Artificial Intelligence in Robotics

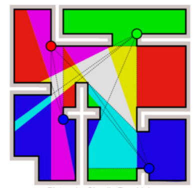
Lecture 10: Visibility based pursuit evasion

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

Visibility based pursuit evasion

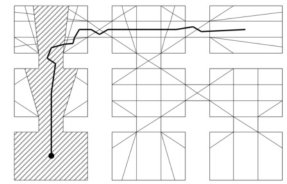
- The goal of pursuers is to just spot the evader.
- The goal of evader is to avoid detection.

Guarding a gallery



Picture by Claudio Rocchini

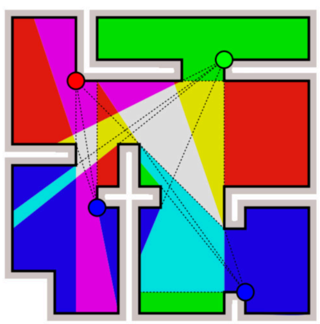
Cleaning an environment



Visibility-based tracking

Static pursuers, no evader

- Art gallery problem (Klee 1973)
  - Simple polygon P
  - vertices : $v_1, v_2, \ldots, v_n$
  - $x \in P$ covers $y \in P$ iff $xy \subseteq P$
  - What is the minimum number of guards that can cover the entire polygon P?

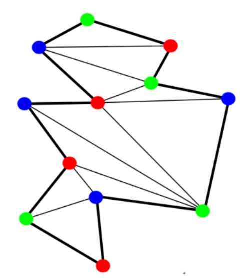


Picture by Claudio Rocchini

Art gallery problem

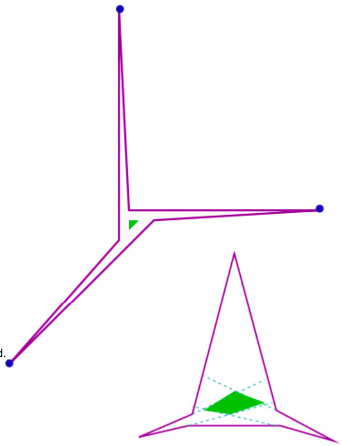
- Theorem (Chvatal 1975)
  - $\lfloor \frac{n}{3} \rfloor$ guards are always sufficient and sometimes necessary.
- Necessary - a comb

- Sufficient (Fisk 1978)
  - Simple polygons always have a triangulation.
  - The triangulation can be colored by three colors.
  - The least used color is used no more than $\lfloor \frac{n}{3} \rfloor$ times.
  - Vertices of each color cover the entire polygon.
- For orthogonal polygons only $\lfloor \frac{n}{4} \rfloor$ guards needed.
- Computing minimal number of guards is NP-hard.



Art gallery problem

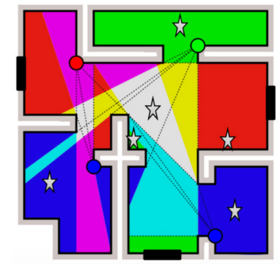
- Pathological cases (from Subhash Suri's slides):
  - less guards may be enough
  - seeing the boundary is not enough
  - optimal positions not on boundary



- Fun facts:
  - For orthogonal polygons, only $\lfloor n/4 \rfloor$ guards are needed.
  - Computing minimal number of guards for a polygon is NP-hard.
  - The problem is closely connected to the set cover problem.

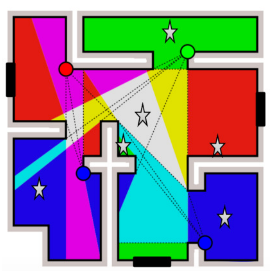
Static pursuer and mobile evader

- More realistic art gallery problem
  - There are m cameras.
  - A guard can watch k cameras.
  - What cameras show?
  - Thief has to enter, steal, exit.
  - Penalty for each seen second/meter.

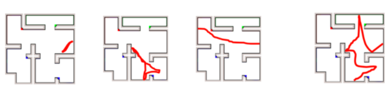


Matrix game representation

- Defender's actions: watch k of m cameras
- Attacker's actions: choose a path door-target-door



∞

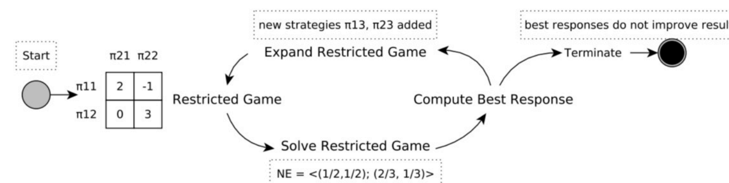


$\binom{4}{1}$

$p * v_1$	$v_2 + v_3$	$p * v_4$	$p * (v_1 + v_2 + v_3 + v_3)$
v_1	$p * (v_2 + v_3)$	$p * v_4$	$p * (v_1 + v_2 + v_3 + v_3)$
v_1	$v_2 + v_3$	$p * v_4$	$v_1 + v_2 + v_3 + v_3$
v_1	$p * (v_2 + v_3)$	$p * v_4$	$p * (v_1 + v_2 + v_3 + v_3)$

p - probability of not being detected if seen

Double oracle framework



McMahan, Gordon, Blum: Planning in the presence of cost functions controlled by an adversary. ICML 2003.

Double oracle in a matrix game

	1			
1	2	4	3	3
→	3	3	7	2
	2	4	5	5
	2	1	4	1

Double oracle in a matrix game

	1			↓
0	2	4	3	3
→ 1	3	3	7	2
	2	4	5	5
	2	1	4	1

Double oracle in a matrix game

	0.5			0.5 ↓
0.5	2	4	3	3
0.5	3	3	7	2
→	2	4	5	5
	2	1	4	1

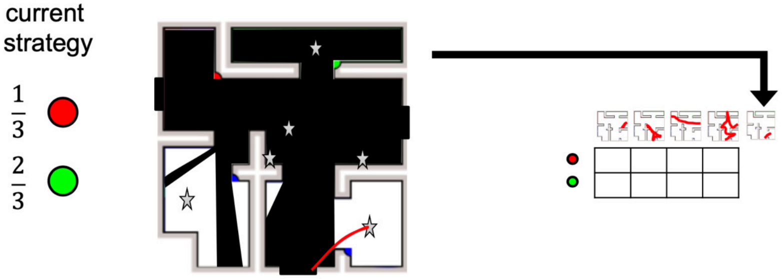
Double oracle in a matrix game

	0.75			0.25 ↓
0	2	4	3	3
0.75	3	3	7	2
→ 0.25	2	4	5	5
	2	1	4	1

Always converges and finds NE.

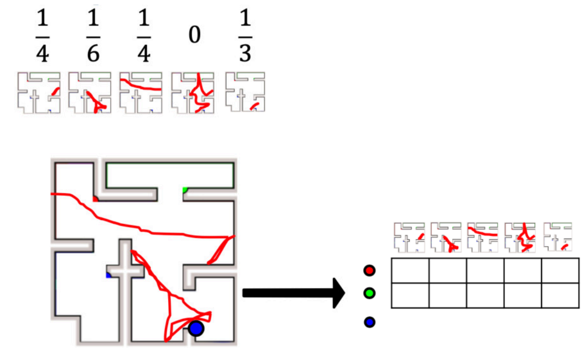
Attacker's best response oracle

Defender's
current
strategy



Path planning with costs defined by cameras in use (A*, TSP, etc.)

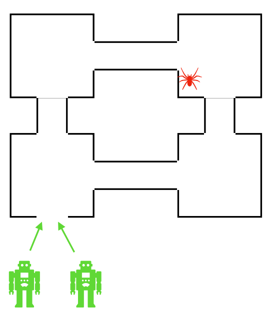
Defender's best response oracle



Greedy / combinatorial search for best k camera positions

Mobile pursuers and infinitely fast evader

- Goal: Clear a polygonal environment (find/decide if anybody is inside)
- Hunters and prey problem
  - Simple polygon F with vertices : $v_1, v_2, \ldots, v_n$
  - k hunters with bounded speed
  - A prey with unbounded speed
  - Can hunters spot the prey?
- Definitions:
  - $h^i : [0,\infty) \mapsto F$ is the Pursuer i's strategy
  - $e : [0,\infty) \mapsto F$ is the evader strategy
  - $V(q) \subseteq F$ is the area visible from $q \in F$
- Solution
  - Strategy $h = h^1, h^2, \ldots, h^k$ is a solution if for every continuous $e : [0,\infty) \mapsto F$ there exists $t \in [0,\infty), i \in \{1,\ldots,k\}$ such that $e(t) \in V(h^i(t))$.
- Definitions: Any region that can contain an evader is referred as contaminated
  - Otherwise it is referred as cleared. If a region is contaminated, becomes cleared, and then becomes contaminated again, it referred as recontaminated.



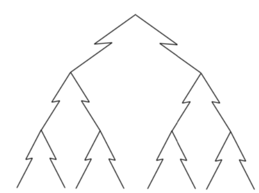
Clearing polygonal environment

Theorem (Urrutia, 1997): $O(\log n)$ hunters are always sufficient and occasionally necessary to spot a prey in polygon with n vertices.

Sufficient

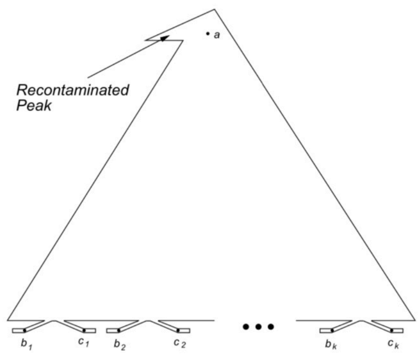
let $f(n)$ be the required number of hunters
each polygon has a diagonal splitting it to two with $\leq \frac{2n}{3}$ vertices
if one guard guards the diagonal, $f(n) \leq f\left(\frac{2n}{3}\right) + 1$
from master theorem, $f(n) \in O(\log n)$

Necessary



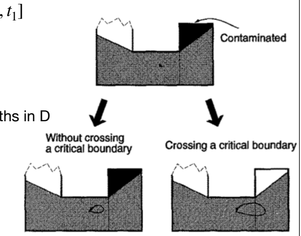
Clearing polygonal environment

Theorem (Guibas et al. 1997): There exists a sequence of simply-connected free spaces clearable by single pursuer, such that O(n) recontaminations are required for n edges.



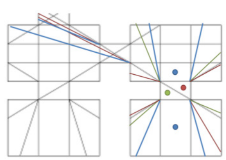
Clearing polygonal environment

- Algorithm for Single pursuer and single evader
  - Guibas, L. J., Latombe, J.-C., Lavalle, et al.: Visibility-Based Pursuit-Evasion in a Polygonal Environment. WADS, 1997
  - critical event analysis (similar to event-based simulation)
- Definitions
  - The state space X is spanned by all (x^p, x^e) where x^p is a position of pursuer and x^e is a position of evader
  - The information state $\eta = (x^p, S)$ where x^p is pursuer's position and $S \subseteq F$ is contaminated region.
  - $\Psi(\eta, h, t_0, t_1)$ is the information state after executing h from η during $[t_0, t_1]$
  - Region $D \subseteq F$ is conservative, if for all continuous $h_1, h_2 : [t_0, t_1] \mapsto D$ $h_1(t_0) = h_2(t_0) \wedge h_1(t_1) = h_2(t_1) \implies \Psi(\eta, h_1, t_0, t_1) = \Psi(\eta, h_2, t_0, t_1)$
    - The information state cannot be altered by moving along closed paths in D



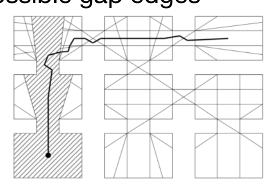
Gap edges

- Let $q \in F$. We define two types of edges in visibility polygon
  - Obstacle edge
  - Gap edge (bordering two free spaces)
    - Label 0 if it borders two cleared spaces
    - Label 1 if one space is contaminated
  - Let $B(q)$ be a binary sequence of gap edges labeling.
  - q and $B(q)$ characterize the current information state.
- The information state can change only when a gap edge appears or disappears



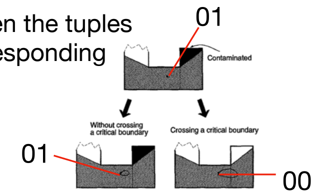
Clearing polygonal environment

- Partition space F into convex cells by identifying critical places at which edge visibility changes. (See the article and references there for the details)
- This partition defines the cell graph G_C. Vertices are the convex cells. Edges between adjacent cells.
- A directed information state graph, G_I, is derived from G_C. Vertices of G_I are tuples $(v_C, B(q))$ where $q \in v_C$. v_C is a cell from G_C. We add tuples for all possible gap edges labelling.

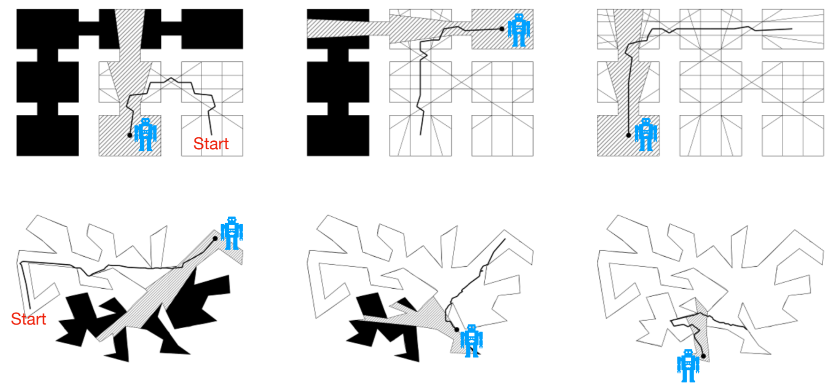


Clearing polygonal environment

- Edges of G_I are defined:
  - Inspect all possible transition between the tuples of adjacent cells and create the corresponding edges.
- Search the graph G_I
  - Start in a vertex. Gap edges labelling $B(q)$ is all 1s.
  - Find a path in G_I to a cell with $B(q)$ all “0”. (Using Dijkstra, bfs,…)



Cleaning polygonal environment



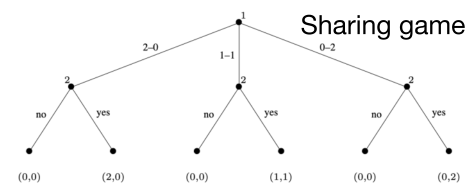
Mobile pursuer mobile evader

- Visibility based tracking
  - Graph of locations $G = (V, E)$
  - visibility relation $Sees(v_1, v_2)$
  - k pursuers, 1 evader
    - both move on the graph
    - both unit speed
  - Goal
    - The pursuers want to see the evader as often as possible
    - Minimize the set of possible positions

Extensive form games

- Definition: A (finite) perfect-information game (in extensive form) is a tuple $G = (N, A, H, Z, \chi, \rho, \sigma, u)$, where
- N is a set of players
- A is a set of actions
- H is a set of nonterminal nodes
- Z is a set of terminal nodes, disjoint from H

Sharing game

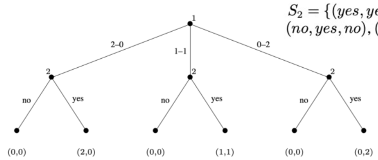


- $\chi : H \mapsto 2^A$ is the action function, assigns available action in a given node
- $\rho : H \mapsto N$ is the player function, it assigns a player to each nonterminal node
- $\sigma : H \times A \mapsto H \cup Z$ is the successor function, maps a choice node and an action to a new node. Such that $\forall h_1, h_2 \in H \, \forall a_1, a_2 \in A : \sigma(h_1, a_1) = \sigma(h_2, a_2) \implies h_1 = h_2 \wedge a_1 = a_2$ (tree condition)
- $u = (u_1, \ldots, u_n)$, where $u_i : Z \mapsto \mathbb{R}$ is a utility function of player i on the terminal nodes.

Extensive form games

$S_1 = \{2\text{–}0, 1\text{–}1, 0\text{–}2\}$

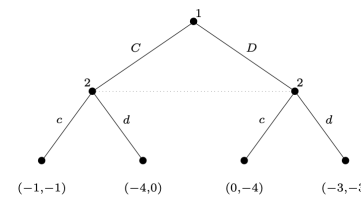
$S_2 = \{(yes, yes, yes), (yes, yes, no), (yes, no, yes), (yes, no, no), (no, yes, yes), (no, yes, no), (no, no, yes), (no, no, no)\}$



- The set of pure strategies (not mixed) of player i consists of Cartesian product $\prod_{h \in H, \rho(h)=i} \chi(h)$.
- Mixed strategies are probabilistic distributions over the set of pure strategies.
- **Theorem**. Every (finite) perfect-information game in extensive form has a pure-strategy Nash equilibrium.
- Could be find by minimax algorithm for two players zero sum games.

Extensive form games

- Definition: An imperfect-information game (in extensive form) is a tuple $G = (N, A, H, Z, \chi, \rho, \sigma, u, I)$ where $G = (N, A, H, Z, \chi, \rho, \sigma, u)$ is a perfect information EFG
- $I = (I_1, \ldots, I_n)$, where $I_i = (I_{i,1}, \ldots, I_{i,k_i})$ is a partition of $\{h \in H : \rho(h) = i\}$ with the property $\forall j \, \forall h, h' \in I_{i,j} : \chi(h) = \chi(h')$

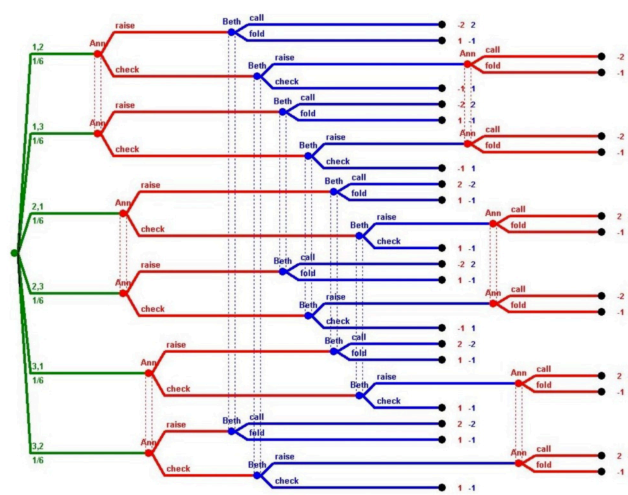


The Prisoner's Dilemma game in extensive form

Extensive form games

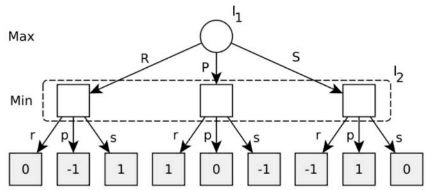
- The set of pure strategies (not mixed) of player i consists of Cartesian product $\prod_{I_{i,j} \in I_i} \chi(I_{i,j})$.
- Equilibrium could be find by solving the corresponding linear program.

Extensive form game

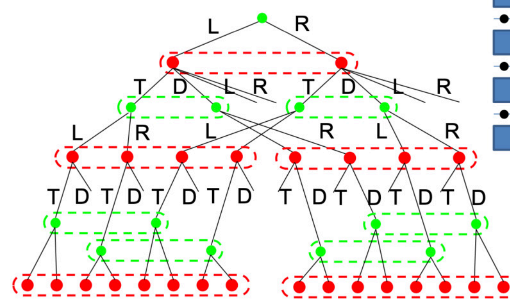


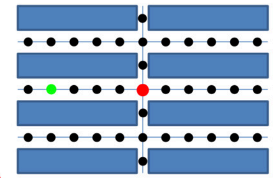
Simultaneous moves in EFG

	r	p	s
R	0	−1	1
P	1	0	−1
S	−1	1	0



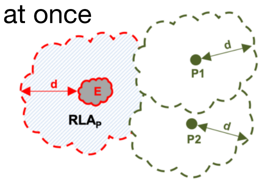
Pursuit evasion as EFG





Pursuit evasion as EFG

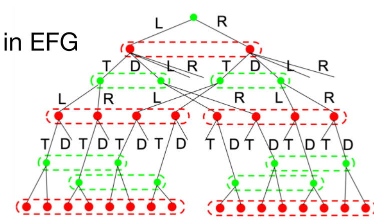
- Relaxed look-ahead heuristic (Raboin at al. 2011)
  - evader can be on worst possible position pursuers can be everywhere at once



|positions reachable by evader - positions that can be possibly seen|

Pursuit evasion as EFG

- Information set tree
  - Nodes in IST are Information Sets in EFG
  - + IST is much smaller
  - + solved as perfect information
  - - overly pessimistic (worst possible observation)



- Use depth limited minimax search on IST with relaxed look ahead heuristics
- Or Monte Carlo Tree Search

Summary

- Static camera position
- Camera switching
- Spotting fast evader
- Tracking realistic evader

Resources

Urrutia, J. (1997). Art Gallery and Illumination Problems. Handbook of Computational Geometry, 973–1027.

Guibas, L. J., Latombe, J. C., LaValle, S. M., Lin, D., & Motwani, R. (1997, August). Visibility-based pursuit-evasion in a polygonal environment. In Workshop on Algorithms and Data Structures (pp. 17-30).

McMahan, Gordon, Blum (2003): Planning in the presence of cost functions controlled by an adversary. ICML.

Raboin, E., Nau, D., Kuter, U., Gupta, S. K., & Svec, P. (2010). Strategy generation in multi-agent imperfect-information pursuit games. AAMAS, pp. 947-954.