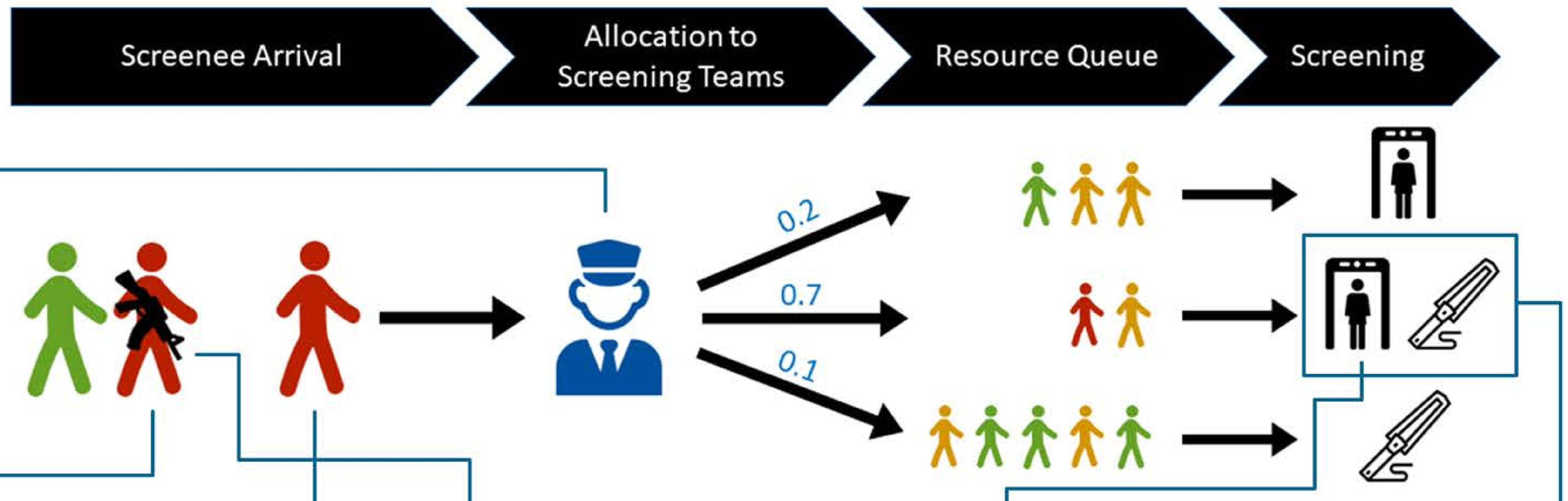
Solving Online Threat Screening Games using Constrained Action Space Reinforcement Learning

Sanket Shah[1], Arunesh Sinha[1], Pradeep Varakantham[1], Andrew Perrault[2], Milind Tambe[2]

[1]Singapore Management University, [2]Harvard University

Problem Description

Aim: *Find the best Defender mixed strategy in this Stackelberg Security Game*



Attacker Utility *U*
Metric attacker wants to maximise
eg: Number of Fatalities
*The **Security Risk** is the expected utility of the attacker*

Delay
Time taken to get to front of the queue
For a security team, delay is max of delay across constituent resources

Defender
Goes first. Chooses a mixed strategy that minimises a combination of Security Risk and Delay.

Attacker
Goes second. Impersonates a screenee to maximise utility. *Can be thought of as the worst-case among screenees that arrive.*

Screenees
People to be screened.
eg: Passengers
Types of Categories c:
Uncontrollable θ (eg: TSA pre-check), Controllable κ (eg: Flight)

Attack Methods m
Possible ways to attack
eg: guns, chemical weapons

Security Resources r
Ways to detect attack
eg: metal detector, K9 unit

Security Teams t
Combinations of resources
eg: Wand+Metal Detector
Properties: Efficacy
E—probability of detecting attack

Past Work & Motivation

- Multi-objective problem: Security Risk and Delay. Past work focuses on **minimizing Security Risk**
- **Limitations**:
  - Has a **roundabout notion of Delay** (encoded in the idea of 'time windows'). Because they can't effectively quantify Delays, generated policies are suboptimal.
  - **Modeled as offline optimization;** most threat screening is online
  - **Unscalable and pessimistic** approach to screenee arrival uncertainty

Contributions

- Turn problem on its head, find Pareto optimal point by **minimizing Delay** with a constraint on Security Risk.
- Allows us to model the game as an **MDP**, which addresses limitations:
  - MDP is online: uncertainty in screenee arrival is more natural
  - Minimize exact delay using rewards
- **Security Risk** bounded by linear constraints on action space
- Solve MDP using **Deep RL** with novel constraint enforcement strategy.

MDP Model

- **State**: Information about the current screenee along with context. Formally, consists of the tuple $<c, \xi, h, \tau>$. Here c is the screenee category, ξ is the queue information, h is a summary of the history and τ is the wallclock time.
- **Action**: Mixed strategy π, i.e., probability of assigning screenee to each security team. Probability of detection z is a linear function of this action and Security Risk, in turn, is a linear function of z.

$$z_m = \sum_{t \in T} E_{t,m}\pi_t$$

$$P_\theta(z_m U^+_{\kappa,m} + (1 - z_m)U^+_{\kappa,m}) \leq \psi_\theta, \forall m$$

As a result, security risk can be bounded by linear constraints on the actions.

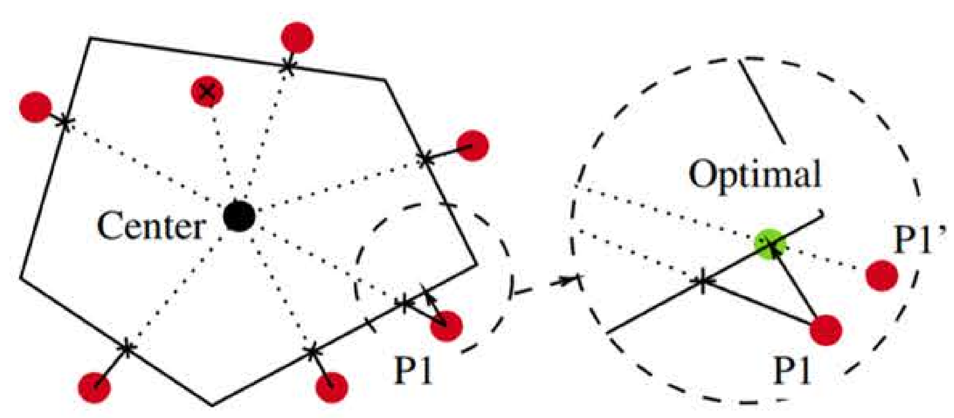
- **Reward**: -1 * Delay
- **Transitions**: Update context and get info about the next screenee

Overall Solution Method

- Solve using **Deep RL**: Deep Deterministic Policy Gradients (DDPG)
- Enforce constraints by appending **novel projection layer** to 'actor' network.

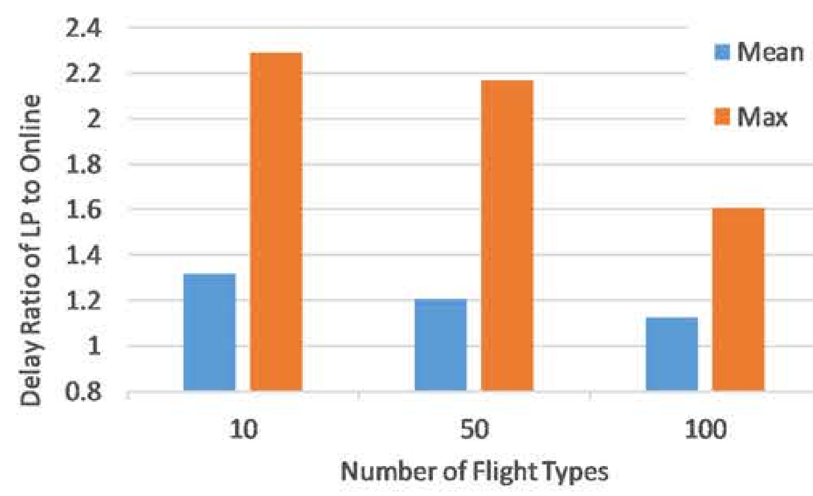
Enforcing Constraints

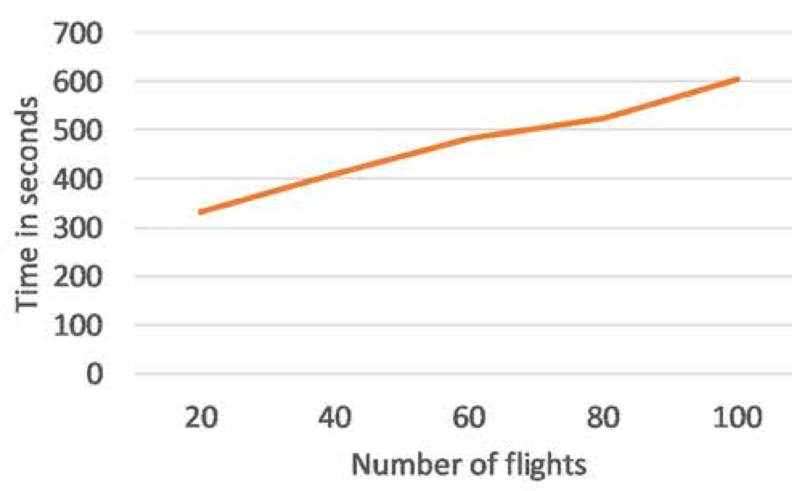
- Security domain; must enforce **hard constraints** on the security risk.
- Incorporate constraint enforcement as **layer in the network**
  - **Input**: unconstrained action; **Output**: constrained action
- **Past work**: QP that minimizes L2 distance between constrained and unconstrained actions and backprops through it
  - **Slow** (DDPG requires many iterations to converge)
- Our work: minimises distance from fixed internal point.
  - Also enforces constraints, runs quickly and can be written in few lines of TensorFlow



Experiments

- Up to 100% reduction in expected Delay; no increase in security risk





- Scales well and can be applied to real-world scenarios

Acknowledgments: *This research was supported by the Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2 grant MOE2016-T2-1-174, Ministry of Education Academic Research Fund (AcRF) Tier 1 grant 19-C220-SMU-011 and the Lee Kong Chian Fellowship awarded by Singapore Management University. This work was also sponsored by the Army Research Office and accomplished under MURI Grant Number W911NF-17-1-0370. Perrault was supported by the Center for Research on Computation and Society.*