Artificial Intelligence in Robotics

Lecture 11: Patrolling

Pavel Rytir
Artificial Intelligence Center
Department of Computer Science, Faculty of Electrical Engineering
Czech Technical University in Prague

Mathematical programming

- Linear programming
- Mixed integer programming
  - LP + some variables need to be an integer

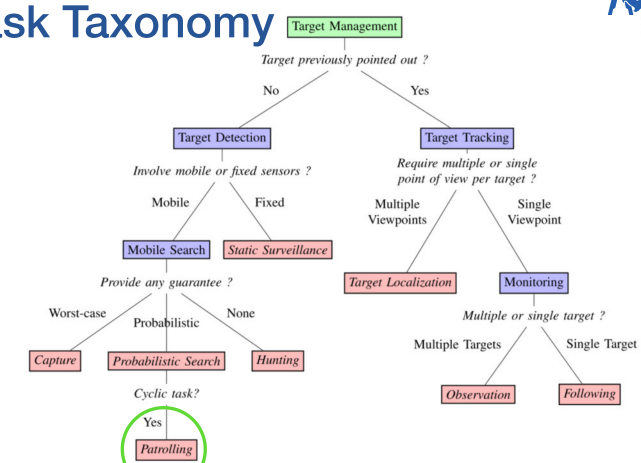
$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax \le b \\ \text{and} & x \ge 0 \end{array}$$

- Convex programing
  - f, g_i are convex
  - h_i are affine

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f(\mathbf{x}) & \\ \text{subject to} & g_i(\mathbf{x}) \le 0, & i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, & i = 1, \dots, p, \end{array}$$

- Non-convex programing
- Many solvers available

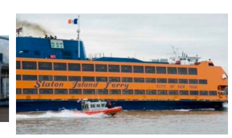
Task Taxonomy

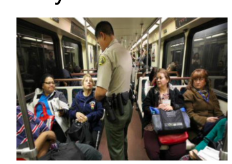


Robin, C., & Lacroix, S. (2016). Multi-robot target detection and tracking: taxonomy and survey. Autonomous Robots, 40(4), 729–760.

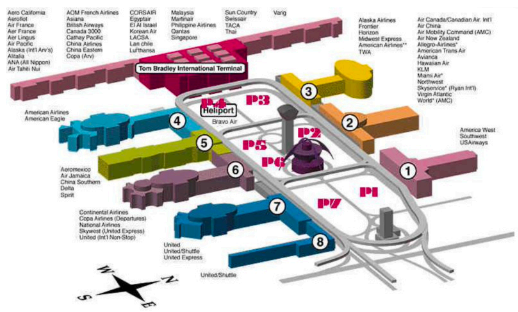
Resource allocation games

- Developed by team of prof. Milind Tambe at USC (2008-now)
  - Now at Harvard + Google Research India
  - Goal: Optimally use limited resources using randomization
- In daily use by various organizations and security agencies





Resource allocation games



Which parts of the terminal should be inspected by guards?

Stackelberg equilibrium

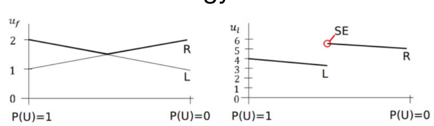
- the leader l – publicly commits to a strategy
- The follower(s) - play(s) a best response to the leader
- $\arg \max_{s_l \in \Pi(A_l), s_f \in \mathrm{BR}_f(s_l)} u(s_l, s_f)$
- The defender needs to commit in practice (laws, regulations, etc.)
- It may lead to better expected utility
- Useful for non-zero sum games

Stackelberg equilibrium

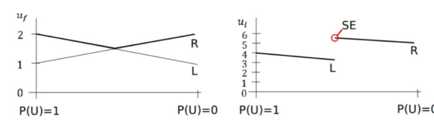
- Example

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

- (U, L) is an equilibrium. Payoff of row player is 4.
- If row player commits (credibly) to play D. (D, R) is also an equilibrium. Row players gets 5.
- Can row player get even more? Yes, if the leader can commit to a mixed strategy.



Stackelberg equilibrium



- The followers need to break ties in case there are multiple NE:
- arbitrary but fixed tie breaking rule
- Strong SE – the followers select such NE that maximizes the outcome of the leader (when the tie-braking is not specified we mean SSE),
- Weak SE – the followers select such NE that minimizes the outcome of the leader.
- Exact Weak Stackelberg equilibrium does not have to exist.
- The leader can often induce the favorable strong equilibrium by selecting a strategy arbitrarily close to the equilibrium that causes the the follower to strictly prefer the desired strategy

Resource allocation games
Compact security game model

- Set of targets: $T = \{t_1, \ldots, t_n\}$ - pure strategies of the attacker. One attacker.
- Limited (homogeneous) set of security resources $R = \{r_1, \ldots, r_m\}$. Each resource can fully protect (cover) a single target. $\binom{T}{m}$ - pure strategies of the defender. [Usually too big for normal form.]
- Attacker's utility for covered/uncovered attack: $U_\Psi^C(t) < U_\Psi^U(t)$
- Defender's utility for covered/uncovered attack: $U_\Theta^C(t) > U_\Theta^U(t)$
- Coverage vector $C = (C_{t_1}, \ldots, C_{t_n})$ - probabilities that a target is covered
- Attack vector $A = (A_{t_1}, \ldots, A_{t_n})$ - probabilities that a target is attacked

: **Example payoffs for an attack on a target.**

	Covered	Uncovered
Defender	5	−20
Attacker	−10	30

Resource allocation games
Compact security game model

- The defender's expected payoff given attack and coverage vectors is $U_\Theta(C,A) = \sum_{t \in T} a_t \cdot (c_t \cdot U_\Theta^c(t) + (1 - c_t) U_\Theta^u(t))$
- The expected payoff for an attack on target t, given C $U_\Theta(t,C) = c_t U_\Theta^c(t) + (1 - c_t) U_\Theta^u(t)$
- The attack set contains all targets that yield the maximum expected payoff for the attacker given coverage C $\Gamma(C) = \{t : U_\Psi(t,C) \geq U_\Psi(t',C) \, \forall \, t' \in T\}$

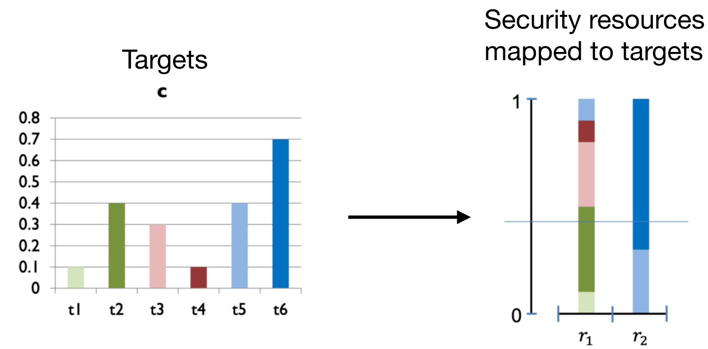
In a strong Stackelberg equilibrium, the attacker selects the target in the attack set with maximum payoff for the defender.

Resource allocation games
Compact security game model

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

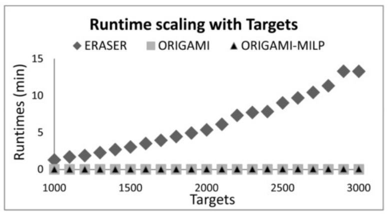
- Theorem. A pair of attack and coverage vectors (C,A) is optimal for the ERASER MILP correspond to at least one SSE of the game.
- Kiekintveld, et al.: Computing Optimal Randomized Resource Allocations for Massive Security Games, AAMAS 2009

The coverage vector



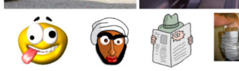
Scalability

- 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)

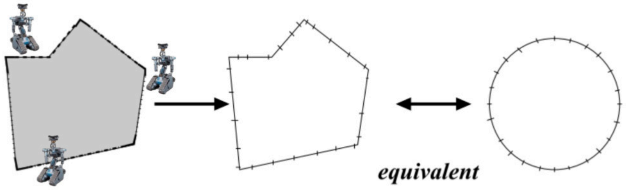
Perimeter patrolling

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

The attacker can see the patrol!

Perimeter patrolling

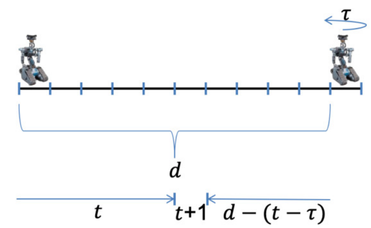
- Polygon P, perimeter split to N segments



- Defender has homogenous $k > 1$ mobile robots $R_1, \ldots, R_k$
  - 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 t be the duration of a penetration of a segment
- Let $d = \frac{n}{k}$ be the distance between equidistant robots
- There is a perfect deterministic patrol strategy if $t \geq d$
  - The robots just keep going in one direction
- What about $t = \frac{4}{5}d$?



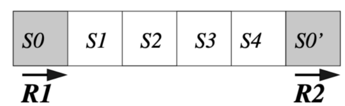
The attacker can guarantee success if $t + 1 < d - (t - \tau) \implies t < \frac{d + \tau - 1}{2}$

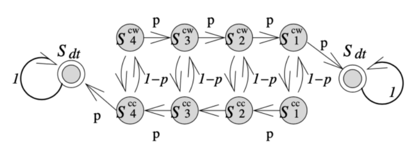
Optimal patrolling strategy

- 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:
  - the probability of visiting the worst case segment between robots decreases with increasing distance between the robots
  - 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





	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}
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

Probability of penetration

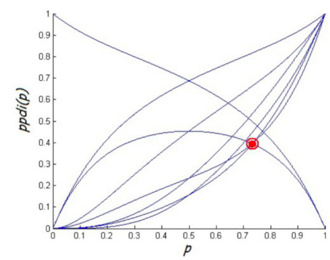
Algorithm 1 Algorithm FindFunc(d, t)

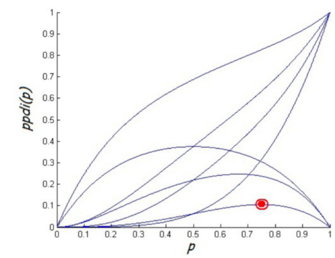
```
Create matrix M of size (2d+1)(2d+1), initialized with 0s
Fill out all entries in M as follows:
M[2d+1, 2d+1] = 1
for i ← 1 to 2d do
    M[i, max{i+1, 2d+1}] = p
    M[i, min{1, i−2}] = 1 − p
Compute MT = M^t
Res = vector of size d initialized with 0s
for 1 ≤ loc ≤ d do
    V = vector of size 2d+1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

- All computations are symbolic. The result are functions $\mathrm{ppd}_i : [0,1] \mapsto [0,1]$ expressing the probability of catching attacker at s_i for a given probability p of turn.

Optimal turn probability

- Maximin value for $p_{opt} = \operatorname{argmax}_{0 \le p \le 1}\{\min_{1 \le i \le d} \mathrm{ppd}_i(p)\}$
- Each line represents one segment (ppd_i)





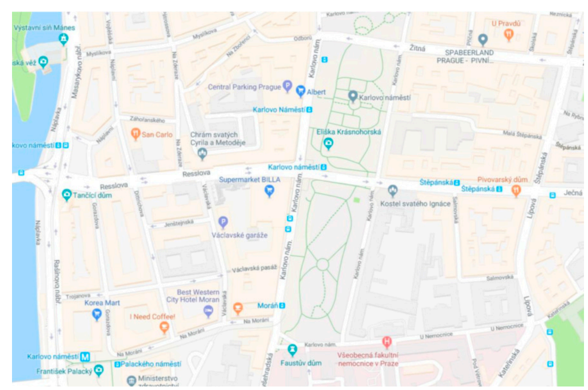
two possible maximin points (marked by a full circle).

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 $(1 - p)$

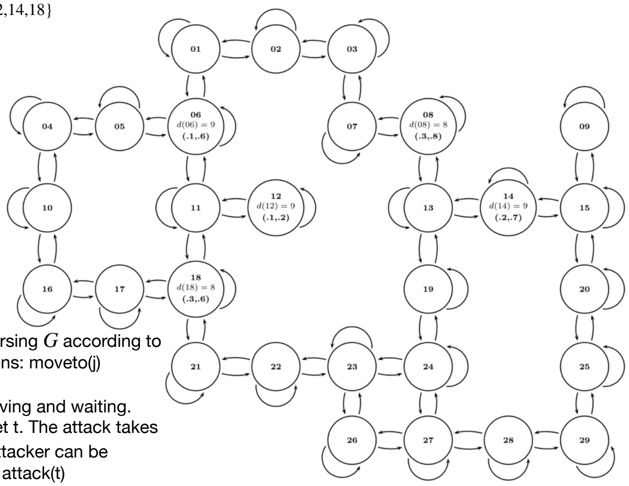
Area patrolling

- 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 $G = (V, A)$, V - vertices, A - arcs (edges)
- Targets $T \subseteq V$, $T = \{6,8,12,14,18\}$
- Penetration time $d(t)$
- Target values $(v_d(t), v_a(t))$



- Single defender: traversing G according to a Markov policy. Actions: moveto(j)
- Single attacker: observing and waiting. Then attacking a target t. The attack takes $d(t)$ time during the attacker can be caught. Actions: wait, attack(t)

Area patrolling - Formal model

- Defender utility function $u_{\mathbf{d}}(x) = \begin{cases} \sum_{i \in T} v_{\mathbf{d}}(i), & x = \textit{intruder-capture} \text{ or } \textit{no-attack} \\ \sum_{i \in T \setminus \{t\}} v_{\mathbf{d}}(i), & x = \textit{penetration-t} \end{cases}$
- Attacker utility function $u_{\mathbf{a}}(x) = \begin{cases} 0, & x = \textit{no-attack} \\ v_{\mathbf{a}}(t), & x = \textit{penetration-t} \\ -\epsilon, & x = \textit{intruder-capture} \end{cases}$
  - $\epsilon \in \mathbb{R}^+$ is the penalty

Solving zero-sum patrolling game

- We assume $\forall t \in T : v_a(t) = v_d(t)$, and attacker cannot play no-attack for infinite time.
- $a(i, j) = 1$ if the patrol can move from i to j in one step; else 0
- $P_C(t, h)$ is the probability of catching 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

$$\max u$$

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

$\alpha_{i,j}$ - strategy of the defender

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

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

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

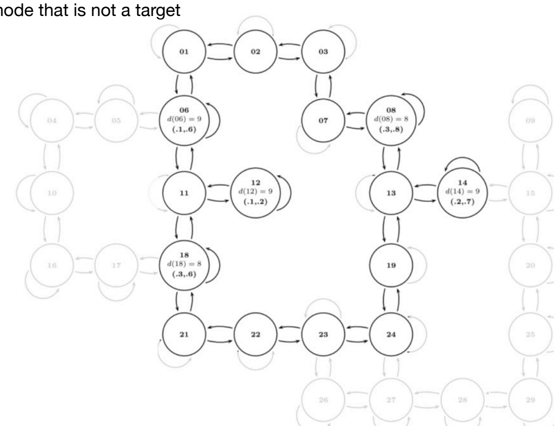
$$\gamma_{i,j}^{w,t} = \sum_{x \in V \setminus \{t\}} \left(\gamma_{i,x}^{w-1,t} \alpha_{x,j}\right) \quad \forall w \in \{2, \ldots, d(t)\},\ t \in T,\ i, j \in V \setminus \{t\}$$

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

$$u \leqslant u_{\mathbf{d}}(\textit{intruder-capture}) P_c(t,h) + u_{\mathbf{d}}(\textit{penetration-t})\left(1 - P_c(t,h)\right) \quad \forall t \in T,\ h \in V$$

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

- Game Theory can be applied to real world problems in robotics
- Pursuit-evasion games
  - Perfect information capture
  - Visibility-based tracking
  - Patrolling
    - Security resources allocation
    - perimeter patrolling
    - area patrolling
- Artificial Intelligence (Game Theory) 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.