IWPSS-13
International Workshop on
Planning and Scheduling for Space
NASA Ames Research Center – 26 March 2013

Planning and Policy Learning for Surveillance Missions

Sara Bernardini

King's College London
(sara.bernardini@kcl.ac.uk)

Joint work with Maria Fox and Derek Long

In this talk…

- Introduction: planning and policy learning for surveillance
- Search and Tracking (SAT) via Temporal Planning
  - ✓ Description of the problem
  - ✓ Tracking
  - ✓ Searching as planning problem
  - ✓ SAT planning instances: domains and problems
  - ✓ Search Pattern Generation
  - ✓ Planner
  - ✓ Simulation and Experimental Results
- Extensions:
  - ✓ Obstacles
  - ✓ Spatial Reasoning
- Policy Learning and Policy Integration
- Conclusions

Surveillance Applications

Observer	Target	Area	
Mobile	Interesting Sites	Large	Intelligence Gathering
Mobile	Mobile and Evasive	Small	Search & Tracking
Mobile and Fixed	Physical Flaws, Environmental Readings	Medium	Hazard Identification

The Problem

- Problem: plan complex sequences of behaviours that achieve surveillance goals

- Challenges:
  - ✓ unpredictable environment
  - ✓ little stability
  - ✓ information changes quickly
  - ✓ rapid responses required
  - ✓ limited resources
  - ✓ mixed-initiative framework
  - ✓ multi-agent framework

- Forward planning required but time-consuming and resource intensive

Our Approach

Automated Plan-Based Policy-Learning

- **Goal:** Learning efficient, light-weight policies for the high-level operation of multiple intelligent vehicles engaged in surveillance in highly dynamic situations

- **Method:** Monte Carlo Sampling

  Assumption: time/resources available offline to train policies

  1. Sample many instances of the stochastic problem
  2. Solve each instance using a high-performing planner
  3. Apply a classifier to learn a policy as a mapping from states to actions, using the set of solutions as input

Challenges

1. **Planning:**
   - Formulate surveillance tasks formulated as deterministic planning problems
   - Plan high quality solutions
   - Planner capable of reasoning with temporal, spatial and continuous constraints
2. **Policy Learning:**
   - Use classification process to obtain policy from thousand of generated plans
   - Find good set of features to obtain robust policies
3. **Policy Integration and Switching:**
   - Use mechanism to integrate policies executed by different observers, with the help of human operators
   - Switch between policies learned for sub-tasks

Search And Tracking (SAT)

- **Target:** vehicle being sought and tracked through a mixed urban, suburban and rural landscape

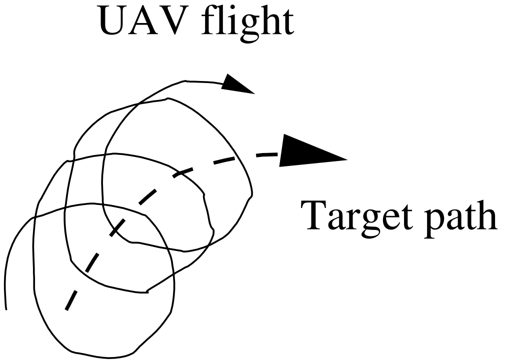
- **Observer:** single fixed-wing UAV
- Goal: follow target to its destination
- **SAT mission:** 2 phases
  1. **Tracking:** UAV flies in a standard pattern over the target, observing its progress
  2. **Search:** UAV has lost the target and flies a series of manoeuvres intended to rediscover the target
- 2 phases interleave until target stops or is lost irrevocably

Our Assumptions

- UAV **controllers** are robust and provide basic flight control to maintain level flight, turning and localising
- UAV equipped with **imaging systems** to observe the target and observation is susceptible to error and interference
- The imager has two modes: **wide-angle** mode (180°) and **narrow-angle** mode (90°)
- A faster moving target in the viewing zone is considered easier to spot
- The target follows the road networks and does **not** perform significant **evasive** actions

Tracking

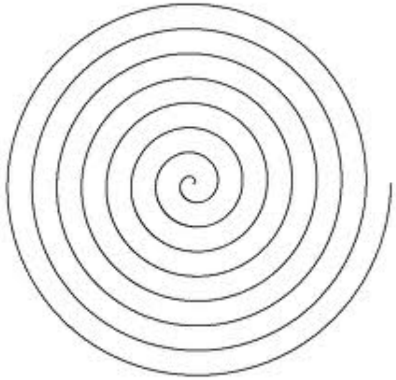
- Tracking is managed by a reactive controller
- Flight path of a fixed-wing UAV in tracking mode is a circle of fixed radius centred on the target
- Radius depends on the capabilities of the UAV (range of the imaging equipment and turning radius of the UAV)
- As the target moves, the circle moves with it, so the flight path of the UAV is a spiralling pattern over the ground



Searching

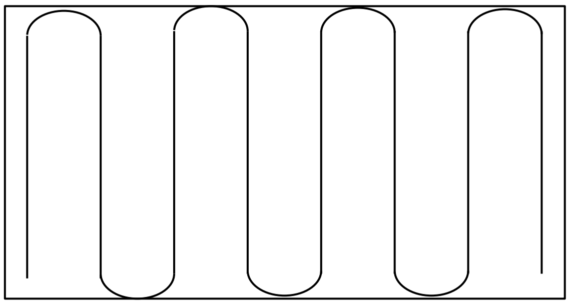
- What happens if the observer looses the target?
- To rediscover the target, search needs to be directed into specific places
- UAV exploits standard search patterns to find the target:

1. Spiral pattern:



To cover areas of high density road network

2. Lawnmower pattern:



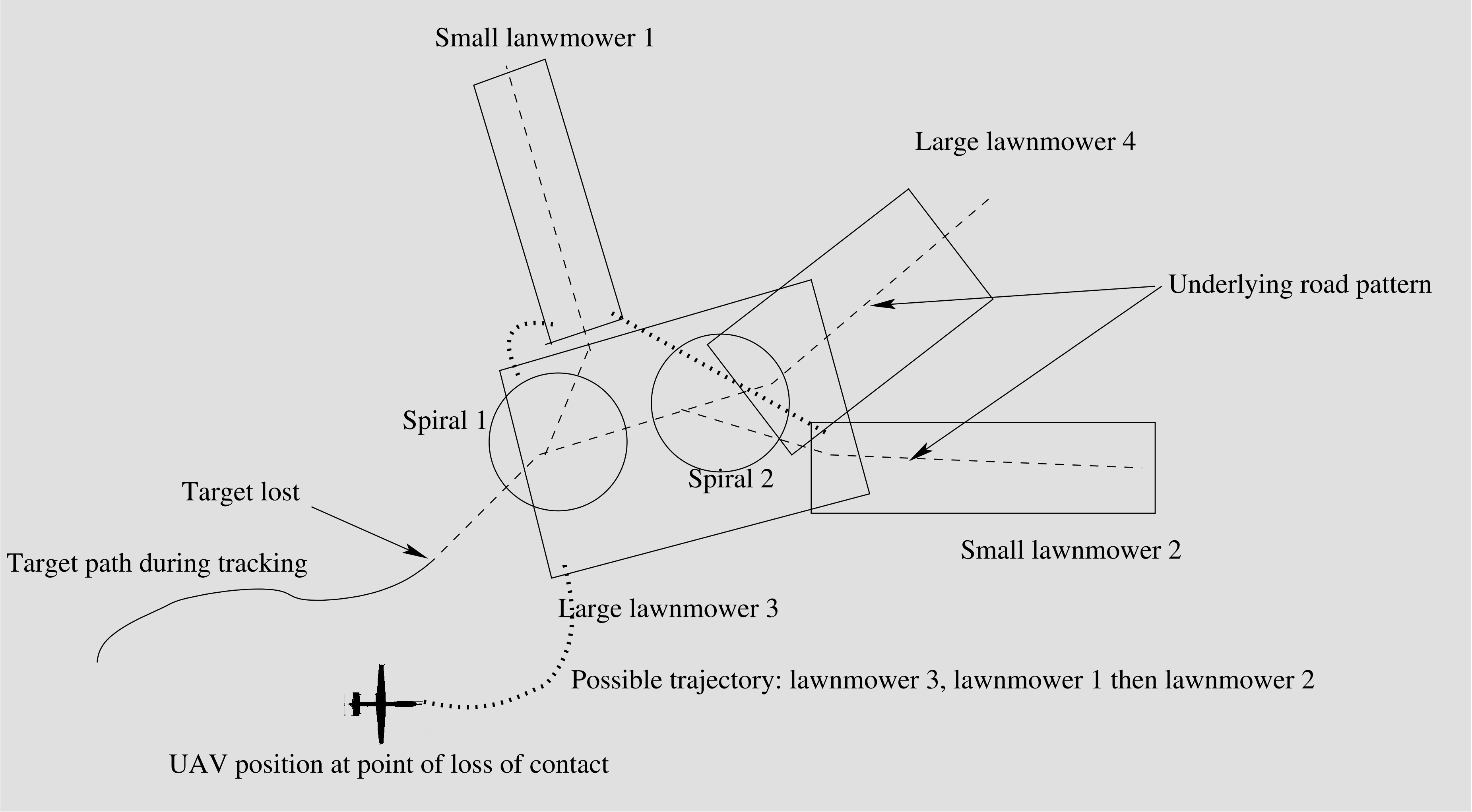
To cover elongated areas as major roads

Searching

How do we distribute the search patterns over the terrain in order to maximize the likelihood of rediscovering the target?

Search as a Planning Problem

The selection of patterns can be seen as a planning problem



Planning Domain

```
(:durative-action doSpiral
 :parameters (?from ?to - waypoint ?p - spiral )
 :duration (=?duration (timefor ?p))
 :condition (and (at start (beginAt ?from ?p))
           (at start (endAt ?to ?p))
           (at start (at ?from)) (at end (active ?p)))
 :effect (and (at end (at ?to)) (at start (not (at ?from)))
        (at end (increase (reward) (rewardof ?p)))))
```

```
(:durative-action fly
 :parameters (?from ?to - waypoint)
 :duration (= ?duration (distance ?from ?to))
 :condition (and (at start (at ?from)))
 :effect (and (at start (not (at ?from)))(at end (at ?to))))
```

```
(:durative-action doSmallLawnMower
 ...)
```

Planning Problem

- **Goal:** the problem has no goal, but the plan metric measures the value of the plan in terms of the accumulated expectation of finding the target

```
(:goal (and (>= (reward) 1)))
(:metric maximize (reward)))
```

- **Initial state:**

  1. Identify candidate search patterns
     -> difficult because infinite many patterns
  2. Assign appropriate rewards to them
     -> difficult because we lack knowledge about the intentions of the target

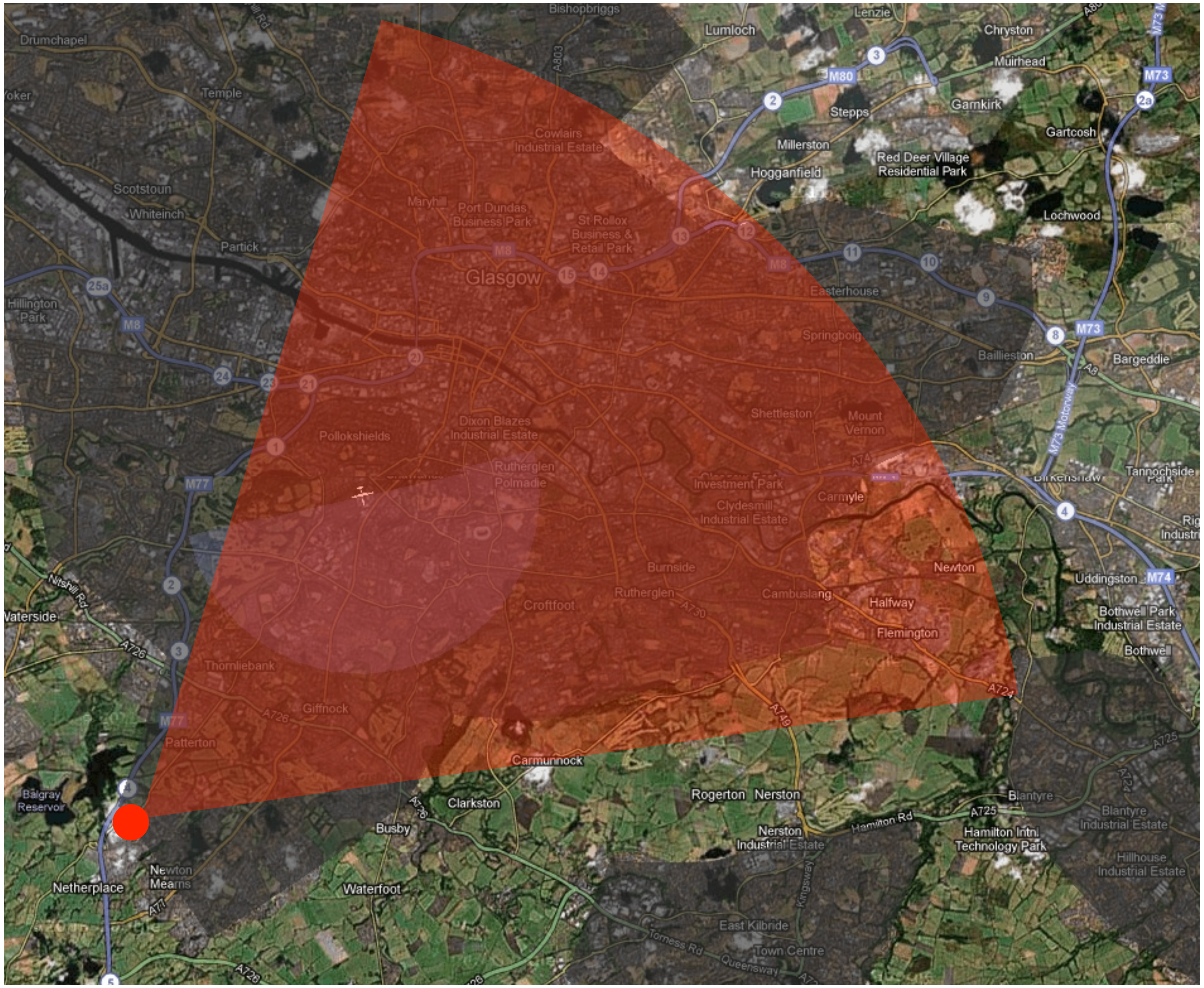
Search Pattern Generation

1. **Search area selection:** we first select a subset of the general search area in which the target is more likely to be rediscovered

2. **Sampling:** we sample points in the selected area using a probability distribution laid over the area based on:
   - Density of roads across the area
   - Terrain type
   - Distance from the last known location of the target

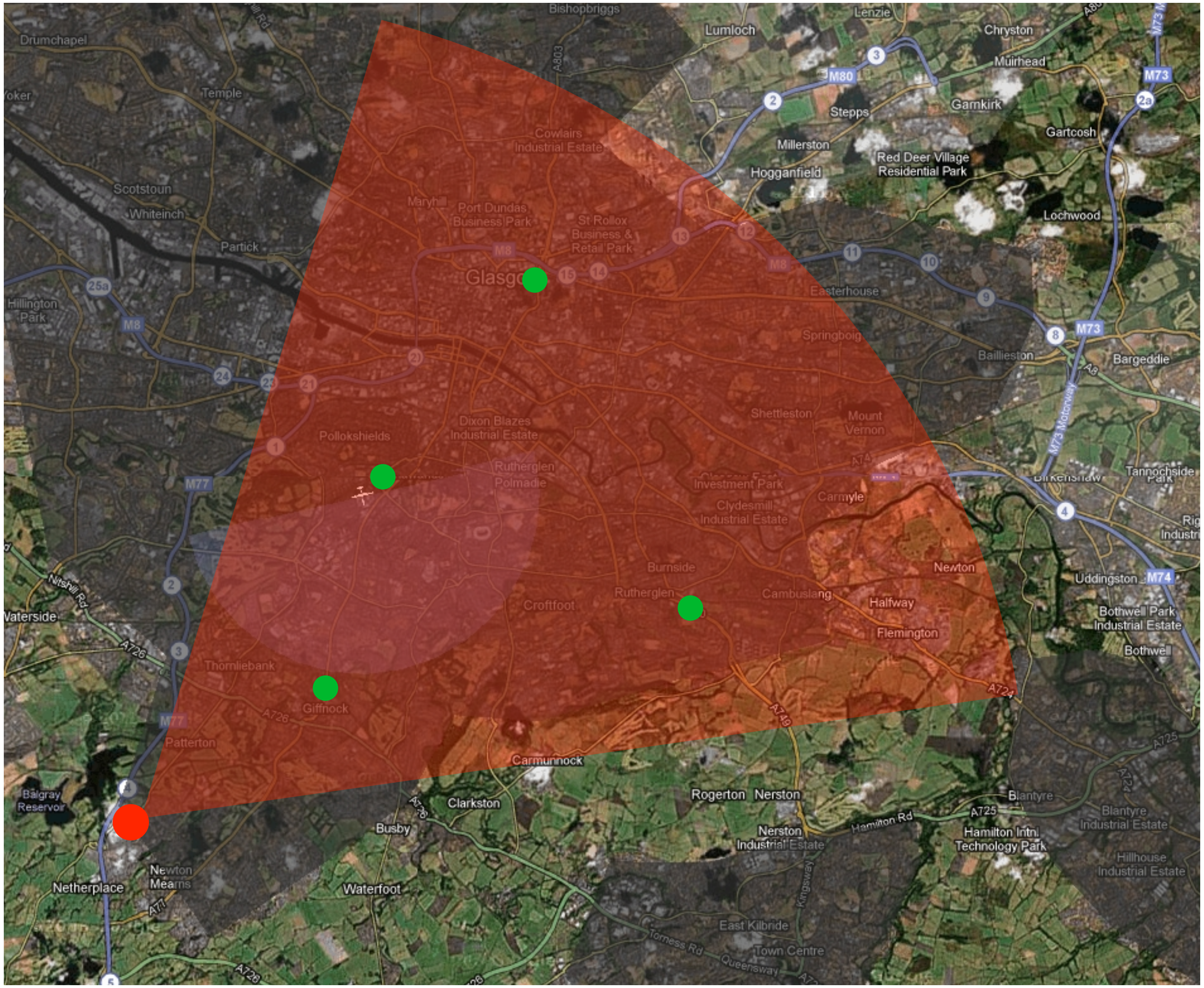
3. **Search Pattern Generation:** we decide the type of pattern to use for each point
   - Spirals: in the search area closest to the origin
   - Lawnmowers: in rural areas or areas of lower road density

Search Pattern Generation



Search Pattern Generation

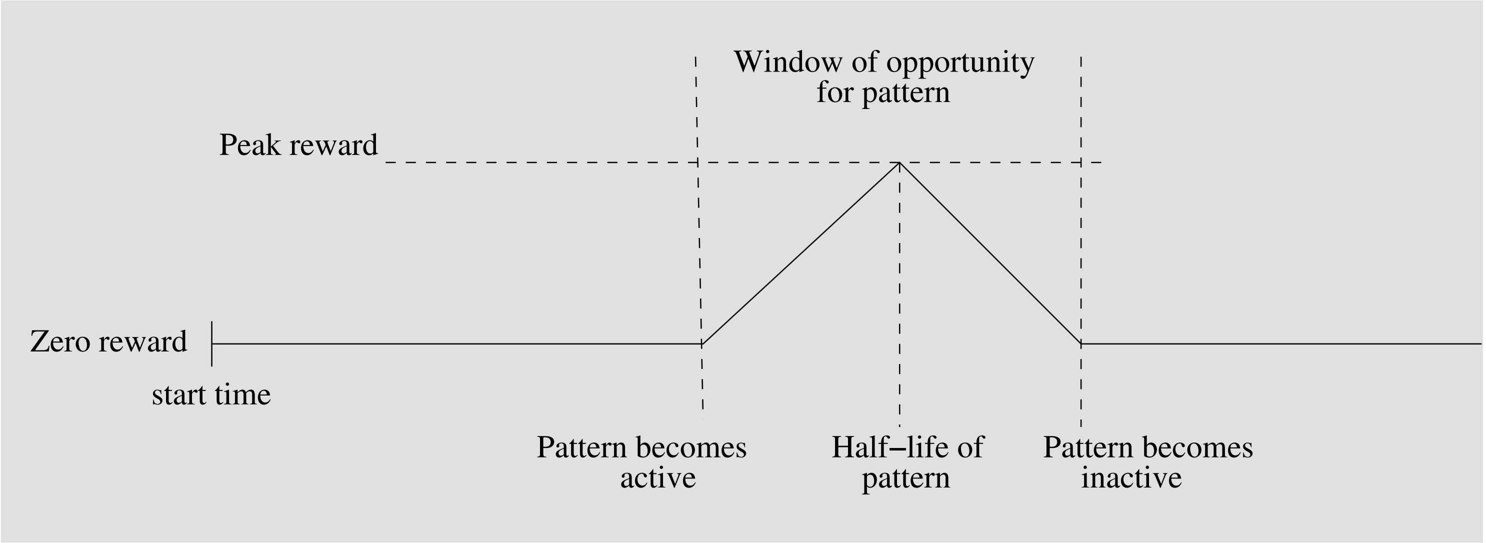
Search Pattern Generation



Search Pattern Generation

Reward Assignment

- We associate with each pattern a reward by using a time-dependent function



- Function shape: approximate lifted Gaussian distribution

- This time-dependent reward function is managed by timed-initial fluents in the problem specification

Planning Instance

Initial State

Time point 482
(Time initial literal)

```
(= (rewardOf spiral1) 866)
(at 482 (= (rewardOf spiral1) 1732))
(at 541 (= (rewardOf spiral1) 866))
(= (timefor spiral1) 299)
(beginAt s1s spiral1)
(endAt s1e spiral1)
(at 189 (active spiral1))
(at 629 (not (active spiral1)))
(= (distance origin s1s) 31)
(= (distance s1e s1s) 36)
(= (distance s1e s2s) 27)
...
```

Goal

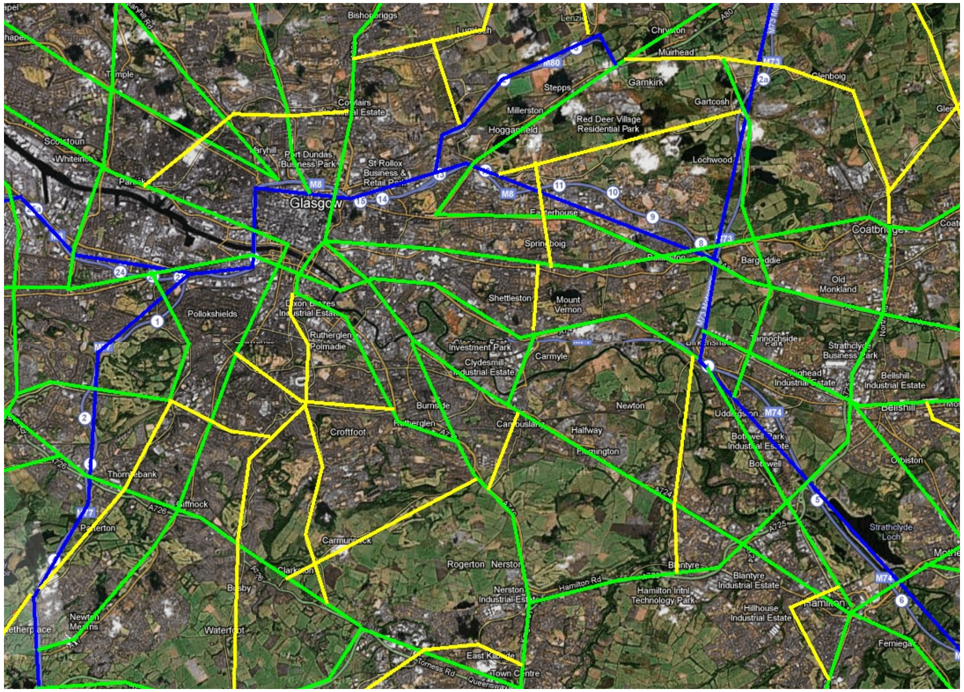
```
(:goal (and (>= (reward) 1)))
(:metric maximize (reward)))
```

Planner

- **OPTIC:** anytime cost-improving search (Benton, Coles, & Coles 2012)
- Time-bounded search of 10 seconds given that we are in a time-critical situation
- The planner finds a first solution very easily, but it then improves on this by adding further search patterns to the plan, or trying different collections of patterns
- On average, OPTIC produces around 6 plans in its 10 second window per problem instance
- **Static policy:** we currently "freeze" time while the observer is planning

SAT Simulation

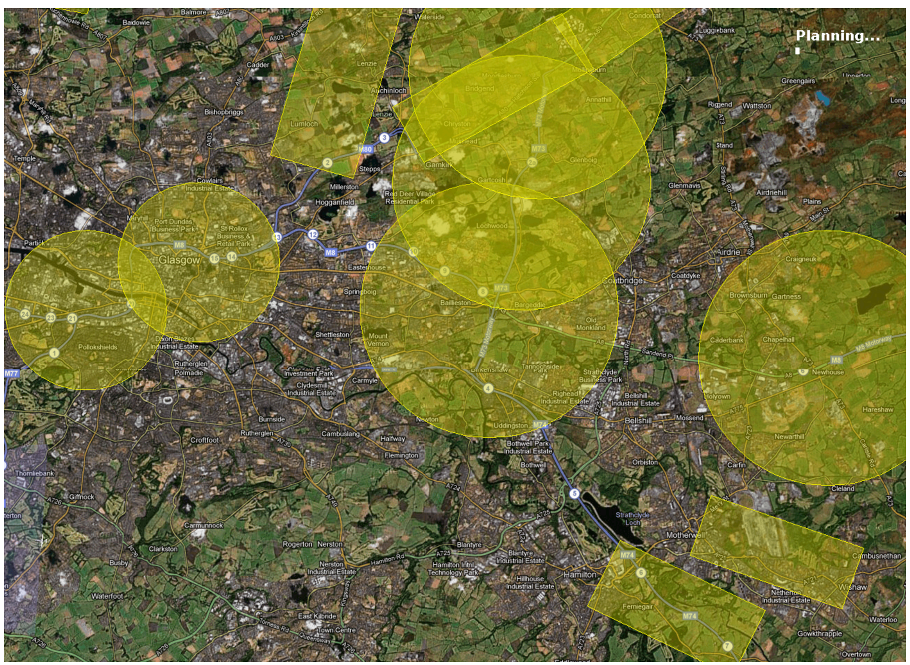
- Abstraction: we do not consider control of flight surfaces or altitude
- The area of operations is a part of Scotland of about 100 Km square with Glasgow and Edinburgh approximately defining its lower corners



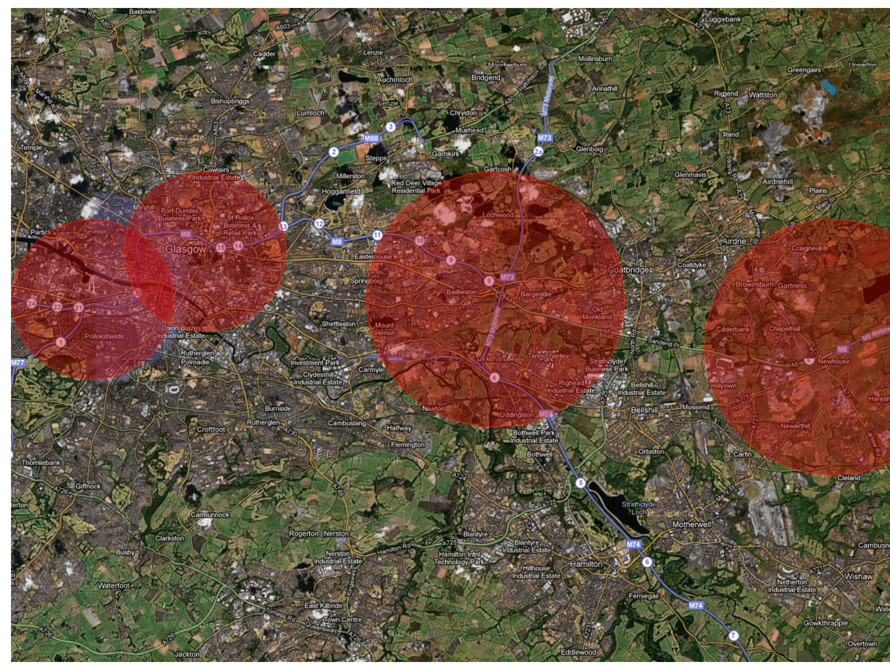
SAT Simulation

- **The target follows a path acquired using Google Maps, using a selected (configurable) origin and destination**
- **Simulation integrates the planner and displays the stages of the planning process**

Initial State



Plan



Experimental Results

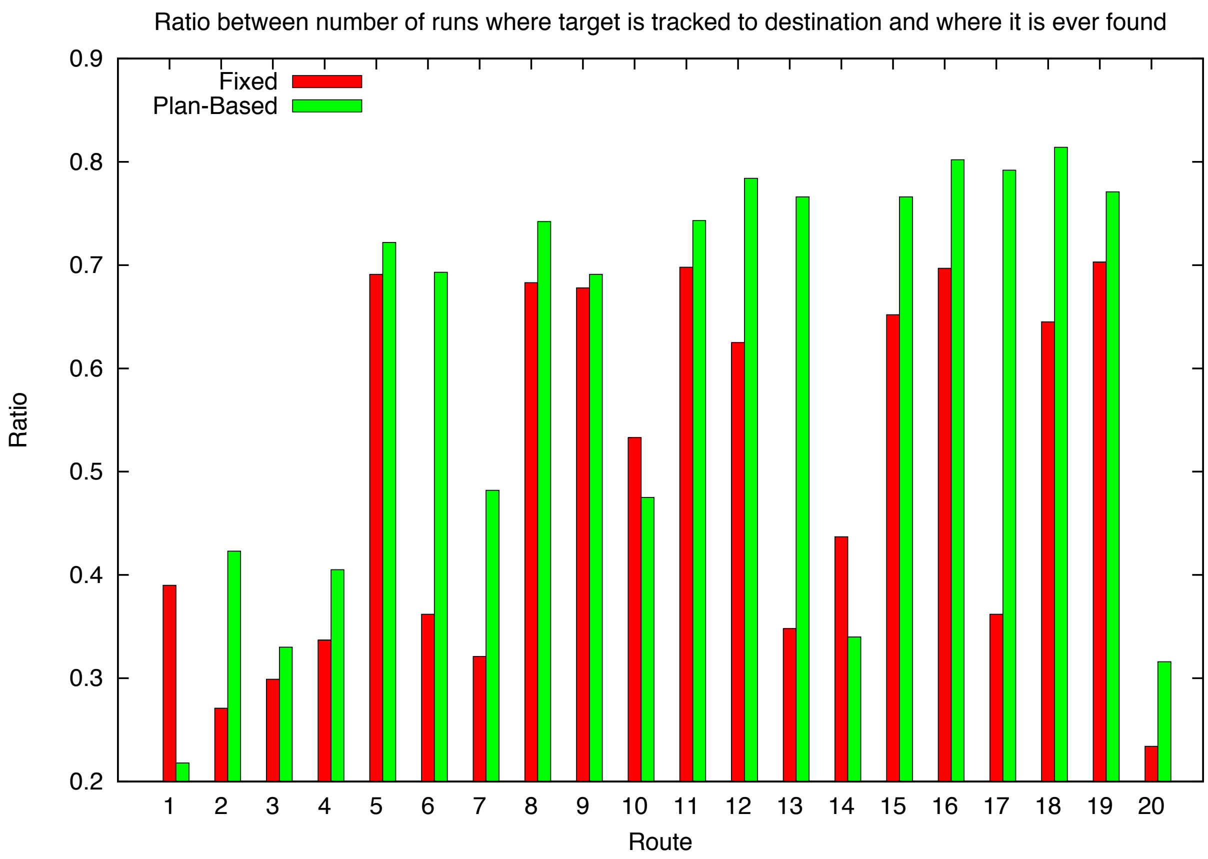
- Compare our plan-based policy with a fixed policy

- UAV fixed policy:
  - Track the target until in sight
  - Continue to track the predicted location of the target for about 3 minutes after target is lost
  - Execute fixed sequence of patterns:
    1. Spiral search around the point where it lost the target
    2. Large lawnmower pattern over a 20 km square area

- 20 routes -- We executed the simulation on each route 1000 times for each of the 2 strategies (40000 runs)

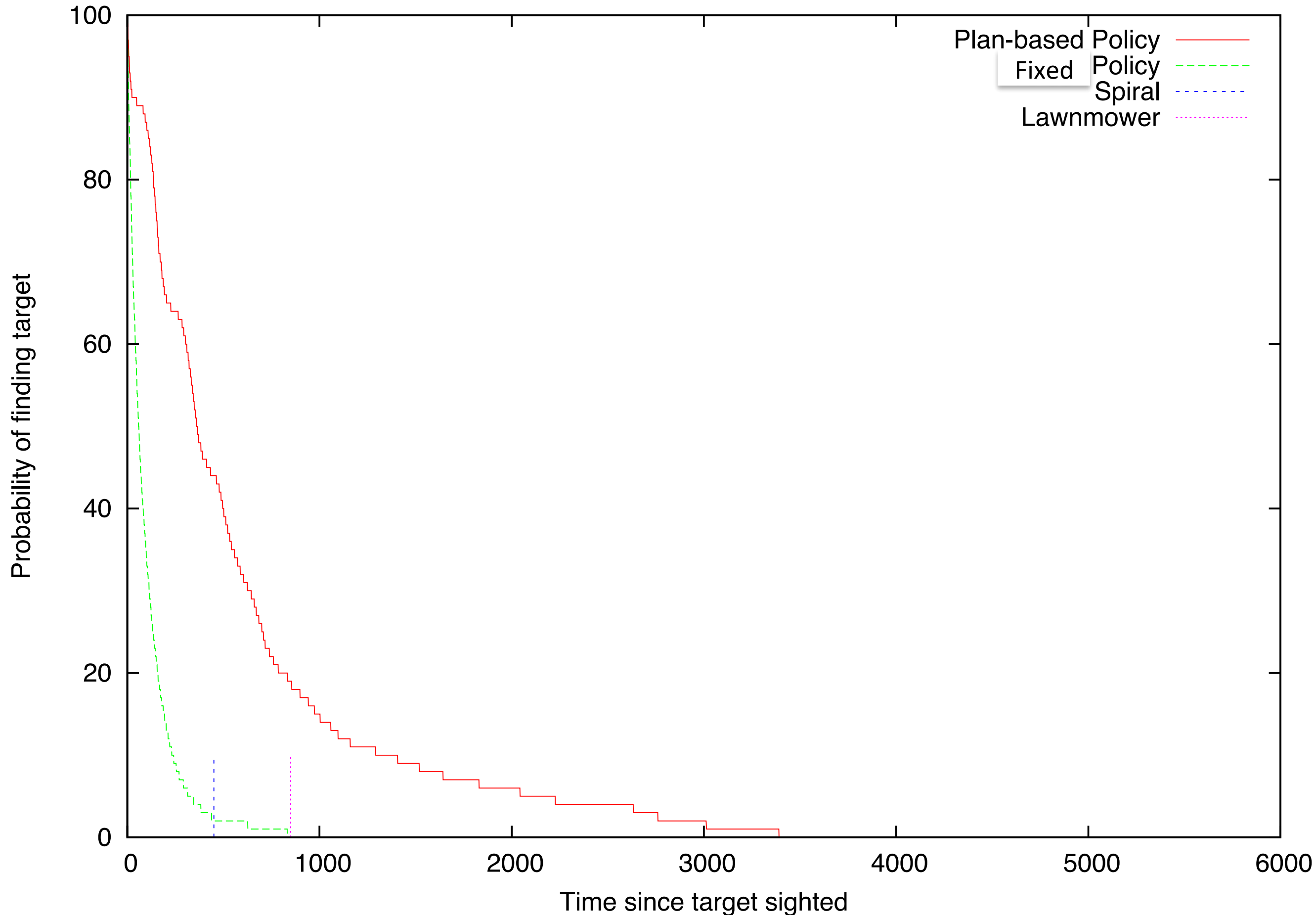
Experimental Results

- **Metric to evaluate the strategies: proportion of runs in which the target is tracked to its destination**



Experimental Results

- Metric to evaluate the strategies: probability distribution over time of relocating the target after losing it, considering only cases in which it was successfully relocated



Extensions

We are currently working on extending our approach to SAT in two main directions:

1. **Include the effects of obstacles that prevent the UAV from tracking the target**

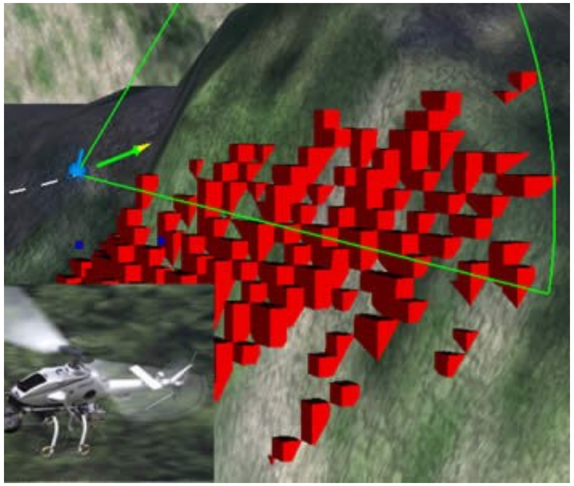
2. **Model and handle qualitative spatial constraints**

Obstacles

- Obstacles: regions that the UAV cannot violate or regions that the UAV can enter with an associated risk

    1. Urban development
    2. Terrain (vegetation, mountains, etc.)
    3. Prohibited or restricted areas (“no-fly zones”)
    4. Adverse weather conditions (rain, showers, ...)

- Different characteristics:
    - hard / soft
    - static / dynamic
    - permanent / temporary
    - see-through / opaque
    - big / smalls



- We model obstacles as convex polygons

Issues with Obstacles

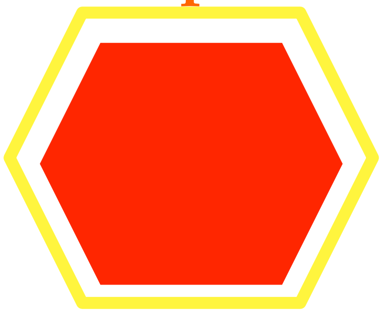
1. **Tracking with obstacles: how to react to the target entering an area that is an obstacle for the UAV, while the UAV is in tracking mode**

2. **Search pattern generation with obstacles: how to treat obstacles during the pattern generation phase**

3. **Navigation with obstacles: how to navigate between obstacles**

Pattern Generation with Obstacles

2. How to treat obstacles during the pattern generation phase
   i. Generate patrol search pattern around each obstacle



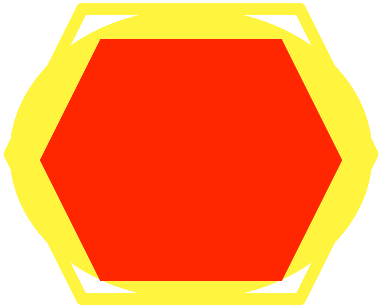
   ii. Generate side-patrol search patterns along most probable exit sides of the obstacles

   iii. Generate spiral and lawnmower patterns without considering obstacles but, if pattern intersects with the obstacle, modify pattern

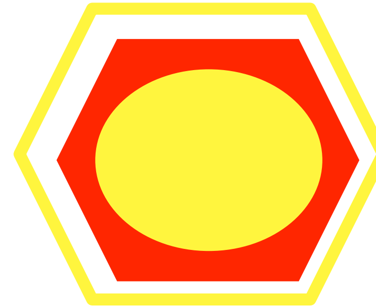
Sara Bernardini

Pattern Generation with Obstacles

Case 1: **Pattern almost overlaps obstacle**

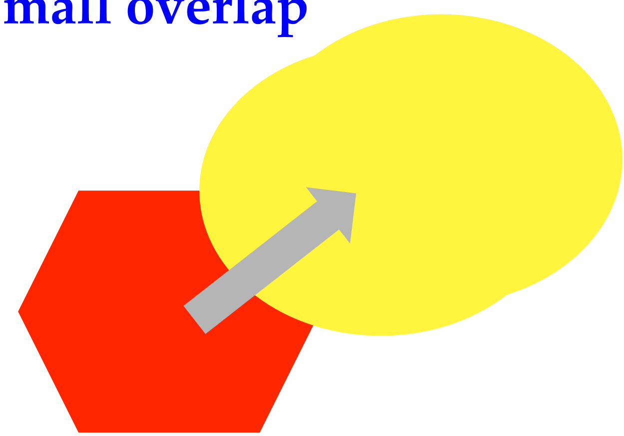


Case 2: **Pattern inside obstacle**



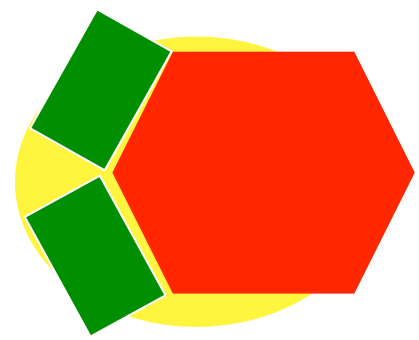
Remove pattern and create patrol pattern around obstacle

Case 3: **Small overlap**



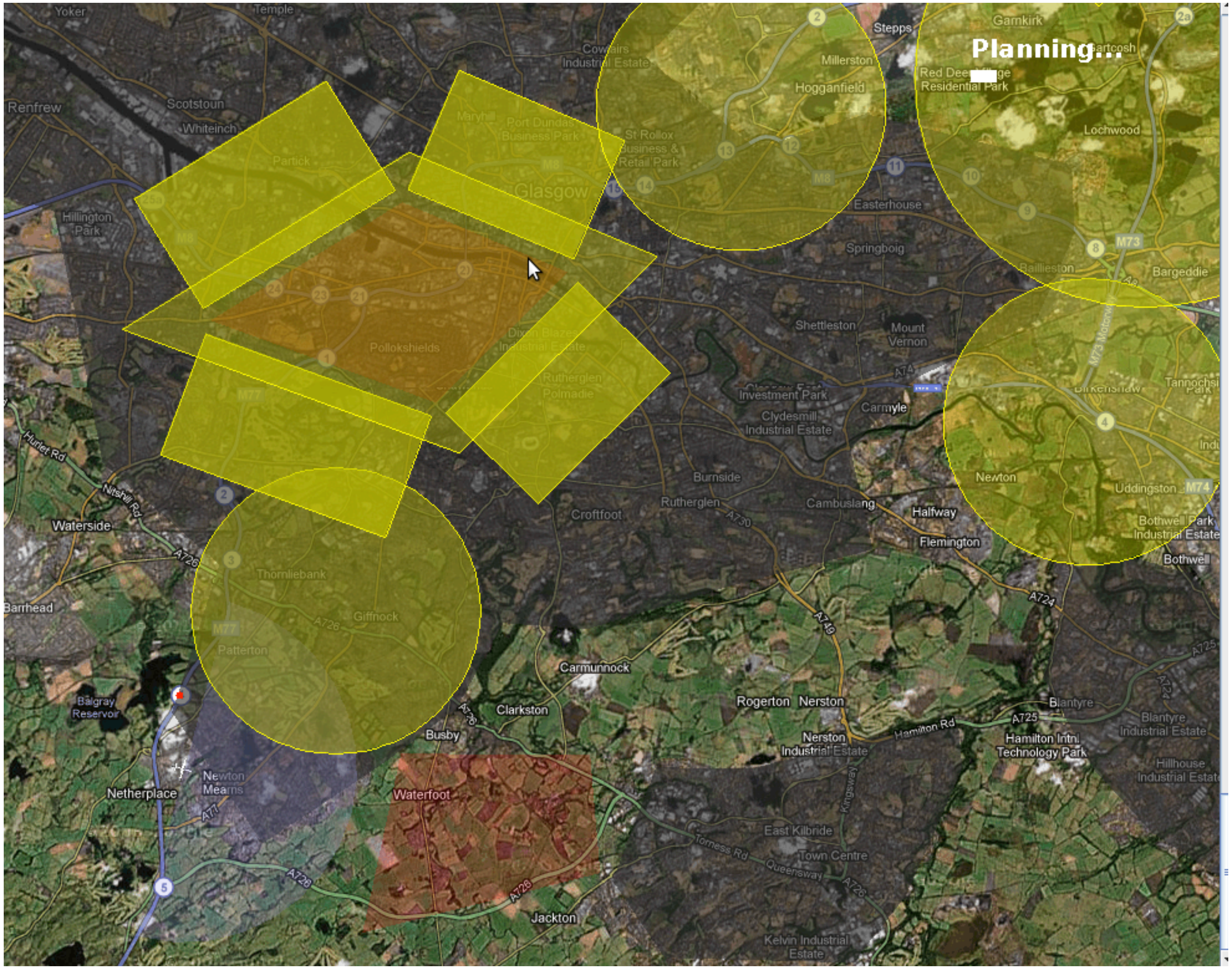
Find minimum translation vector and translate pattern

Case 4: **Big overlap**



Replace original pattern with new smaller patterns covering the area close to the obstacle

Simulation with Obstacles



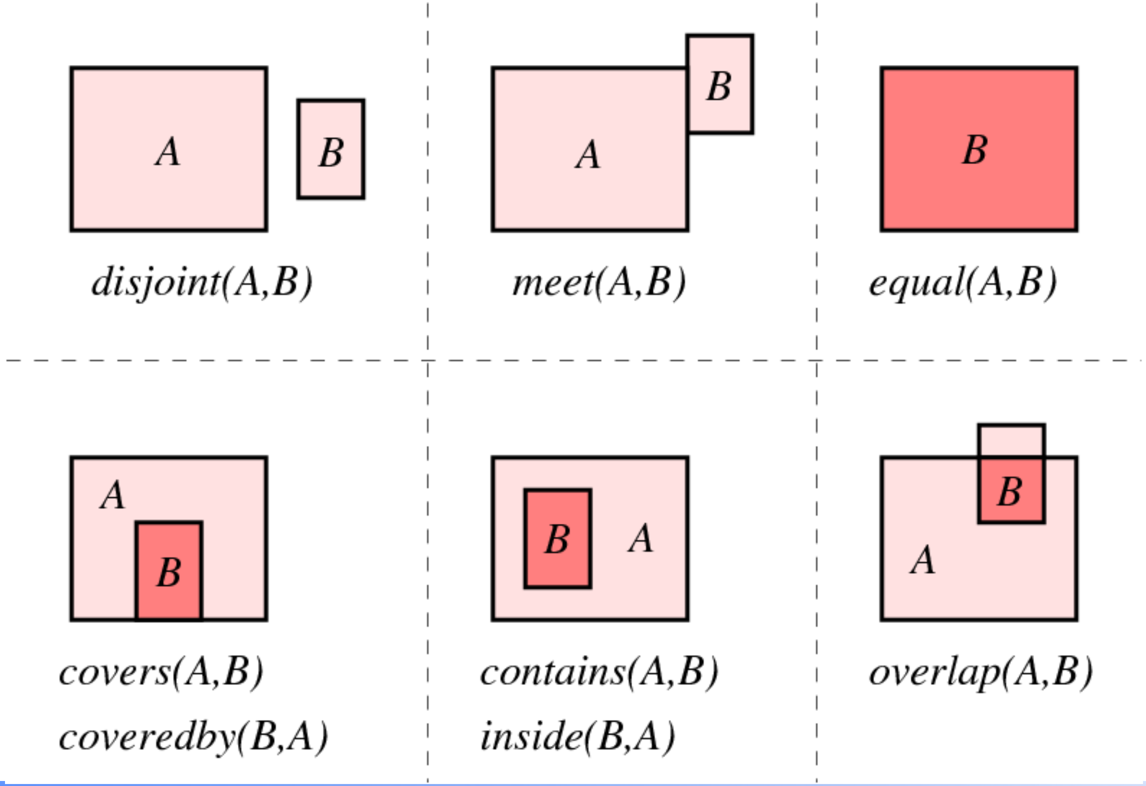
Spatial Reasoning

- Our scenario involves polygons to represent obstacles, spirals and rectangles to represent search patterns, segments to represents roads
- Currently spatial constraints cannot be modelled in PDDL, neither they can be handled by any state-of-the-art planner
- We aim to extend our planning technology to cope with these novel spatial features
- Qualitative Spatial Reasoning
- Region Connection Calculus (RCC)
  (Randell, Cui, & Cohn 1992)
- We plan to use a specialised external solver to manage qualitative spatial constraints (e.g. GQR for RCC)

Region Connection Calculus

- First order logic to represent topological relationships between regions of space
- One primitive relation: given two spatial regions x and y, the relation C(x, y) means that x and y are connected

RCC8



Challenges

1. Planning:
   - Formulate surveillance tasks formulated as deterministic planning problems
   - Plan high quality solutions
   - Planner capable of reasoning with temporal, spatial and continuous constraints
2. Policy Learning:
   - Use classification process to obtain policy from thousand of generated plans
   - Find good set of features to obtain robust policies
3. Policy Integration and Switching:
   - Use mechanism to integrate policies executed by different observers, with the help of human operators
   - Switch between policies learned for sub-tasks

Conclusions

- SAT: search problem as planning problem in which search patterns must be selected and sequenced to maximise the expectation of rediscovering the target
- A number of benefits:
  - ✓ Behaviour of UAV following a plan is predictable and well understood
  - ✓ A plan can be used as a common medium of exchange between the UAV and human observers
  - ✓ Use of generic planning provides flexibility
- Long term research plan: plan-based policy learning
- Benefit: providing a rapid response capability in situations where agents must decide quickly between alternative actions which are expensive to evaluate in detail