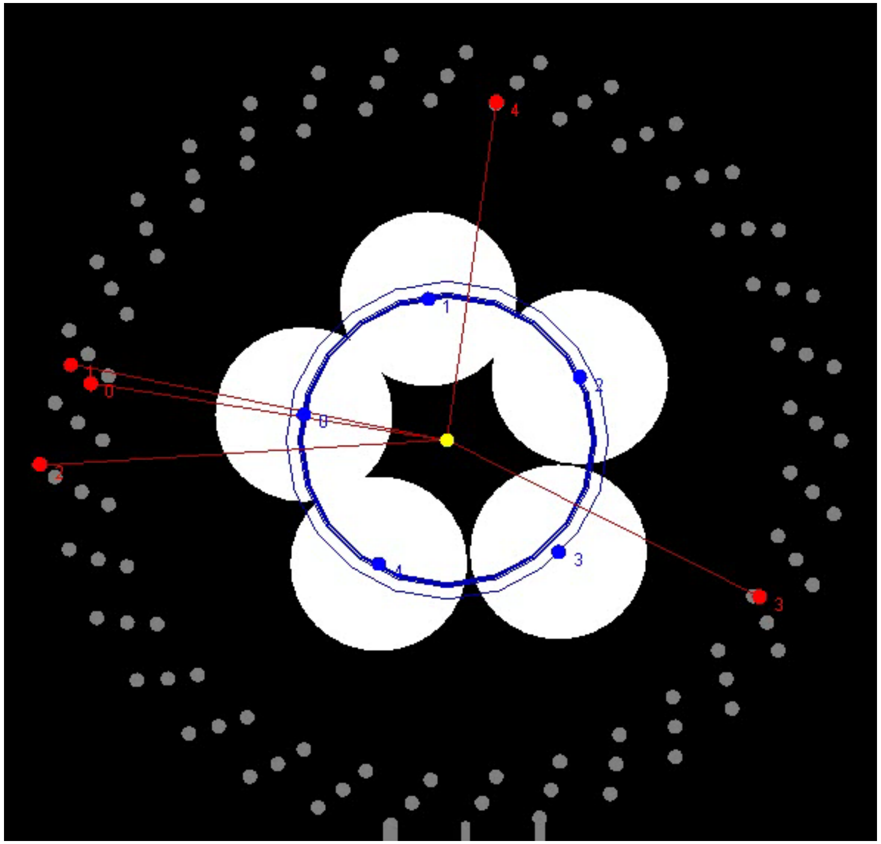
Max. velocity of USSVs: 10 m/s

Max. velocity of intruders: 9 m/s

Visibility radius of vehicles: 60 m

Range of initial distances of intruders from the target: 250-300 m

Average optimized initial distance of USSVs from the target: 128 m



Hybrid defender policy

Max. velocity of USSVs: 10 m/s

Max. velocity of intruders: 9 m/s

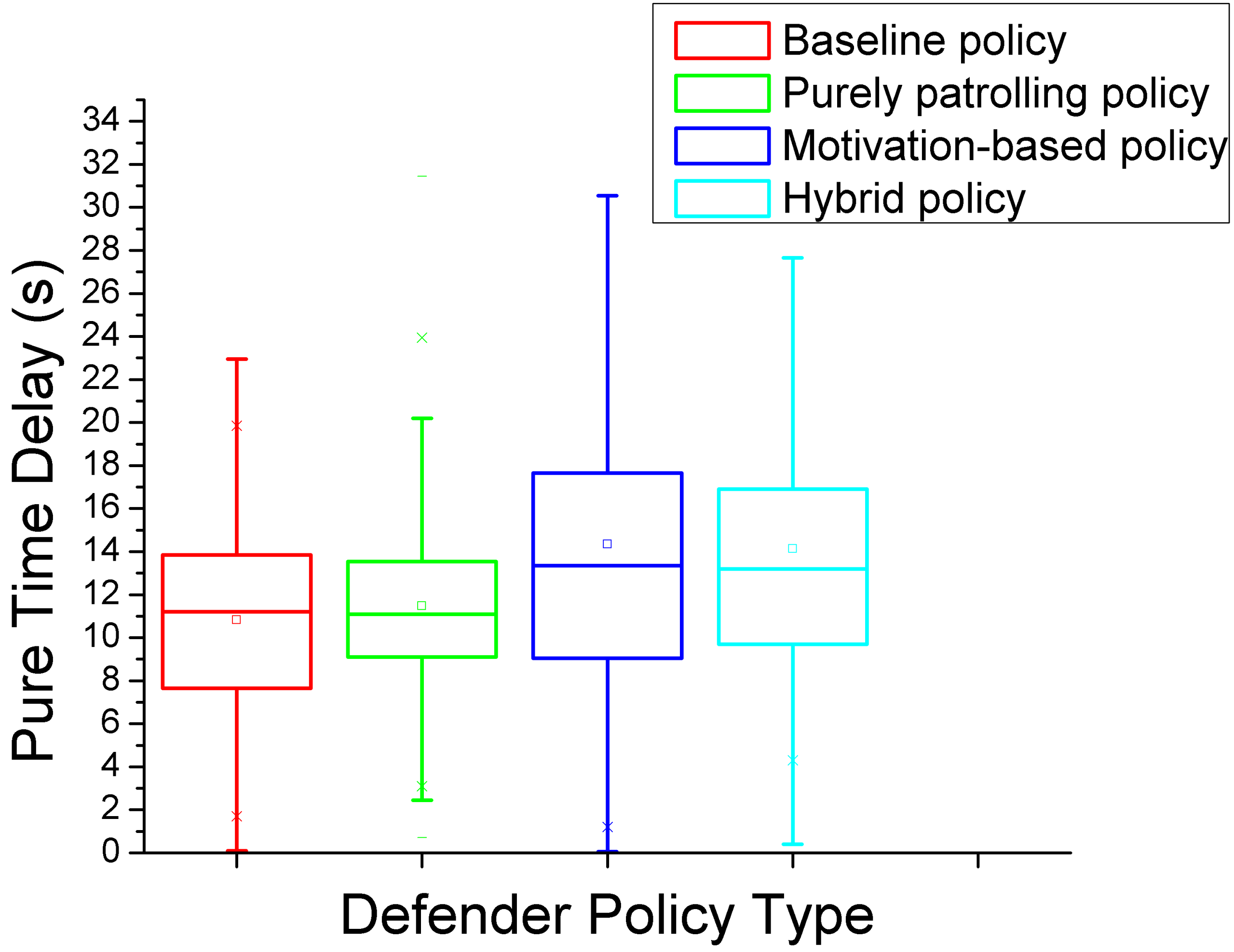
Visibility radius of vehicles: 60 m

Range of initial distances of intruders from the target: 250-300 m

Average optimized initial distance of USSVs from the target: 101 m

Multi-USSV Patrolling Behaviors: Results (continued)

- Policies evaluated using 10000 simulation runs



Summary

- Formalisms and algorithms for multi-agent tracking-and-evasion
    - Temporarily sacrificing visibility can provide a strategic advantage
  - » Gridworld, full communication among trackers
    - ***RLA*** heuristic, imperfect-info game-tree search
  - » Continuous Euclidean space, limited communication among trackers
    - ***LEL*** (Limited-Communication Euclidean Lookahead) heuristic

- Generation of behaviors for multi-USSV blocking and patrolling in the presence of intelligent adversaries
  - » Adaptation of patrolling behavior
  - » When to switch between patrolling and blocking behaviors once intruder is detected
  - » Allocation of blocking behaviors