Pursuit-Evasion Games I

Tomáš Kroupa

AI Center
Department of Computer Science
Faculty of Electrical Engineering
Czech Technical University in Prague

2023

Pursuit-Evasion in Mobile Robotics

One or more pursuers try to capture one or more evaders who try to avoid capture.

- The study of motion planning problems in adversarial settings
  - Detecting intruders
  - Playing hide-and-seek
  - Catching burglars
- The planner seeks an optimal strategy against the worst-case adversary

Classes of Pursuit-Evasion Games

Differential

- Hamilton-Jacobi-Isaacs differential equations model the dynamics
- Their solutions are players' strategies as control inputs for achieving the objectives
- 👍 Velocity or acceleration are expressed explicitly as differential constraints
- 👎 The resulting equations are very complicated and difficult to solve

Combinatorial

- A real environment is modeled as a polygon or graph
  - *The Cops and Robbers Game*
  - *Parson's game*
  - *The lion-and-man game*
- 👍 Complexity results and guarantees in terms of the size of game
- 👎 Abstraction from the continuous features of environment

Lecture Goals and Outline

To understand how

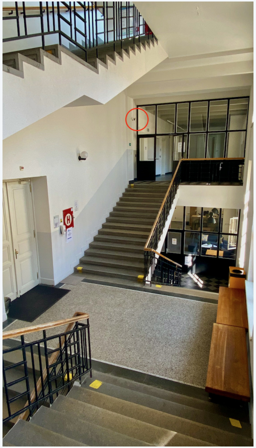
- *the robotic motion planning changes in the presence of an adversary pursuing their own goals and*
- *the robot's navigation can be enhanced using the game-theoretic methods in this case.*

1. Motivation: Am adversarial path planning problem
2. Two-player zero-sum games
3. Double Oracle algorithm for solving large games

Adversarial Path Planning Problem

What Path Should the Robot Follow to Avoid CCTV?

👉 The position of cameras is known



Adversarial Path Planning Problem

What Path Should the Robot Follow to Avoid CCTV?

👉 The position of cameras is known

The planner navigates a robot to a goal location in a previously mapped environment.

Planner

- Models the problem as a single-agent *Markov decision process*
- Must find a path minimizing the robot's visibility to cameras

Adversary

- Not present in the model

What Path Should the Robot Follow to Avoid CCTV?

👉 The adversary deploys cameras

Adversarial Path Planning Problem

What Path Should the Robot Follow to Avoid CCTV?

👉 The adversary deploys cameras

Both the planner and adversary can control the environment actively.

Planner

- Path π for the robot
- Finite set of paths Π
- Probability distribution $p \in \Delta_\Pi$

Adversary

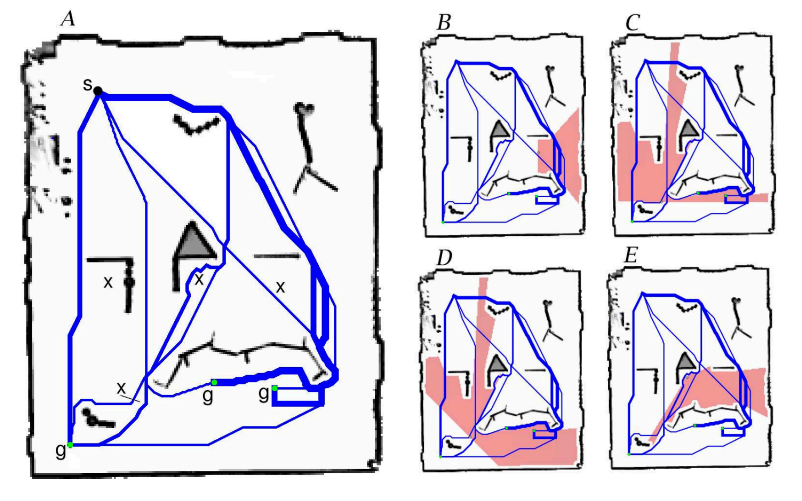
- Cost vector $\mathbf{c}$ (position of cameras)
- Finite set of cost vectors C
- Probability distribution $q \in \Delta_C$

Let $V(\pi, \mathbf{c})$ be the value of policy π and cost vector $\mathbf{c}$. Solve:

$$\min_{p \in \Delta_\Pi} \max_{q \in \Delta_C} \sum_{\pi \in \Pi} \sum_{\mathbf{c} \in C} p(\pi) q(\mathbf{c}) V(\pi, \mathbf{c})$$

Example of Solution

Blum et al. (2003)



Adversarial Path Planning Problem

Planning Paths: Experiments

Blum et al. (2003)

- The gridworld of size up to 269×226
- The robot can move in any of 16 compas directions
- Each cell has cost 1 and a cost proportional to the distance of camera

Computational limits

👉 Sets Π and C should be reasonably small

👉 Already $\binom{100}{2} = 4\,950$ positions exist for 2 cameras in the gridworld 10×10

Two-Player Zero-Sum Games

Two-Player Zero-Sum Game

👉 aka Matrix game

1. Players are the planner and the adversary
2. Strategy sets are M (planner) and N (adversary)
3. The loss matrix $\mathbf{C} = [c_{ij}]_{i \in M,\, j \in N}$ of the planner

The loss c_{ij} for planner playing $i \in M$ $=$ the reward for adversary playing $j \in N$

For example:

$$|M| = 2, \qquad |N| = 4, \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 4 & -1 \\ -1 & 1 & -2 & 5 \end{bmatrix}$$

Minmax/Maxmin Strategies

👉 We seek the optimal performance against the worst-case adversary

- Assume that the agents adopt minmax/maxmin strategies $\bar{i} \in M$ and $\bar{j} \in N$:

$$\bar{i} = 1, \qquad \bar{j} = 2, \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 4 & -1 \\ -1 & 1 & -2 & 5 \end{bmatrix}$$

The floor on the reward of adversary $= 0 \ \leq \ 4 =$ The ceiling of the loss of planner

$$\max_{j \in N} \min_{i \in M} c_{ij} \ \leq \ c_{\bar{i}\,\bar{j}} \ \leq \ \min_{i \in M} \max_{j \in N} c_{ij}$$

- Now, the adversary can increase the profit by playing $j = 3$
- In this case the planner would adopt $i = 2$
- Then the adversary would play $j = 4$ etc.

Mixed Strategies

👉 Randomize to play optimally!

A mixed strategy of a player is a probability distribution over the strategy set.

- Let Δ_M and Δ_N be the sets of mixed strategies of planner/adversary
- If the agents play $\mathbf{x} \in \Delta_M$ and $\mathbf{y} \in \Delta_N$, the expected loss of planner is

$$\sum_{i \in M} \sum_{j \in N} x_i y_j c_{ij} = \mathbf{x}^\intercal \mathbf{C} \mathbf{y}$$

In particular, if the adversary uses a pure strategy $\mathbf{e}_j \in \Delta_N$ with $j \in N$,

$$\sum_{i \in M} x_i c_{ij} = \mathbf{x}^\intercal \mathbf{C} \mathbf{e}_j$$

Minmax/Maxmin in Mixed Strategies

1. A minmax strategy of the planner is a mixed strategy $\bar{\mathbf{x}} \in \Delta_M$ such that

$$\max_{\mathbf{y}\in\Delta_N} \bar{\mathbf{x}}^\top \mathbf{C}\mathbf{y} = \min_{\mathbf{x}\in\Delta_M} \max_{\mathbf{y}\in\Delta_N} \mathbf{x}^\top \mathbf{C}\mathbf{y}$$

2. A maxmin strategy of the adversary is a mixed strategy $\bar{\mathbf{y}} \in \Delta_N$ such that

$$\min_{\mathbf{x}\in\Delta_M} \mathbf{x}^\top \mathbf{C}\bar{\mathbf{y}} = \max_{\mathbf{y}\in\Delta_N} \min_{\mathbf{x}\in\Delta_M} \mathbf{x}^\top \mathbf{C}\mathbf{y}$$

Then

$$\underbrace{\max_{\mathbf{y}\in\Delta_N} \min_{\mathbf{x}\in\Delta_M} \mathbf{x}^\top \mathbf{C}\mathbf{y}}_{\text{The lower bound on the reward}} \leq \bar{\mathbf{x}}^\top \mathbf{C}\bar{\mathbf{y}} \leq \underbrace{\min_{\mathbf{x}\in\Delta_M} \max_{\mathbf{y}\in\Delta_N} \mathbf{x}^\top \mathbf{C}\mathbf{y}}_{\text{The upper bound on the loss}}$$

Minimax Theorem

von Neumann, 1928

$$\underbrace{\min_{\mathbf{x}\in\Delta_M}\max_{\mathbf{y}\in\Delta_N} \mathbf{x}^\intercal\mathbf{C}\mathbf{y}}_{\text{The value of the game}} = \max_{\mathbf{y}\in\Delta_N}\min_{\mathbf{x}\in\Delta_M} \mathbf{x}^\intercal\mathbf{C}\mathbf{y}$$

- An equilibrium is a pair of minmax/maxmin strategies $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$
- For any equilibrium $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, we obtain

$$\bar{\mathbf{x}}^\intercal\mathbf{C}\bar{\mathbf{y}} = \text{the value of the game}$$

Computing Minmax Strategy

👉 Linear programming

The inner max can be evaluated using pure strategies only:

$$\min_{\mathbf{x}\in\Delta_M} \max_{\mathbf{y}\in\Delta_N} \mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{y} = \min_{\mathbf{x}\in\Delta_M} \max_{j\in N} \mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{e}_j.$$

Thus, we obtain a convex optimization problem for the planner, which is equivalent to the linear program with variables x_i ($i \in M$) and v :

$$\begin{aligned} \text{Minimize} \quad & \max_{j\in N} \mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{e}_j \\ \text{subject to} \quad & \mathbf{x} \in \Delta_M \end{aligned}$$

$$\begin{aligned} \text{Minimize} \quad & v \\ \text{subject to} \quad & \mathbf{x}^\mathsf{T}\mathbf{C}\mathbf{e}_j \le v, \quad \forall j \in N \\ & x_i \ge 0, \quad \forall i \in M \\ & \sum_{i\in M} x_i = 1 \end{aligned}$$

Computing Minmax Strategy

Example

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 4 & -1 \\ -1 & 1 & -2 & 5 \end{bmatrix}$$

$$\begin{aligned} \text{Minimize} \quad & v \\ \text{subject to} \quad & x_1 - x_2 \leq v \\ & x_2 \leq v \\ & 4x_1 - 2x_2 \leq v \\ & -x_1 + 5x_2 \leq v \\ & x_1, x_2 \geq 0 \\ & x_1 + x_2 = 1 \end{aligned}$$

The equilibrium strategies are $\bar{\mathbf{x}} = (\frac{7}{12}, \frac{5}{12})$, $\bar{\mathbf{y}} = (0, 0, \frac{1}{2}, \frac{1}{2})$, and the value is $\bar{v} = \frac{3}{2}$.

- The strategy sets M and N are too large in the path planning problems
- The set of paths and the loss matrix may not be given a priori

We show an iterative method relying on 2 principles:

1. Small subgames can be solved efficiently
2. Subgames are expanded with best responses

The best response of planner to a mixed strategy $\mathbf{y} \in \Delta_N$ is a strategy $i' \in M$ such that

$$\min_{i \in M} \mathbf{e}_i^\intercal \mathbf{C}\mathbf{y} = \mathbf{e}_{i'}^\intercal \mathbf{C}\mathbf{y}.$$

Double Oracle Algorithm

Double Oracle Algorithm

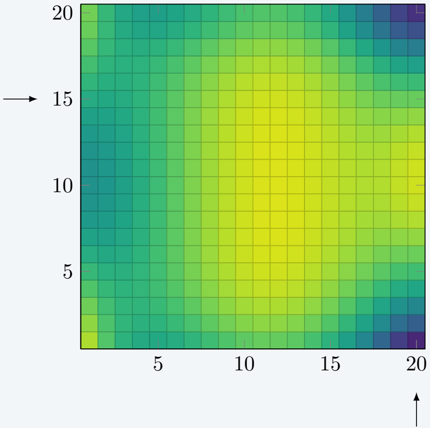
Blum et al. (2003)

1. Pick initial subsets of strategies for each player
2. Compute an equilibrium of the subgame — MASTER PROBLEM
3. Expand the current strategy sets with the best responses — SUB-PROBLEM
4. Repeat 2. and 3. until the current equilibrium is good enough

Double Oracle Algorithm

Initialize

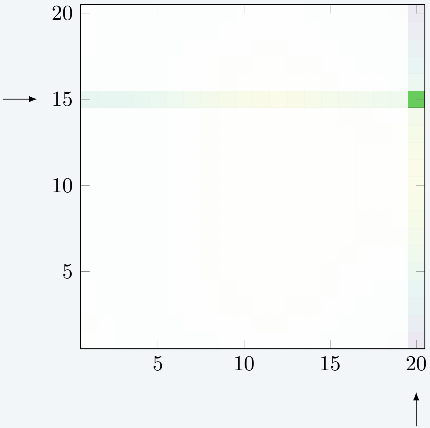
Initialize with random pure strategies.



Double Oracle Algorithm

Master Problem

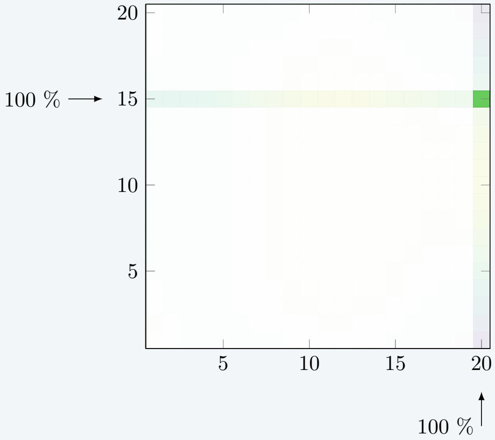
Find an equilibrium of the 1×1 subgame.



Double Oracle Algorithm

Master Problem

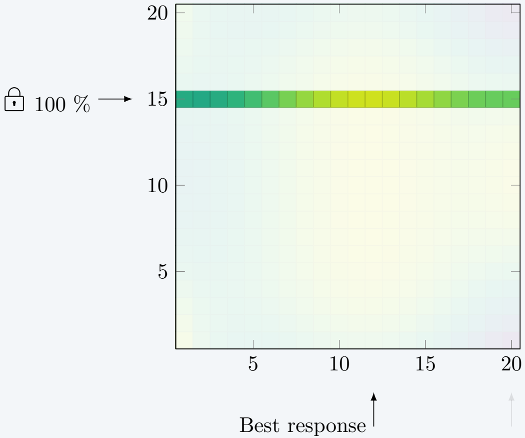
Find an equilibrium of the 1×1 subgame.



Double Oracle Algorithm

Find adversary's best response against a fixed strategy of the planner.

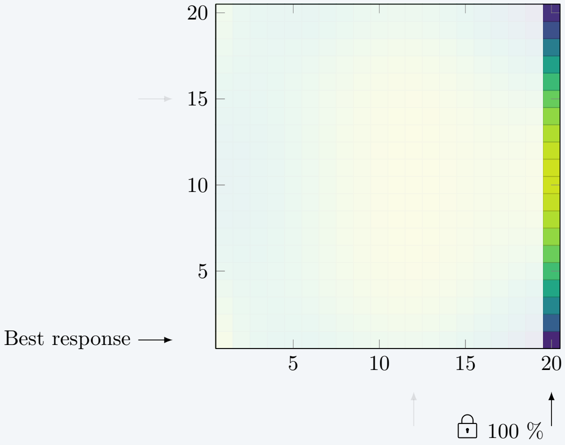
Best Response (adversary)



Double Oracle Algorithm

Find planner's best response against a fixed strategy of the adversary.

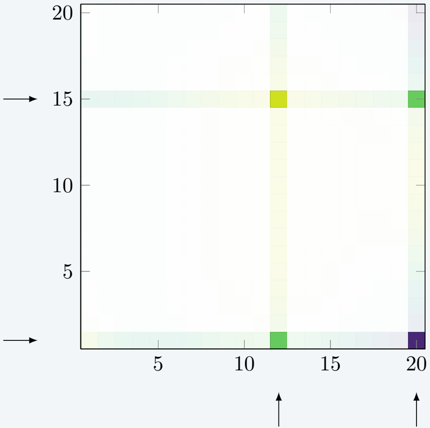
Best Response (planner)



Double Oracle Algorithm

Master Problem (Iteration 2)

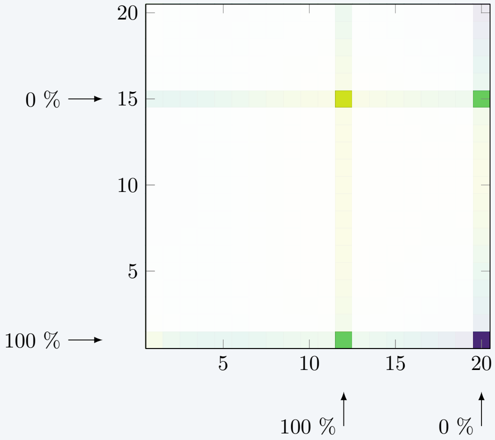
Find an equilibrium of the 2×2 subgame.



Double Oracle Algorithm

Find an equilibrium of the 2×2 subgame.

Master Problem (Iteration 2)



- The algorithm recovers an exact equilibrium in finitely many steps
- The approximation of equilibrium/value of the game
- Easy to implement using efficient LP solvers
- It may need $O(|M| + |N|)$ iterations

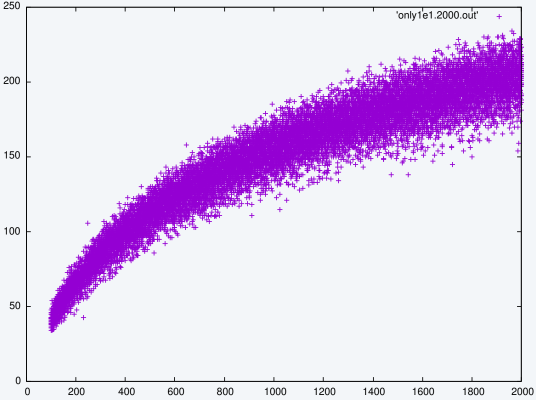
Let $\epsilon > 0$ and $\mathbf{C}_k$ be the matrix corresponding to the subgame in iteration k.

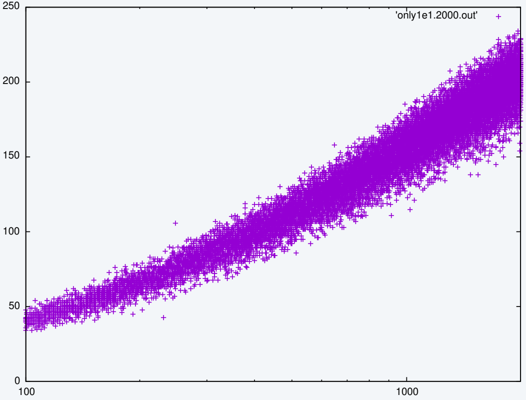
- An equilibrium of the subgame with matrix $\mathbf{C}_k$ is $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$
- Let i_{k+1} and j_{k+1} be the best responses to $\bar{\mathbf{y}}_k$ and $\bar{\mathbf{x}}_k$, respectively, in the original game with matrix $\mathbf{C}$
- Let $\mathbf{c}_{i_{k+1}}$ and $\mathbf{b}_{j_{k+1}}$ be the i_{k+1}-th row and j_{k+1}-th column of $\mathbf{C}$, respectively
- The upper bound on the value of the game is $\mathbf{c}_{i_{k+1}}^T \bar{\mathbf{y}}_k$
- The lower bound on the value of the game is $\bar{\mathbf{x}}_k^T \mathbf{b}_{j_{k+1}}$

One possible stopping condition is that the difference between the upper and lower bound is $< \epsilon$, which guarantees that $(\bar{\mathbf{x}}_k, \bar{\mathbf{y}}_k)$ is an ϵ-equilibrium.

Convergence to 0.1-equilibrium

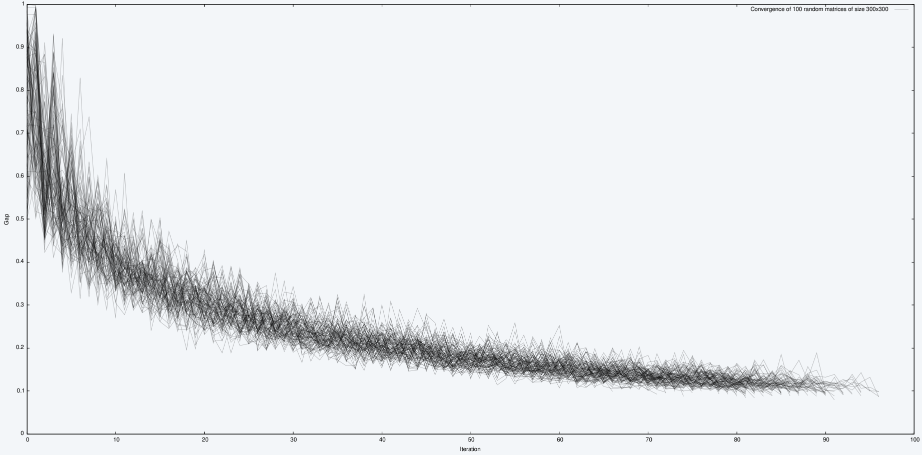
👉 The size of matrix vs #iterations





Convergence to 0.1-equilibrium

👉 #iterations vs convergence criterion for 300×300 games



References

- Chung, Timothy H., Geoffrey A. Hollinger, and Volkan Isler. Search and Pursuit-Evasion in Mobile Robotics. *Autonomous Robots* 31 (4): 299–316, 2011.
- McMahan, H. Brendan, Geoffrey J. Gordon, and Avrim Blum. Planning in the Presence of Cost Functions Controlled by an Adversary. In *Proceedings of the 20th International Conference on Machine Learning* (ICML-03), 2003.