Artificial Intelligence in Robotics

Lecture 11: GT in Robotics

Viliam Lisý

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

Game Theory

Mathematical framework studying strategies of players in situations where the outcomes of their actions critically depend on the actions performed by the other players.

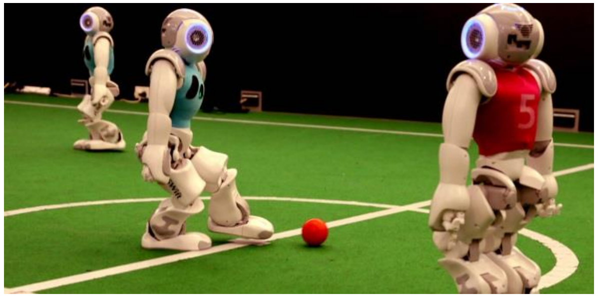


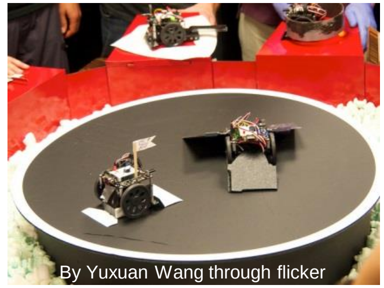




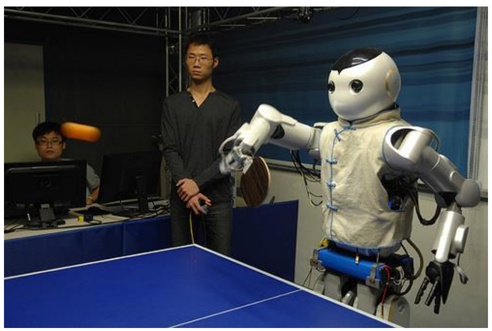


Robotic GT Applications





By Yuxuan Wang through flicker



Adversarial vs. Stochastic Environment

Deterministic environment

- The agent can be predict next state of the environment exactly

Stochastic environment

- Next state of the environment comes from a known distribution

Adversarial environment

- The next state of the environment comes from an unknown (possibly nonstationary) distribution

Game theory is optimizes behavior in adversarial environments

GT and Robust Optimization

It is sometimes useful to model unknown environmental variables as chosen by the adversary

- the position of the robot is the worst consistent with observations
- the planned action depletes the battery the most that it can
- the lost person in the woods moves to avoid detection

GT can be used for robust optimization without adversaries

Normal form game

N is the set of players

A_i is the set of actions (pure strategies) of player $i \in N$

$r_i: \prod_{j \in N} A_j \to \mathbb{R}$ is immediate payoff for player $i \in N$

Mixed strategy

$\sigma_i \in \Delta(A_i)$ is a probability distribution over actions

we naturally extend r_i mixed strategies as the expected value

Best response

of player i to strategy profile of other players σ_{-i} is

$$BR(\sigma_{-i}) = \arg \max_{\sigma_i \in \Delta(A_i)} r_i(\sigma_i, \sigma_{-i})$$

Nash equilibrium

Strategy profile σ^* is a NE, iff $\forall i \in N : \sigma_i^* \in BR(\sigma_{-i}^*)$

Normal form game

Player 2
Column player
Minimizer

Player 1
Row player
Maximizer

	r	p	s
R	0.5	0	1
P	1	0.5	0
S	0	1	0.5

0-sum game

Pure strategy, mixed strategy, Nash equilibrium, game value

Computing NE

LP for computing Nash equilibrium of 0-sum normal form game

$$\max_{\sigma_1, U} U$$

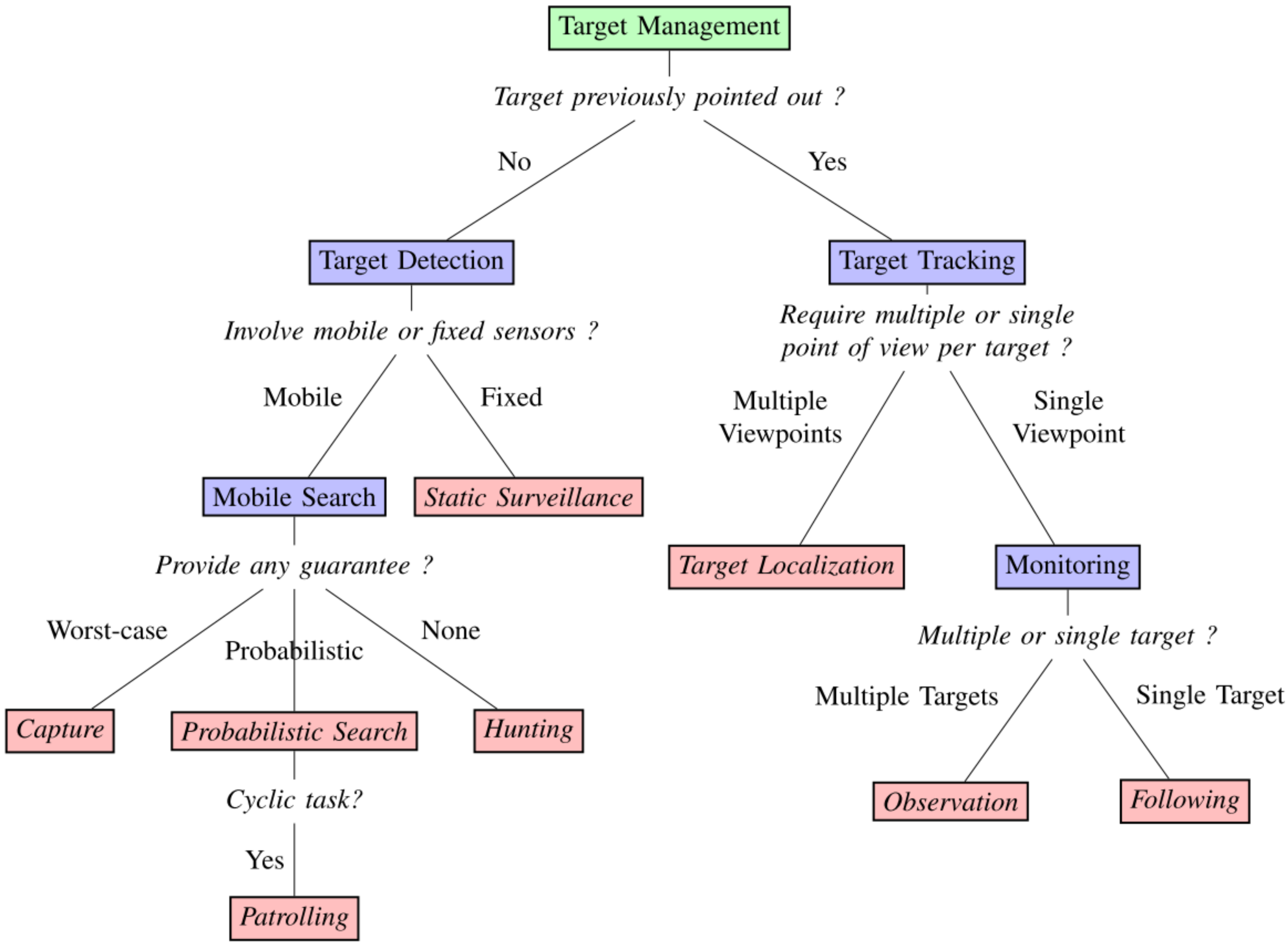
$$s.t. \sum_{a_1 \in A_1} \sigma_1(a_1) r(a_1, a_2) \geq U \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} \sigma_1(a_1) = 1$$

$$\sigma_1(a_1) \geq 0 \qquad \forall a_1 \in A_1$$

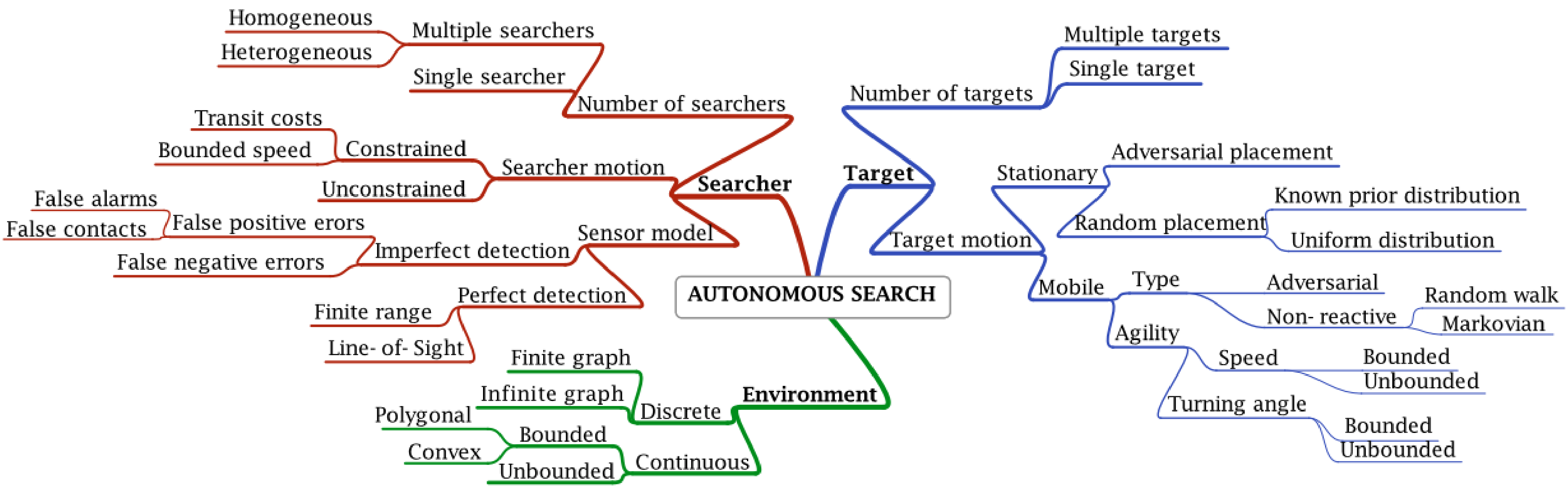
Pursuit-Evasion Games

Task Taxonomy



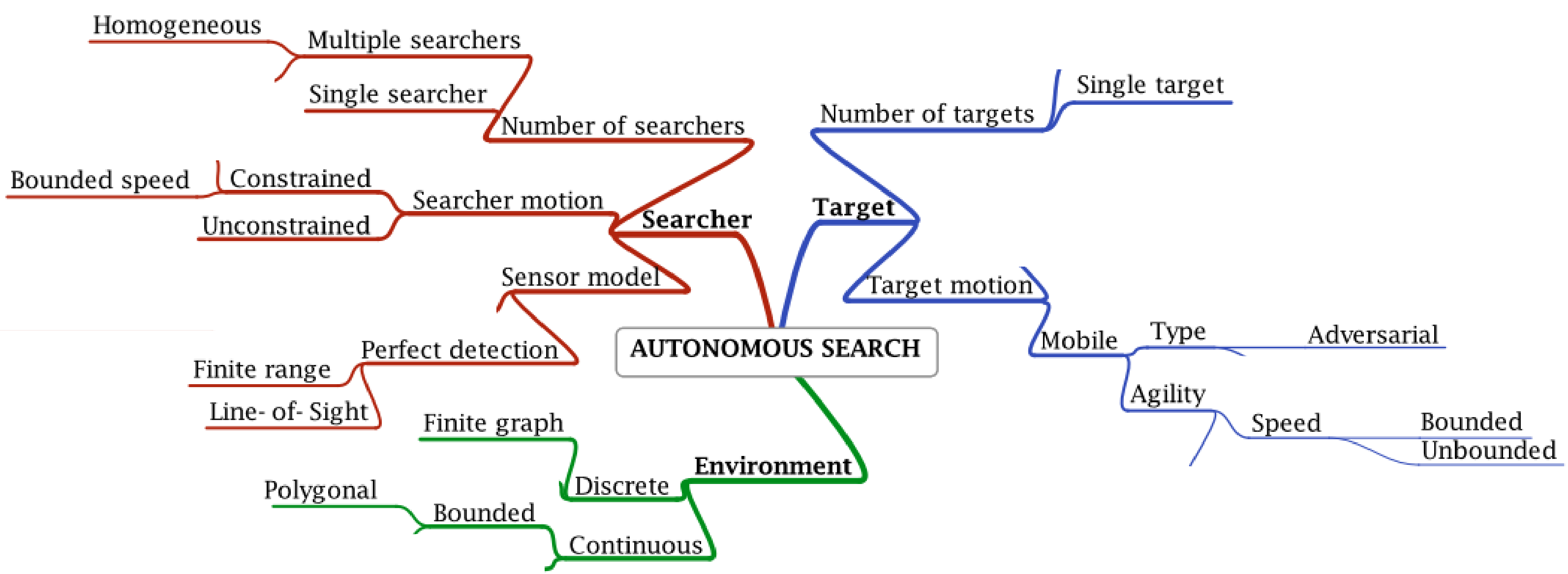
Robin, C., & Lacroix, S. (2016). Multi-robot target detection and tracking: taxonomy and survey. Autonomous Robots, 40(4), 729–760.

Problem Parameters



Chung, T. H., Hollinger, G. A., & Isler, V. (2011). Search and pursuit-evasion in mobile robotics`: A survey. Autonomous Robots, 31(4), 299–316.

Problem Parameters



Chung, T. H., Hollinger, G. A., & Isler, V. (2011). Search and pursuit-evasion in mobile robotics`: A survey. Autonomous Robots, 31(4), 299–316.

PERFECT INFORMATION CAPTURE

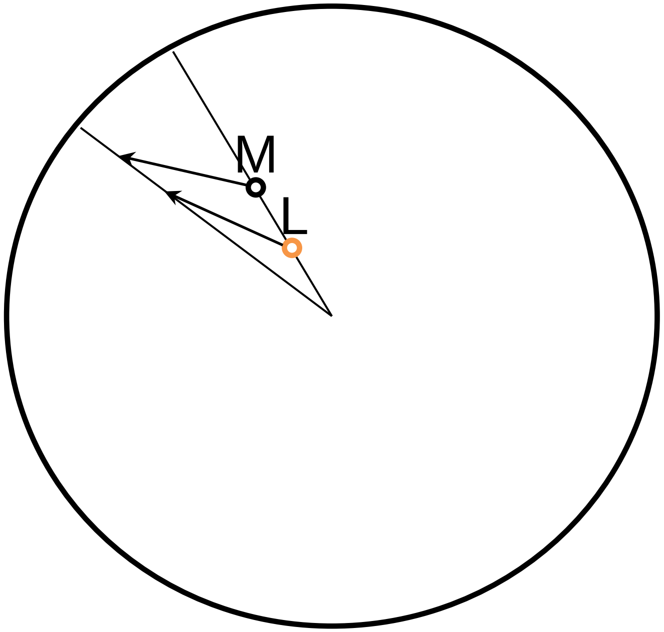
Lion and man game

arena with radius r

man and lion have unit speed

alternating moves

can lion always capture the man?



Algorithm for the lion

start from the center

stay on the radius that passes the man

move as close to the man as possible

Analysis

capture time with discrete steps $O(r^2)$ [Sgall 2001]

no capture in continuous time

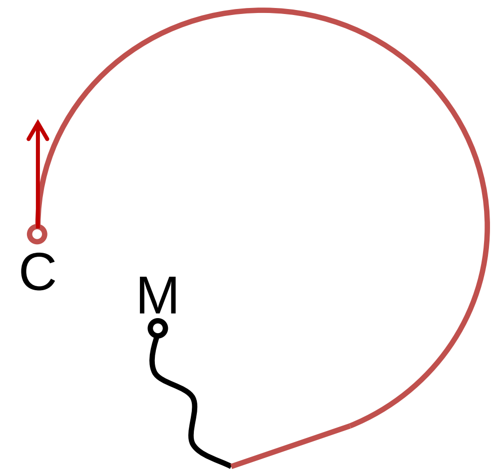
the lion can get to distance c in time $O(r \log \frac{r}{c})$ [Alonso at al 1992]

single lion can capture the man in any polygon [Isler et al. 2005]

Modelling movement constraints

Homicidal chauffeur game [Isaacs 1951]

- unconstraint space
- pedestrian is slow, but highly maneuverable
- car is faster, but less maneuverable (Dubin's car)
- can the car run over the pedestrian?



$$\dot{x}_M = u_M,\ |u_M| \leq 1;\ \dot{x}_C = (v\cos\theta, v\sin\theta);\ \dot{\theta} = u_C,\ u_C \in \{-1,0,1\}$$

Differential games

- $\dot{x} = f(x, u_1(t), u_2(t)),\ L_i(u_1, u_2) = \int_{t=0}^{T} g_i(x(t), u_1(t), u_2(t))\,dt$
- analytic solution of partial differential equation (gets intractable quickly)

Incremental Sampling-based Method

S. Karaman, E. Frazzoli: Incremental Sampling-Based Algorithms for a Class of Pursuit-Evasion Games, 2011.

1 evader, several pursuers

Open-loop evader strategy (for simplicity)

Stackelberg equilibrium

- the evader picks and announces her trajectory
- the pursuers select trajectory afterwards

Heavily based on RRT* algorithm

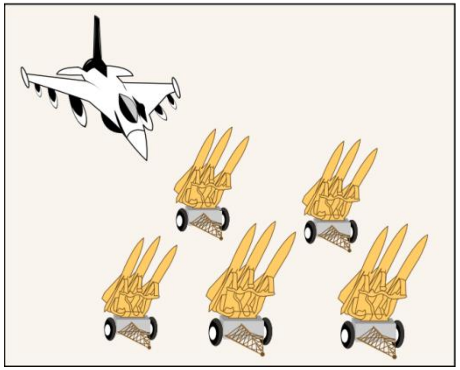


Image by MIT OpenCourseWare

Incremental Sampling-based Method

Algorithm

Initialize evader's and pursuers' trees T_e and T_p

For $i = 1$ to N do

$n_{e,new} \leftarrow Grow(T_e)$

if $\{n_p \in T_p : dist(n_{e,new}, n_p) \leq f(i)\ \&\ \mathrm{time}(\mathrm{n_p}) \leq \mathrm{time}(\mathrm{n_{e,new}})\} \neq \emptyset$ then

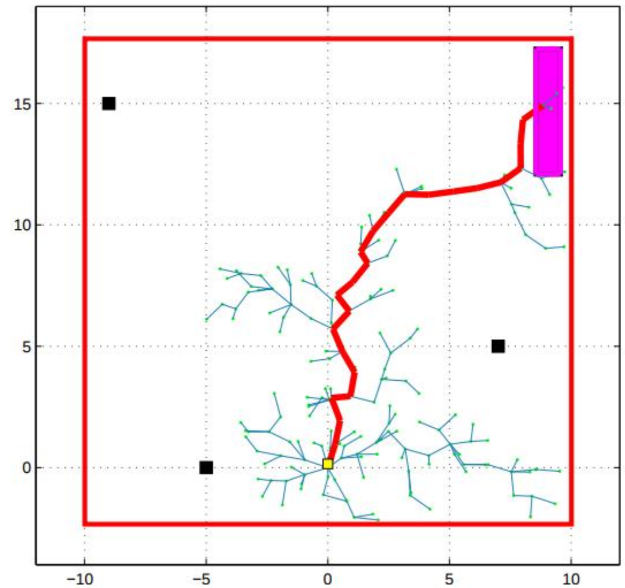
delete $n_{e,new}$

$n_{p,new} \leftarrow Grow(T_p)$

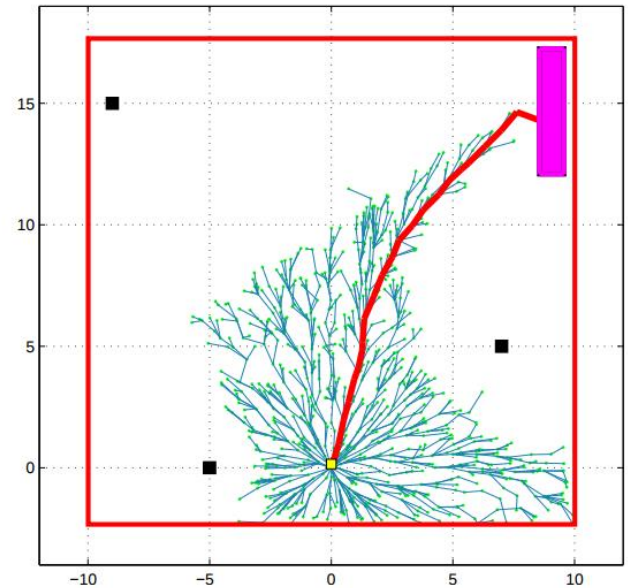
$C = \{n_e \in T_e : dist(n_e, n_{p,new}) \leq f(i)\ \&\ \mathrm{time}(\mathrm{n_{p,new}}) \leq \mathrm{time}(\mathrm{n_e})\}$

delete $C \cup \mathrm{descendants(C, T_e)}$

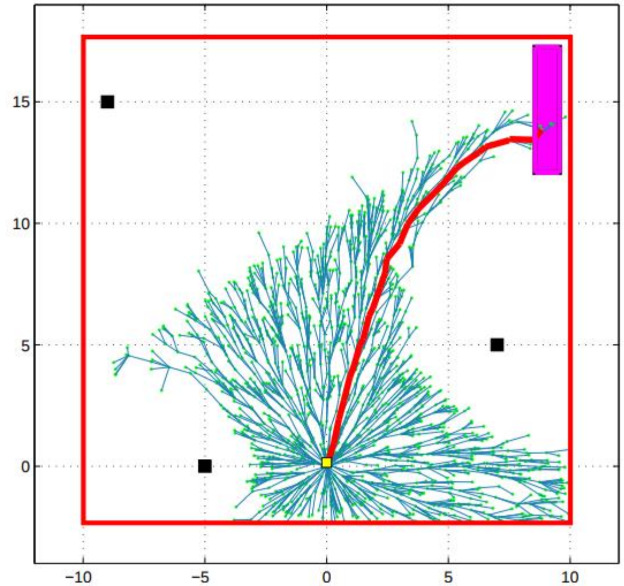
For computational efficiency pick $f(i) \approx \frac{\log |T_e|}{|T_e|}$



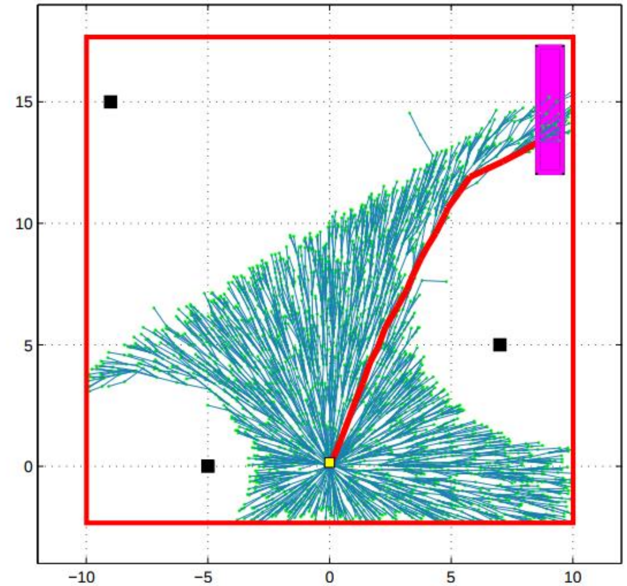
iteration 500



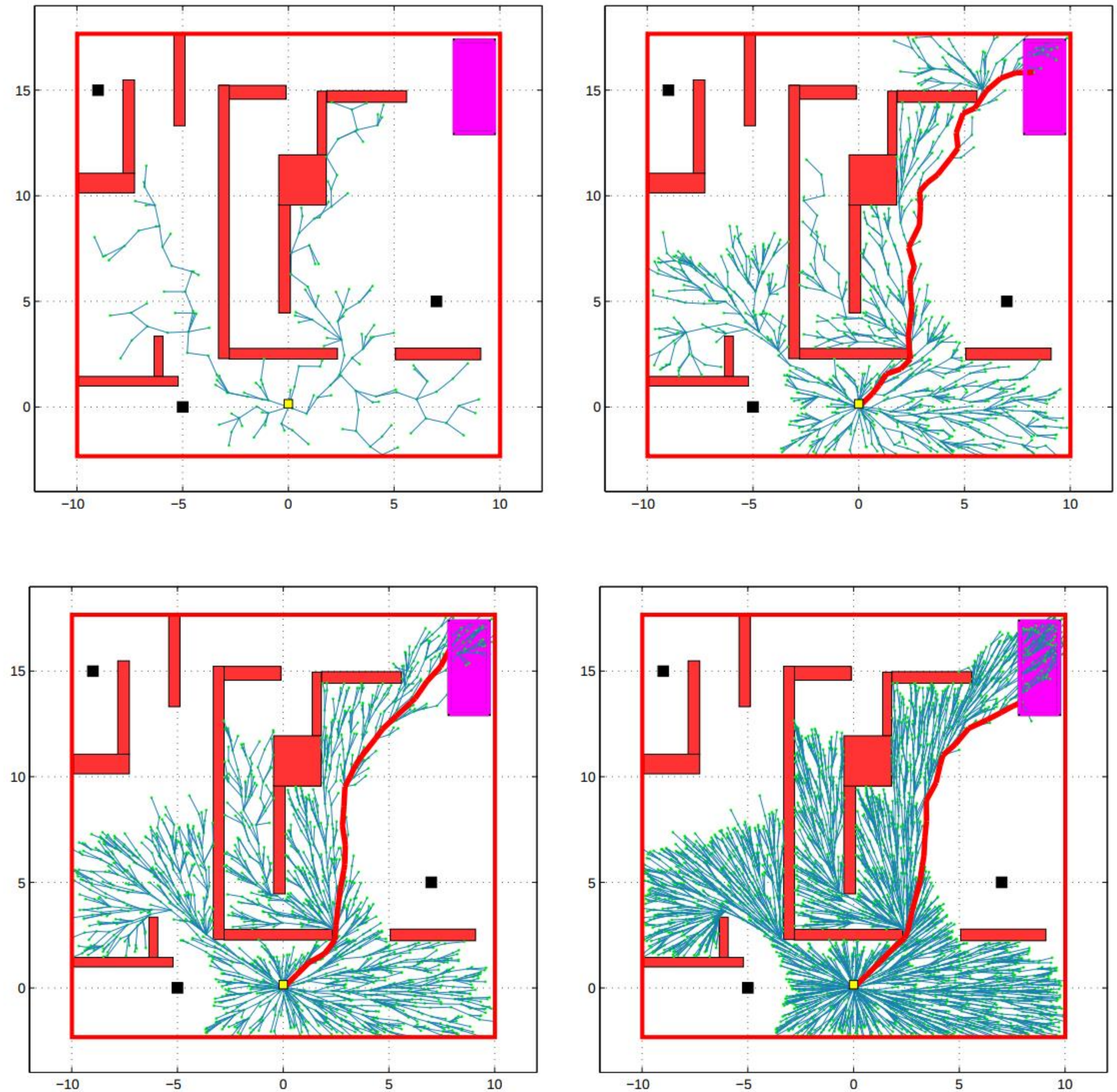
iteration 3000



iteration 5000



iteration 10000


15
10
5
0
−10
−5
0
5
10

Discretization-based approaches

Open-loop strategies are very restrictive

Closed-loop strategies are generally intractable

Cops and robbers game

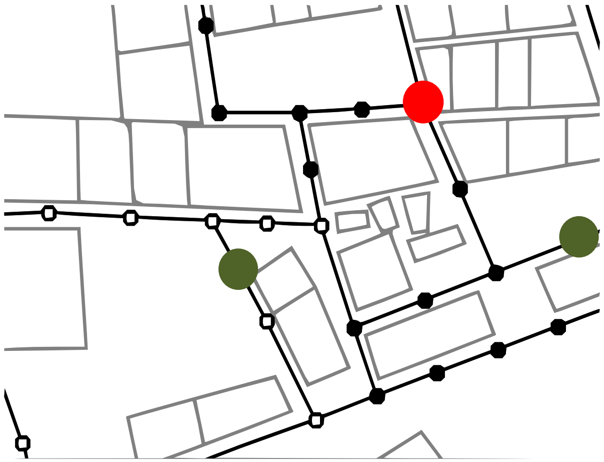
Graph $G = (V, E)$

Cops and robbers in vertices

Alternating moves along edges

Perfect information

Goal: step on robber's location



Cop number: Minimum number of cops necessary to guarantee capture or the robber regardless of their initial location.

Cops and robbers game

Neighborhood $N(v) = \{u \in V : (v, u) \in E\}$

Marking algorithm (for single cop and robber):

1. For all $v \in V$, mark state (v, v)
2. For all unmarked states (c, r)
   If $\forall r' \in N(r) \exists c' \in N(c)$ such that (c', r') is marked, then mark (c, r)
3. If there are new marks, go to 2.

If there is an unmarked state, robber wins

If there is none, the cop's strategy results from the marking order

Cops and robbers game

Time complexity of marking algorithm for k cops is $O\left(n^{2(k+1)}\right)$.

Determining whether k cops with a given locations can capture a robber on a given undirected graph is EXPTIME-complete [Goldstein and Reingold 1995].

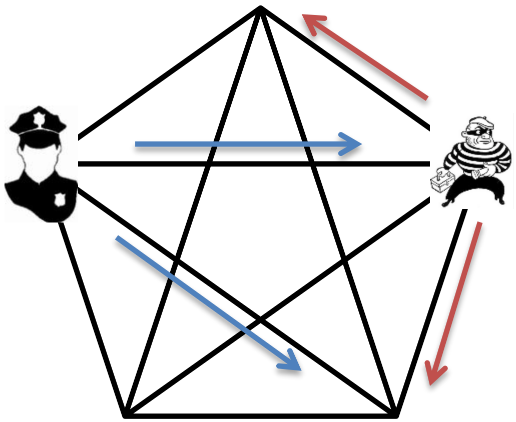
The cop number of trees and cliques is one.

The cop number on planar graphs is at most three [Aigner and Fromme 1984].

Cops and robbers game

Simultaneous moves

- No deterministic strategy



- Optimal strategy is randomized

Stochastic (Markov) Games

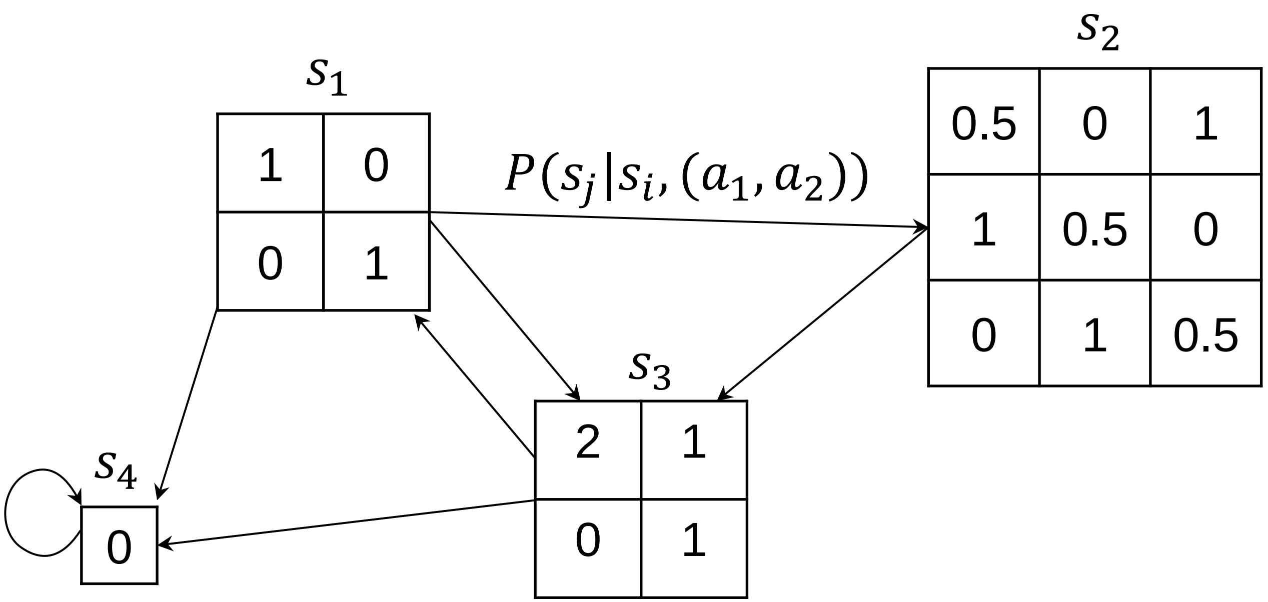
N is the set of players

S is the set of states (games)

$A = A_1 \times \cdots \times A_n$, where A_i is the set of actions of player i

$P: S \times A \times S \to [0,1]$ is the transition probability function

$R = r_1, \ldots, r_n$, where $r_i: S \times A \to \mathbb{R}$ is immediate payoff for player i



Stochastic (Markov) Games

Markovian policy: $\sigma_i : S \rightarrow \Delta(A)$

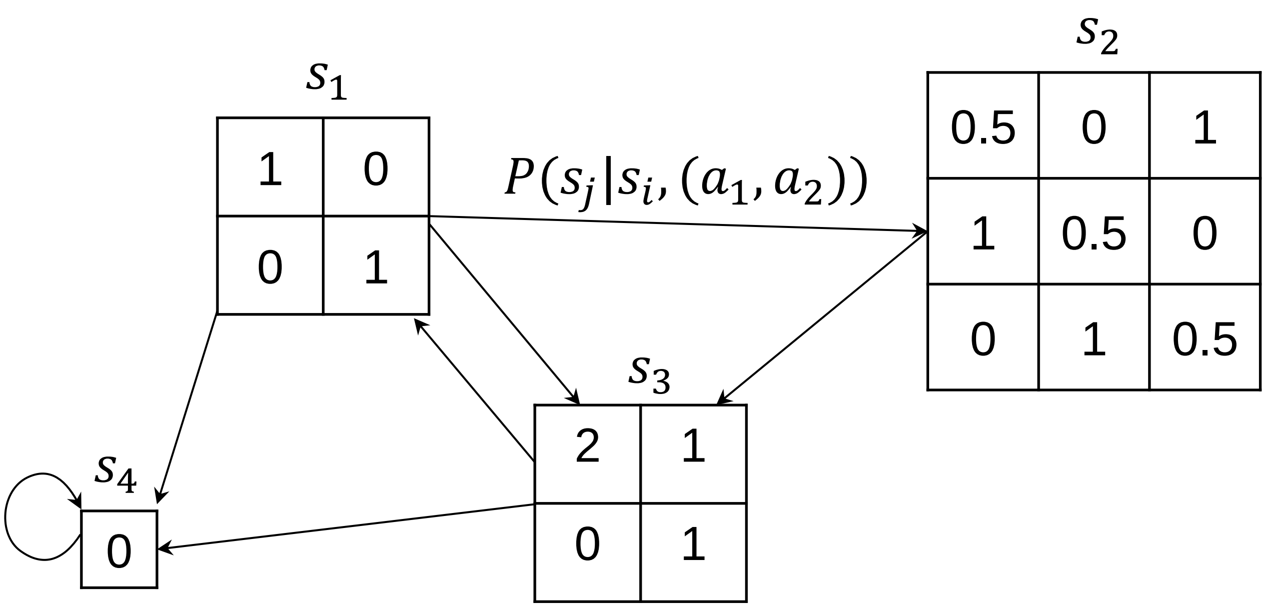
Objectives

Discounted payoff: $\sum_{t=0}^{\infty} \gamma^t r_i(s_t, a_t)\,, \gamma \in [0,1)$

Mean payoff: $\lim_{T\rightarrow\infty} \frac{1}{T} \sum_{t=0}^{T} r_i(s_t, a_t)$

Reachability: $P\big(reach(G)\big),\;\; G \subseteq S$

Finite vs. infinite horizon



Value Iteration in SG

Adaptation of algorithm from Markov decision processes (MDP)

For zero-sum, discounted, infinite horizon stochastic games

- $\forall s \in S$ initialize $v(s)$ arbitrarily (e.g., $v(s) = 0$)
- until v converges
  - for all $s \in S$
    - for all $(a_1, a_2) \in A(s)$
      - $Q(a_1, a_2) = \mathrm{r}(\mathrm{s}, \mathrm{a}_1, \mathrm{a}_2) + \gamma \sum_{s' \in S} P(s' | s, a_1, a_2) v(s')$
    - $v(s) = \max_x \min_y xQy$ // solves the matrix game Q

Converges to optimum if each state is updated infinitely often

- the state to update can be selected (pseudo)randomly

Pursuit Evasion as SG

$N = (e, p)$ is the set of players

$S = \left(v_e,\ v_{p_1}, \dots, v_{p_n}\right) \in V^{n+1} \cup T$ is the set of states

$A = A_e \times A_p$, where $A_e = E, A_p = E^n$ is the set of actions

$P: S \times A \times S \rightarrow [0,1]$ is deterministic movement along the edges

$R = r_e, r_p$, where $r_e = -r_p$ is one if the evader is captured

Summary

PEGs studied in various assumptions

Simplest cases can be solved analytically

More complex cases have problem-specific algorithms

Even more complex cases best handled by generic AI methods