Game theory - lab 1

David Milec

Czech Technical University in Prague

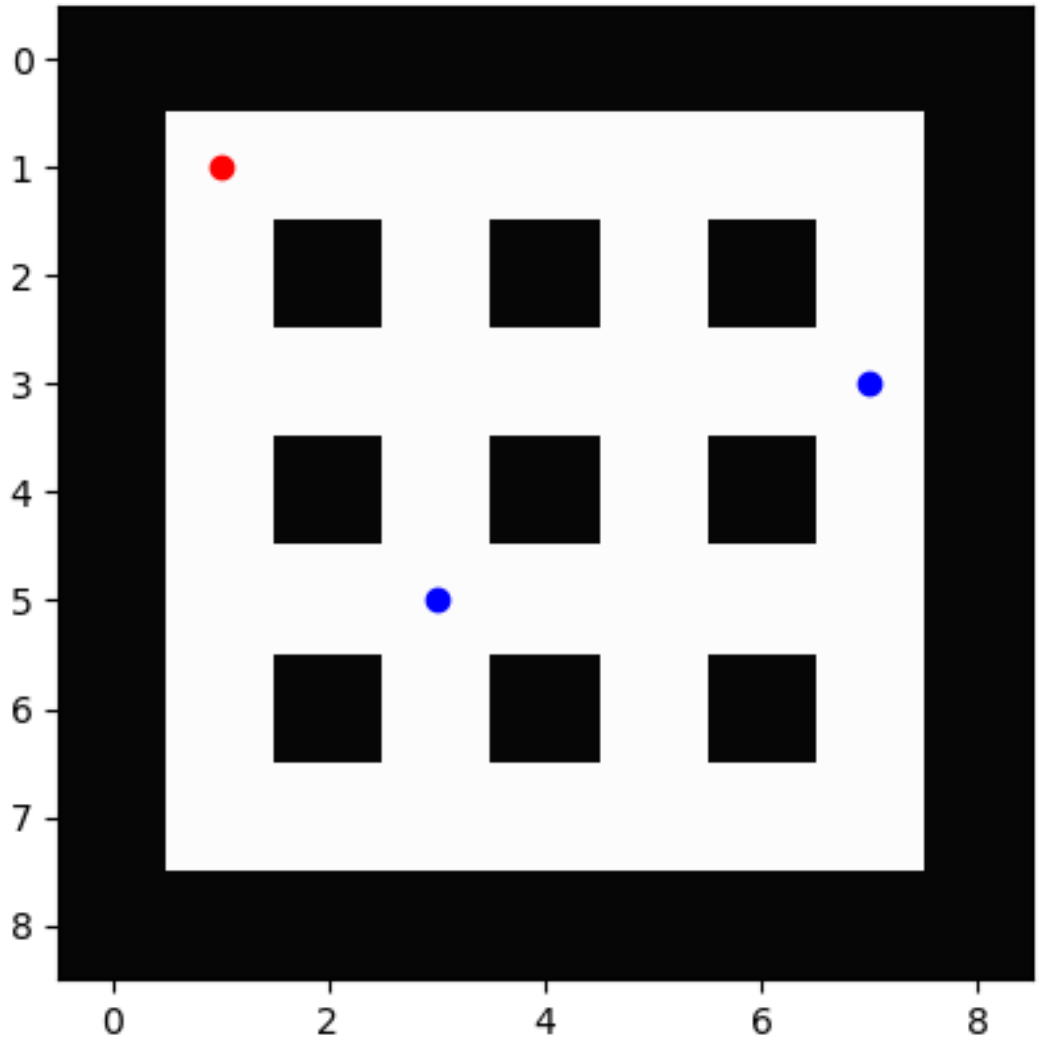
milecdav@fel.cvut.cz

November 29, 2019

Overview

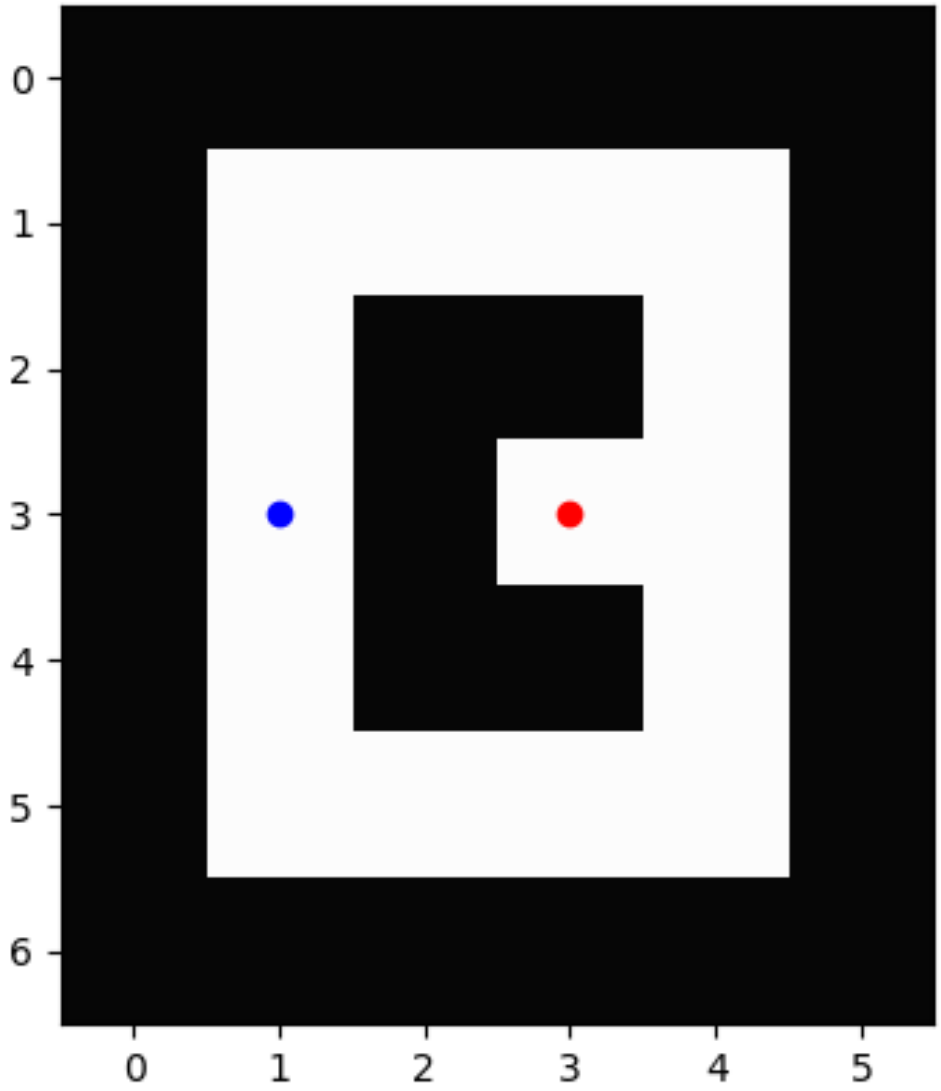
Pursuit Evasion Game

- Grid environment
- Simultaneous discrete moves
- Perfect information
- Evader (red) gets payoff for escaping for fixed amount of steps
- Pursuers (blue) get payoff for catching evader



Heuristic approaches

- Doing a move in such a way that I end in space closest/furthest to/from the opponent
- Euclidean distance does not work for pursuer even against stationary opponent
- Closest path is better but does not work with more pursuers in circular environment



Getting Close to Optimal Strategy

Monte Carlo tree search

- Explores the possible action tree in a way that tries to balance exploration and exploitation
- When the node is visited for the first time, evaluate using heuristic/rollout
- Save the value received from rollout, update all nodes up to the tree and go again from the root
- Selects the nodes to visit based on the values from previous rollouts

Monte Carlo in an Image

Graphical example of the steps performed in one Monte Carlo tree search update

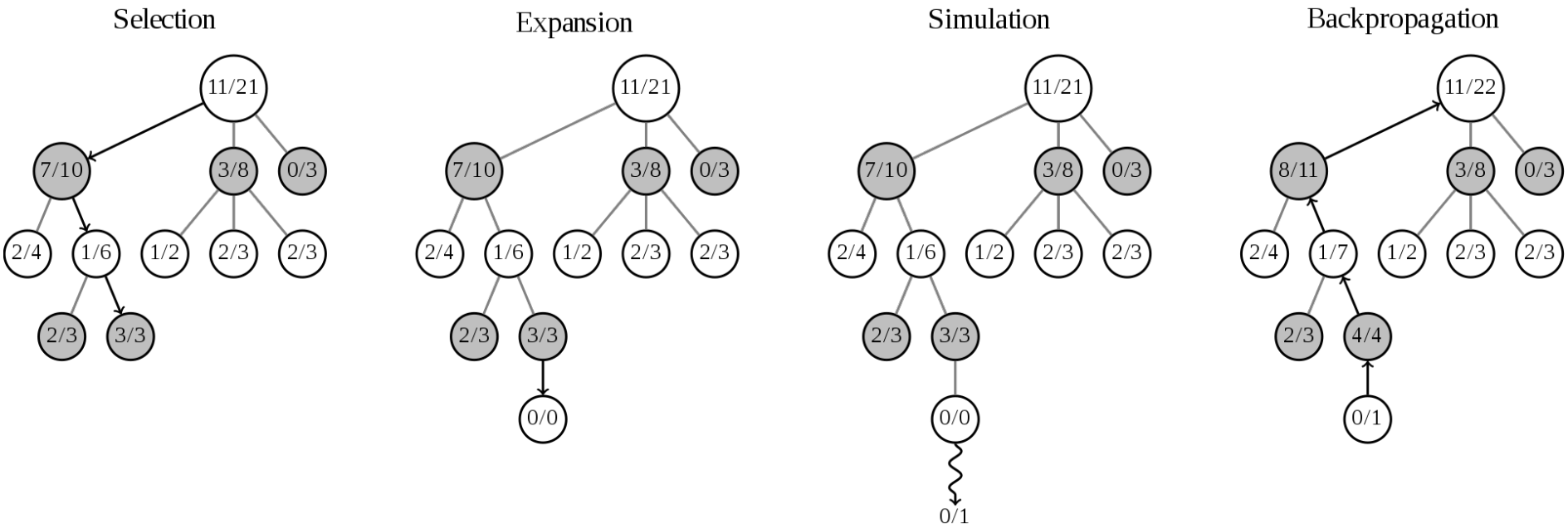


Image from
`https://en.wikipedia.org/wiki/Monte_Carlo_tree_search`

Asymptotically Optimal Strategy

Value iteration

- Adaptation of algorithm used to solve MDPs
- Stores values for all possible states of the game
- Iteratively updates those values based on possible actions in each state, solving matrix game created from next state values
- In the end uses the computed values to computed best strategy

Value Iteration

S is the state space, $v : S \to \mathbb{R}$ is value in each state, $\mathcal{A}$ is set of all combinations of actions and $A : S \to \mathcal{A}$ is a function returning all possible action tuples available in a given state. Q is a matrix game created for each state in each iteration, $r : S \times \mathcal{A} \to \mathbb{R}$ is immediate payoff and $T : S \times \mathcal{A} \to S$ is a transition function. γ is discounting constant.

$$\forall s \in S \quad \text{initialize} \quad v(s) = 0 \quad \text{and until v converges}$$

$$\forall s \in S$$

$$\forall (a_1, a_2) \in A(s)$$

$$Q(a_1, a_2) = r(s, a_1, a_2) + \gamma v(T(s, a_1, a_2))$$

$$v(s) = \max_x \min_y xQy$$

Project

Patrolling polygonal environment

- Polygon with some fence and we have n robots to guard it
- Robots go around the fence and attacker picks a spot on the fence and attack, going through takes him some time t
- Optimal strategy is to have robots uniformly, facing all the same direction and some probability of turning (Multi-Robot Adversarial Patrolling: Facing a Full-Knowledge Opponent, Noa, 2014)
- Your task is to measure the times, compute the probability and deploy the robots

The End