Artificial Intelligence in Robotics

Lecture 11: Patrolling

Viliam Lisý

Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Eng.
Czech Technical University in Prague

Mathematical programming

LP

$$\begin{array}{ll} \text{maximize} & \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array}$$

MILP

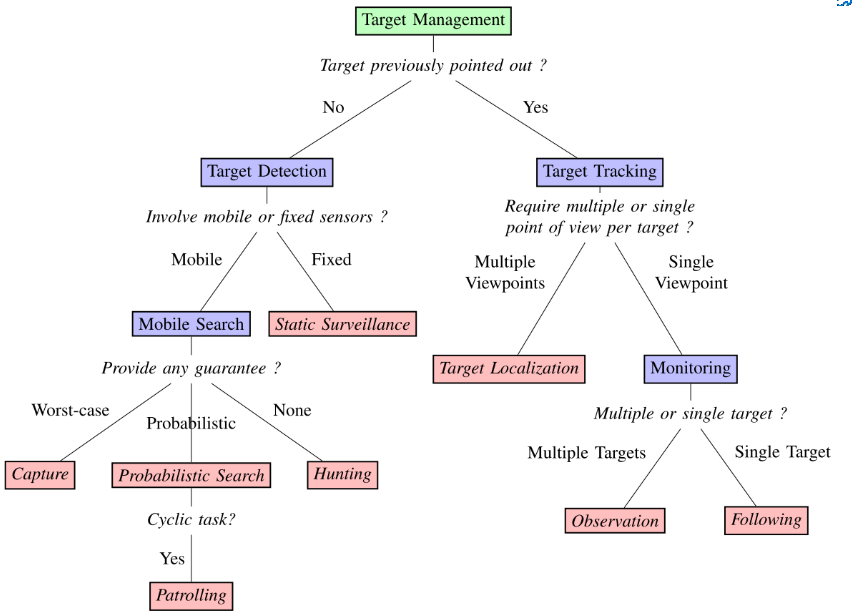
- Some of the variables are integer
- Objective and constraints are still linear

Convex program

- Optimize a convex function over a convex set

Non-convex program

Task Taxonomy



Target Management

Target previously pointed out ?

- No: Target Detection
  - *Involve mobile or fixed sensors ?*
  - Mobile: Mobile Search
    - *Provide any guarantee ?*
    - Worst-case: *Capture*
    - Probabilistic: *Probabilistic Search*
      - *Cyclic task?*
      - Yes: *Patrolling*
    - None: *Hunting*
  - Fixed: *Static Surveillance*
- Yes: Target Tracking
  - *Require multiple or single point of view per target ?*
  - Multiple Viewpoints: *Target Localization*
  - Single Viewpoint: Monitoring
    - *Multiple or single target ?*
    - Multiple Targets: *Observation*
    - Single Target: *Following*

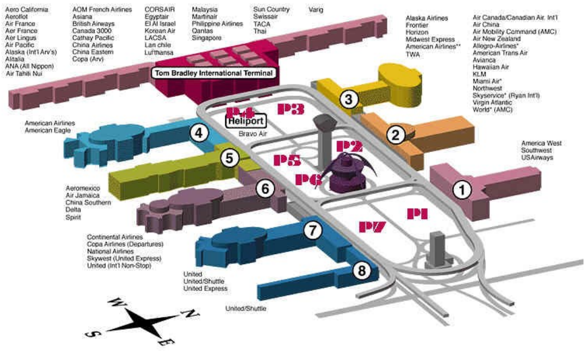
Robin, C., & Lacroix, S. (2016). Multi-robot target detection and tracking: taxonomy and survey. Autonomous Robots, 40(4), 729–760.

Resource allocation games

Developed by team of prof. M. Tambe at USC (2008-now)

In daily use by various organizations and security agencies

Resource allocation games



	①	②	③	④	⑤	⑥	⑦	⑧	-15
Unprotected	10	11	9	15	11	15	14	6	-14
Protected	5	4	5	7	6	5	7	3	-11
Optimal strategy	0	0.14	0	0.62	0.2	0.49	0.56	0	-10

Resource allocation games

Set of targets: $T = t_1, \dots, t_n$

Limited (homogeneous) security resources $r \in \mathbb{N}$

- Each resource can fully protect (cover) a single target

The attacker attacks a single target

Attacker’s utility for covered/uncovered attack: $U_a^c(t) < U_a^u(t)$

Defender’s utility for covered/uncovered attack: $U_d^c(t) > U_d^u(t)$

Stackelberg equilibrium

the leader (l) – publicly commits to a strategy

the follower (f) – plays a best response to leader

$$\arg \max_{\sigma_l \in \Delta(A_l);\, \sigma_f \in BR_f(\sigma_l)} r_l(\sigma_l, \sigma_f)$$

Example

	L	R
U	(4,2)	(6,1)
D	(3,1)	(5,2)

Why?

- The defender needs to commit in practice (laws, regulations, etc.)
- It may lead to better expected utility
    - NE: (U,L) -> 4; Pure SE: (D,R) -> 5; Mixed SE ~ 5.5

Mixed Stackelberg equilibrium

	L	R
U	(4,2)	(6,1)
D	(3,1)	(5,2)

Strong Stackelberg Equilibrium

- Follower breaks ties in favor of the leader (0.5; 0.5) -> 5.5
- Form many settings can be motivated by infinitesimal deviation

Weak Stackelberg Equilibrium

- Follower breaks ties worst for the leader (0.5; 0.5) -> 3.5
- The equilibrium may not exist, because smaller motivation is better

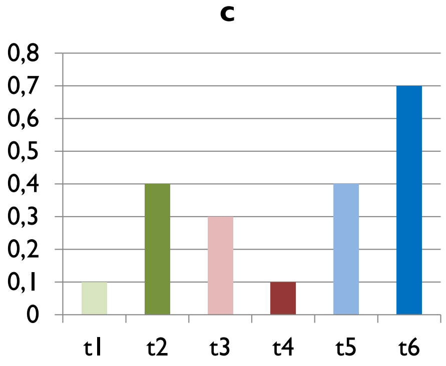
Solving resource allocation games

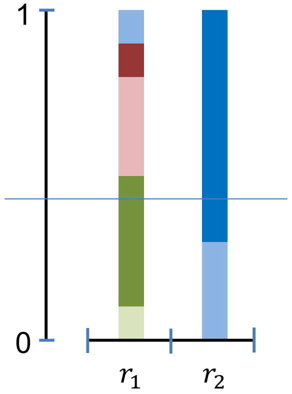
Kiekintveld, et al.: Computing Optimal Randomized Resource Allocations for Massive Security Games, AAMAS 2009

Only coverage vector c_t matters, Z is a sufficiently large number

$$
\begin{array}{rcl}
\max & d & \\
a_t \in & \{0,1\} & \forall t \in T \\
\sum_{t \in T} a_t = & 1 & \\
c_t \in & [0,1] & \forall t \in T \\
\sum_{t \in T} c_t \leq & m & \\
d - U_\Theta(t,C) \leq & (1-a_t) \cdot Z & \forall t \in T \\
0 \leq k - U_\Psi(t,C) \leq & (1-a_t) \cdot Z & \forall t \in T
\end{array}
$$

Sampling the coverage vector



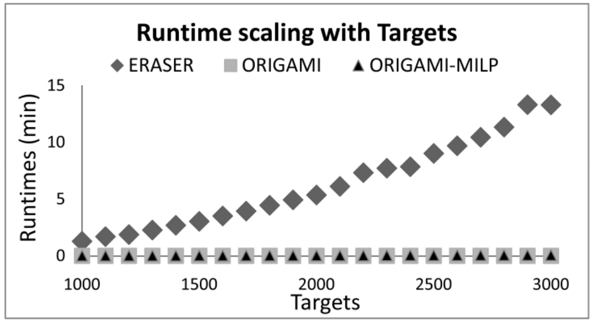


Scalability

25 resources, 3000 targets => 5×10^{61} defender's actions

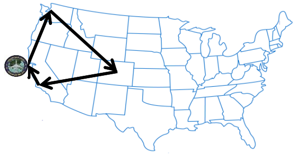
no chance for matrix game representation

The algorithm explained above is ERASER



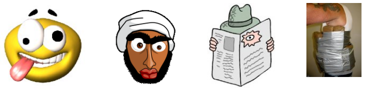
Studied extensions

Complex structured defender strategies



Probabilistically failing actions

Attacker's types



Resource types and teams

Bounded rational attackers

Resource allocation (security) games

Advantages

- Wide existing literature (many variations)
- Good scalability
- Real world deployments

Limitation

- The attacker cannot react to observations (e.g., defender's position)

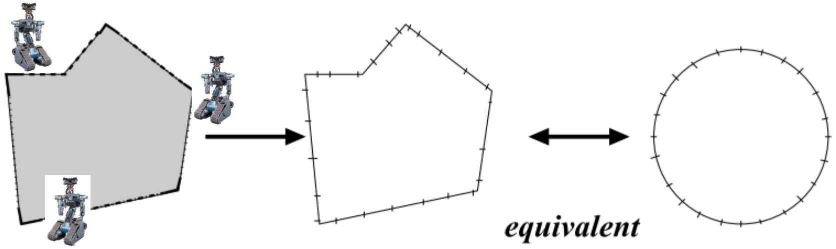
Perimeter patrolling

Agmon et al.: Multi-Robot Adversarial Patrolling: Facing a Full-Knowledge Opponent. JAIR 2011.

The attacker can see the patrol!

Perimeter patrolling

Polygon P, perimeter split to N segments



Defender has homogenous resources $k > 1$

- move 1 segment per time step
- turn to the opposite direction in τ time steps

Attacker can wait infinitely long and sees everything

- chooses a segment where to attack
- requires t time steps to penetrate

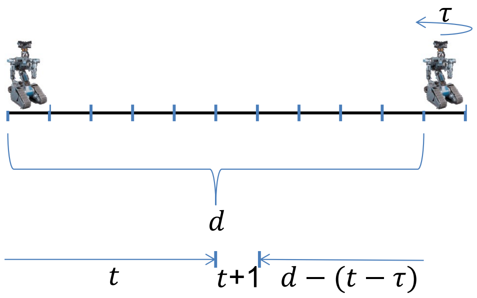
Interesting parameter settings

Let $d = \frac{N}{k}$ be the distance between equidistant robots

There is a perfect deterministic patrol strategy if $t \geq d$

- the robots can just continue in one direction

What about $t = \frac{4}{5} d$?



The attacker can guarantee success if $\mathrm{t} + 1 < \mathrm{d} - (\mathrm{t} - \tau) \Rightarrow t < \frac{d+\tau-1}{2}$

Optimal patrolling strategy

Class of strategies: continue with probability p, else turn around

Theorem: In the optimal strategy, all robots are equidistant and face in the same direction.

Proof sketch:

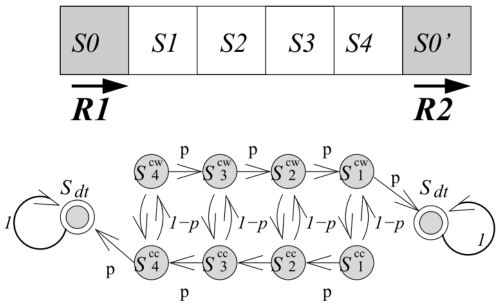
1. the probability of visiting the worst case segment between robots decreases with increasing distance between the robots
2. making a move in different directions increases the distance

Probability of penetration

For simplicity assume $\tau = 1$

Probability of visiting s_i at least once in next t steps

- = probability of visiting the absorbing end state from s_i
- sum of each direction visited separately



$$\begin{array}{c|ccccccccc}
 & S_1^{cc} & S_1^{cw} & S_2^{cc} & S_2^{cw} & S_3^{cc} & S_3^{cw} & S_4^{cc} & S_4^{cw} & S_{dt} \\
\hline
S_1^{cc} & 0 & 1-p & p & 0 & 0 & 0 & 0 & 0 & 0 \\
S_1^{cw} & 1-p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & p \\
S_2^{cc} & 0 & 0 & 0 & 1-p & p & 0 & 0 & 0 & 0 \\
S_2^{cw} & 0 & p & 1-p & 0 & 0 & 0 & 0 & 0 & 0 \\
S_3^{cc} & 0 & 0 & 0 & 0 & 0 & 1-p & p & 0 & 0 \\
S_3^{cw} & 0 & 0 & 0 & p & 1-p & 0 & 0 & 0 & 0 \\
S_4^{cc} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-p & p \\
S_4^{cw} & 0 & 0 & 0 & 0 & 0 & p & 1-p & 0 & 0 \\
S_{dt} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}$$

Probability of penetration

Algorithm 1 Algorithm FindFunc(d, t)

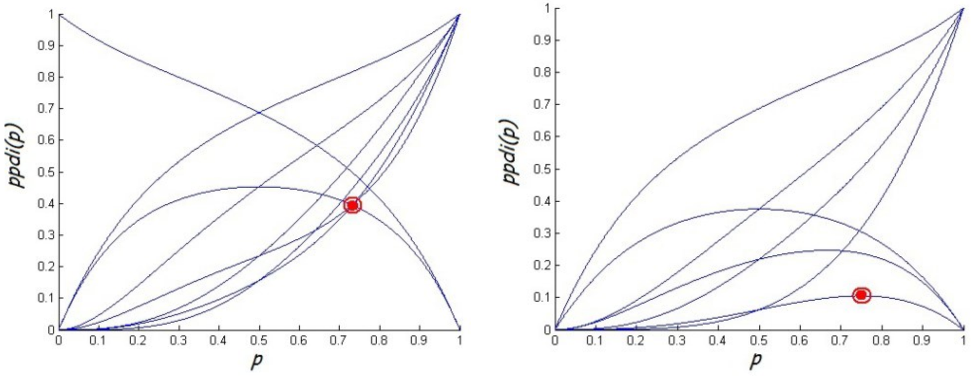
```
Create matrix $M$ of size $(2d+1)(2d+1)$, initialized with 0s
Fill out all entries in $M$ as follows:
$M[2d+1, 2d+1] = 1$
for $i \leftarrow 1$ to $2d$ do
    $M[i, \max\{i+1, 2d+1\}] = p$
    $M[i, \min\{1, i-2\}] = 1 - p$
Compute $MT = M^t$
$Res$ = vector of size $d$ initialized with 0s
for $1 \le loc \le d$ do
    $V$ = vector of size $2d+1$ initialized with 0s.
    $V[2loc] \leftarrow 1$
    $Res[loc] = V \times MT[2d+1]$
Return $Res$
```

All computations are symbolic. The result are functions $ppd_i: [0,1] \to [0,1]$ expressing the probability of penetration at i for a given probability of turn.

Optimal turn probability

Maximin value for p

Each line represents one segment (ppd_i)



Iterate all pairs of intersection and maximal points to find solution

- it is all polynomials

Perimeter patrol – summary

Split the perimeter to segments traversable in unit time

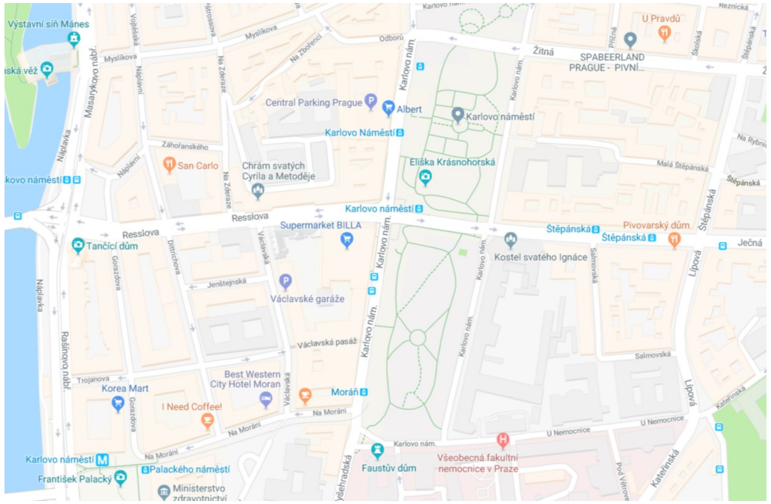
Distribute patrollers uniformly along the perimeter

Coordinate them to always face the same way

Continue with probability p turn around with probability $(1-p)$

Area patrolling

Basilico et al.: Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. AIJ 2012.



Area patrolling - Formal model

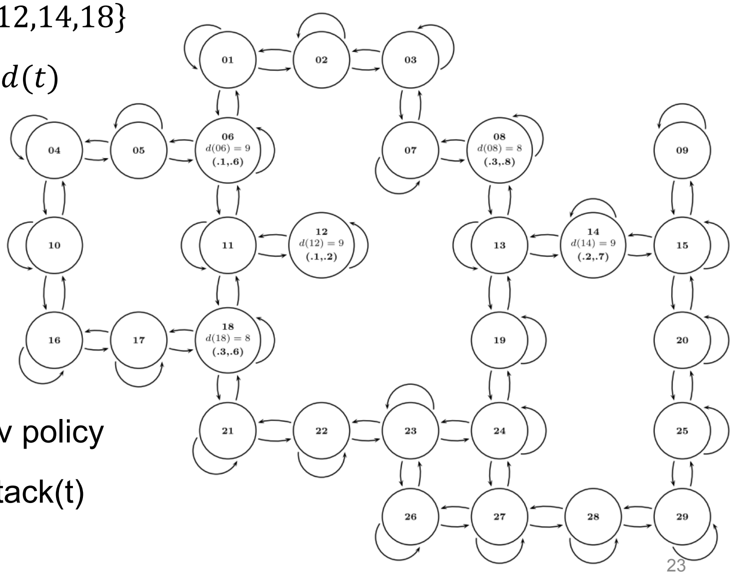
Environment represented as a graph

Targets $T=\{6,8,12,14,18\}$

Penetration time $d(t)$

Target values

$(v_d(t),v_a(t))$



Defender: Markov policy

Attacker: wait, attack(t)

Solving zero-sum patrolling game

We assume $\forall t \in T: \; v_a(t)=v_d(t)$

$a(i,j)=1$ if the patrol can move form i to j in one step; else 0

$P_c(t,h)$ is the probability of stopping an attack at target t started when the patrol was at node h

$\gamma_{i,j}^{w,t}$ is the probability that the patrol reaches node j from i in w steps without visiting target t

$$\max u$$

$$\alpha_{i,j} \geqslant 0 \quad \forall i,j \in V$$

$\alpha_{i,j}$ is the probability of moving from i to j

$$\sum_{j\in V} \alpha_{i,j} = 1 \quad \forall i \in V$$

$$u_{\mathbf{d}}(x) = \begin{cases} \sum_{i\in T} v_{\mathbf{d}}(i), & x = \textit{intruder-capture} \text{ or } \textit{no-attack} \\ \sum_{i\in T\setminus\{t\}} v_{\mathbf{d}}(i), & x = \textit{penetration-t} \end{cases}$$

$$\alpha_{i,j} \leqslant a(i,j) \quad \forall i,j \in V$$

$$\gamma_{i,j}^{1,t} = \alpha_{i,j} \quad \forall t \in T,\; i,j \in V \setminus \{t\}$$

$$\gamma_{i,j}^{w,t} = \sum_{x\in V\setminus\{t\}} \left(\gamma_{i,x}^{w-1,t}\alpha_{x,j}\right) \quad \forall w \in \{2,\ldots,d(t)\},\; t\in T,\; i,j \in V \setminus \{t\}$$

$$P_c(t,h) = 1 - \sum_{j\in V\setminus\{t\}} \gamma_{h,j}^{d(t),t} \quad \forall t \in T,\; h \in V$$

$$u \leqslant u_{\mathbf{d}}(\textit{intruder-capture}) P_c(t,h) + u_{\mathbf{d}}(\textit{penetration-t})\big(1-P_c(t,h)\big)$$

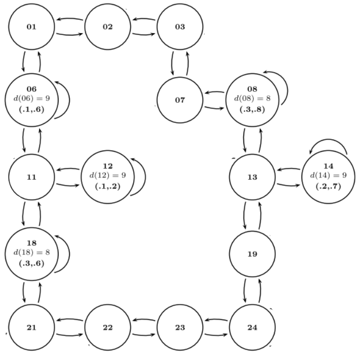
What type of optimization problem is this? LP? MILP? Convex?

Scaling up

No need to visits nodes not on shortest paths between targets

With multiple shortest paths, only the closer to targets is relevant

It is suboptimal to stay at a node that is not a target



Summary

GT can be applied to real world problems in robotics

Pursuit-evasion games
  - Perfect information capture
  - Visibility-based tracking
  - Patrolling
    - resource allocation
    - perimeter patrolling
    - area patrolling

AI (GT) problems can often be solved by transformation to mathematical programming

---

Resources

Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F. and Tambe, M. "Computing optimal randomized resource allocations for massive security games." AAMAS 2009.

Agmon, Noa, Gal A. Kaminka, and Sarit Kraus. "Multi-robot adversarial patrolling: facing a full-knowledge opponent." Journal of Artificial Intelligence Research 42 (2011): 887-916.

Basilico, Nicola, Nicola Gatti, and Francesco Amigoni. "Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder." Artificial Intelligence 184 (2012): 78-123.