Computing Optimal Randomized Resource Allocations for Massive Security Games

Christopher Kiekintveld, Manish Jain, Jason Tsai, James Pita, Fernando Ordonez, Milind Tambe

The Problem

The LAX canine problems deals with approximately 800 different assignments. However there are applications where thousands of resources and targets must be considered (100 targets + 10 resources = 1.7×10^{13} assignments).

We consider deploying Federal Air Marshals (FAMs) to different flights to deter terrorist attacks to gain control of a flight.

There are roughly 27,000 domestic flights and over 2,000 international flights daily.

The possibility alone of a FAM can deter terrorist activities. There is a need for an algorithm that can both randomize FAM scheduling and choose optimal targets to cover (so that terrorists cannot reliably predict which flight FAMs are on).

Stackelberg Security Game

In a Stackelberg game, there is a “leader” who moves first, and a “follower,” who observes the actions of the leader before acting.

In a Stackelberg security game, the “defender” is the Stackelberg leader and the “attacker” is the Stackelberg follower.

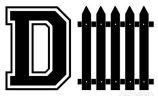
The Stackelberg equilibrium is a form of subgame perfect equilibrium.

We assume Strong Stackelberg Equilibria (the follower chooses the optimal strategy for the leader when he is indifferent towards his own strategy).

Compact Security Game Model

If we have n targets and m resources, there are n choose m different strategies.

In the model we assume that the payoff only depends on the identity of the attacked target and whether or not it is covered by the defender.



5 $U^c_\Theta(t)$	-20 $U^u_\Theta(t)$
-10 $U^c_\Psi(t)$	30 $U^u_\Psi(t)$

Now we have 4 payoffs to calculate for n targets. The total number of payoffs to calculate is 4n.

Defining Payoff

We define a coverage vector (C) that give the probability each target is covered.

The attack vector (A) is the probability of attacking a target (we restrict to attack a single target with a probability of one).

$$U_\Theta(C,A) = \sum_{t \in T} a_t \cdot \underbrace{(c_t \cdot U_\Theta^c(t) + (1-c_t)U_\Theta^u(t))}$$

$$U_\Theta(t,C) = c_t U_\Theta^c(t) + (1-c_t)U_\Theta^u(t)$$

$$\Gamma(C) = \{t : U_\Psi(t,C) \geq U_\Psi(t',C) \, \forall \, t' \in T\}$$

The Attack Set - the targets that yield the max payoff given the coverage vector C

ERASER Algorithm

ERASER stands for Efficient Randomized Allocation of SEcurity Games.

The algorithm takes in a compact form of the security game as input and solves for an optimal coverage vector (C) that is the Strong Stackelberg Equilibrium for the defender.

This is a mixed-integer linear program (MILP).

ERASER Algorithm

$$\max \quad d \qquad (5)$$

define objective function

$$a_t \in \{0,1\} \quad \forall t \in T \qquad (6)$$

$$\sum_{t \in T} a_t = 1 \qquad (7)$$

force an attack vector to assign a single target probability 1

$$c_t \in [0,1] \quad \forall t \in T \qquad (8)$$

$$\sum_{t \in T} c_t \leq m \qquad (9)$$

probability between 0 and 1
the probabilities must be bounded by the number of resources

Defender's expected payoff, contingent on the target attacked in A

$$d - U_{\Theta}(t, C) \leq \quad (1 - a_t) \cdot Z \quad \forall t \in T \qquad (10)$$

* upper bound on d when target is attacked

Attacker's expected payoff, contingent on the target attacked in A

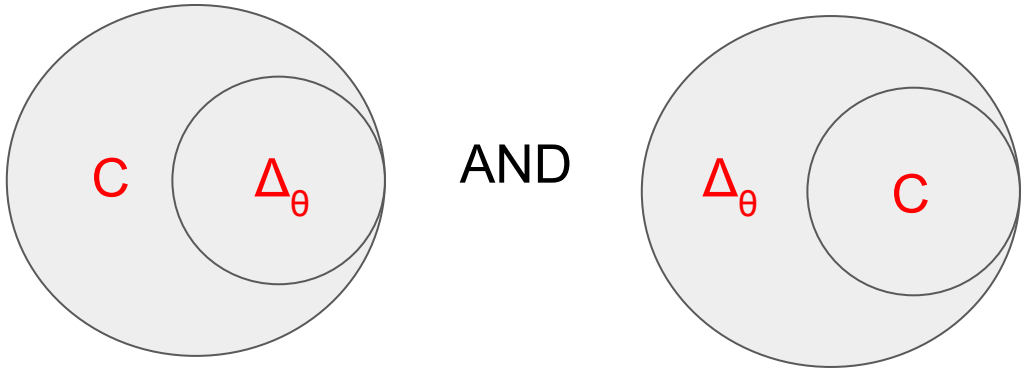
$$0 \leq k - U_{\Psi}(t, C) \leq \quad (1 - a_t) \cdot Z \quad \forall t \in T \qquad (11)$$

k must be at least as large as the maximal payoff for attacking any target

k has an upper bound of U_phi when the target is attacked

THEOREM 1. *For any feasible ERASER coverage vector, there is a corresponding mixed strategy δ_Θ that implements the desired coverage probabilities.*

We can translate the coverage vector C outputted by ERASER into a mixed strategy.



If ERASER can find a coverage vector, we can find the corresponding mixed strategy.

THEOREM 2. *A pair of attack and coverage vectors (C, A) is optimal for the ERASER MILP correspond to at least one SSE of the game.*

Consider any other coverage vector C'

Let t* be a target within the attack set for C' with maximal payoff for the defender

Let A' be the attack vector that places a probability of 1 on t*

$$U_\Theta(C', A') \leq U_\Theta(C, A)$$

Thus, the solution to MILP is a mutual best response, which satisfies the conditions of SSE.

ORIGAMI

We add two additional constraints to ERASER to restrict the payoff function:

$$U_{\Theta}^{u}(t) < U_{\Theta}^{c}(t)$$
$$U_{\Psi}^{u}(t) > U_{\Psi}^{c}(t)$$

These are very intuitive: defenders benefits from covered targets, attackers benefit from uncovered targets

Algorithm 1 ORIGAMI

targets $\leftarrow T$ sorted by $U^u_\Psi(t)$
payoff[t] $\leftarrow U^u_\Psi(t)$, coverage[t] $\leftarrow 0$
left $\leftarrow m$, next $\leftarrow 2$
covBound $\leftarrow -\infty$
while next $\leq n$ **do**
  addedCov[t] $\leftarrow \frac{payoff[next]-U^u_\Psi(t)}{U^c_\Psi(t)-U^u_\Psi(t)}$ - coverage[t]
  if coverage[t] + addedCov[t] ≥ 1 **then**
    covBound $\leftarrow$ Max(covBound, $U^c_\Psi(t)$)
  end if
  if covBound $\geq -\infty$ OR $\sum_{t\in T}$ addedCov[t] $\leq$ left **then**
    BREAK
  end if
  coverage[t] += addedCov[t]
  left -= $\sum_{t\in T}$ addedCov[t]
  next++
end while
ratio[t] $\leftarrow \frac{1}{U^u_\Psi(t)-U^c_\Psi(t)}$
coverage[t] += $\frac{ratio[t]\cdot left}{\sum_{t\in T} ratio[t]}$
if coverage[t] ≥ 1 **then**
  covBound $\leftarrow$ Max(covBound, $U^c_\Psi(t)$)
end if
if covBound $\geq -\infty$ **then**
  coverage[t] $\leftarrow \frac{covBound-U^u_\Psi(t)}{U^c_\Psi(t)-U^u_\Psi(t)}$
end if

We start with an attack set with the target that has the maximum uncovered payoff for the attacker.

The attack set is expanded at each iteration to add a target with a smaller payoff.

The coverage of each target is updated to maintain the indifference of the attacker payoffs within the attack set.

ORIGAMI MILP

$$\min \quad k \qquad (12)$$

$$\gamma_t \in \{0,1\} \quad \forall t \in T \qquad (13)$$

$$c_t \in [0,1] \quad \forall t \in T \qquad (14)$$

$$\sum_{t \in T} c_t \leq m \qquad (15)$$

$$U_\Psi(t, C) \leq k \quad \forall t \in T \qquad (16)$$

$$k - U_\Psi(t, C) \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \qquad (17)$$

$$c_t \leq \gamma_t \quad \forall t \in T \qquad (18)$$

This algorithm is very similar to ERASER except the attacker's payoff is minimized, rather than the defender's payoff be maximized.

There is an added constraint that restricts c_t to 0 for any t not in the attack set.

ERASER-C

This is an extension of ERASER that adds the capability to represent resource and scheduling constraints.

Resources can be assigned to schedules covering multiple targets.

There are different resource types, each with a capability to cover a different subset of the schedule S.

This provides a FAM with a feasible schedule.

$$\max \quad d \qquad (19)$$

$$a_t \in \{0,1\} \quad \forall t \in T \qquad (20)$$

$$c_t \in [0,1] \quad \forall t \in T \qquad (21)$$

$$q_s \in [0,1] \quad \forall s \in S \qquad (22)$$

$$h_{s,\omega} \in [0,1] \quad \forall s,\omega \in S \times \Omega \qquad (23)$$

$$\sum_{t \in T} a_t = 1 \qquad (24)$$

$$\sum_{\omega \in \Omega} h_{s,\omega} = q_s \quad \forall s \in S \qquad (25)$$

$$\sum_{s \in S} q_s M(s,t) = c_t \quad \forall t \in T \qquad (26)$$

$$\sum_{s \in S} h_{s,\omega} Ca(s,\omega) \leq \mathcal{R}(\omega) \quad \forall \omega \in \Omega \qquad (27)$$

$$h_{s,\omega} \leq Ca(s,\omega) \quad \forall s,\omega \in S \times \Omega \qquad (28)$$

$$d - U_\Theta(t,C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \qquad (29)$$

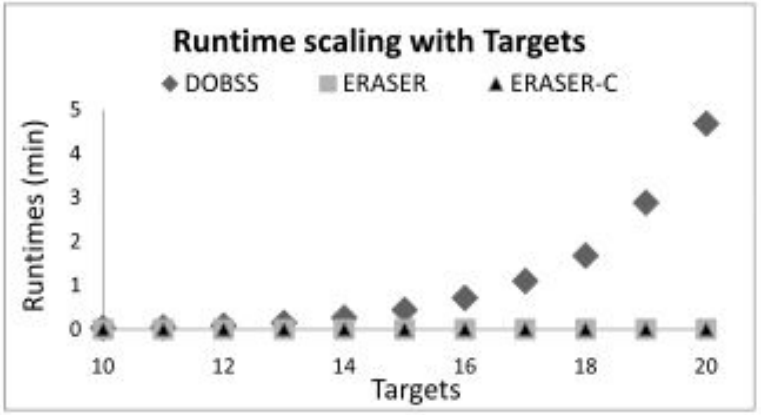
$$0 \leq k - U_\Psi(t,C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \qquad (30)$$

MANY MORE CONSTRAINTS THAT NEED TO BE SATISFIED!

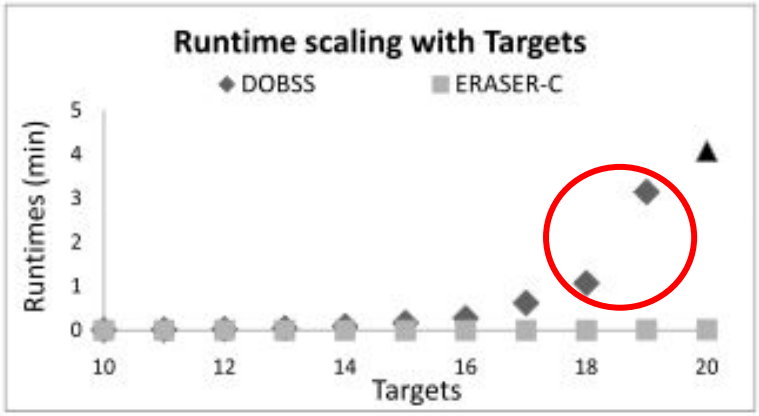
Experimental Evaluation

The four algorithms were compared to the existing method DOBSS for both randomly-generated security games and real data.

All methods generate optimal SSE solutions, so computation time and memory usage were the comparable metrics.

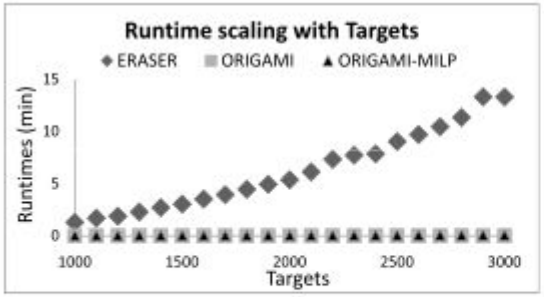


DOBSS is exponential in both computational time and memory (memory limitations tend to restrict the algorithm first)

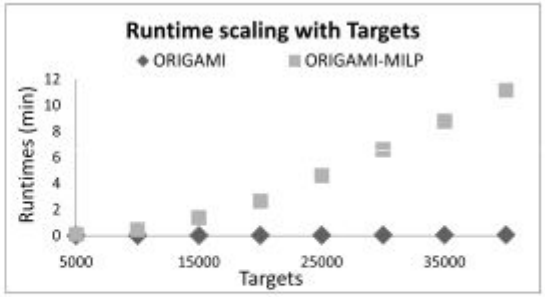


There is no significant difference between ERASER and ERASER-C for < 20 targets. There is a statistically significant difference for > 20 targets.

Compare algorithms on very large games beyond the limits of DOBSS.



25 resources
3,000 targets



1,000 resources
40,000 targets

Conclusion: the size of the games scales to very large instances, especially for the ORIGAMI algorithm.

The final experiment involved testing the algorithms on real data from the LAX canine and FAM scheduling domains.

	Actions	DOBSS	ERASER (-C)
LAX (6 canines)	784	0.94s	0.23s
FAMS (small)	~6,000	4.74s	0.09s
FAMS (large)	~85,000	435.6s*	1.57s

**DOBSS was not able to complete the large FAMS problem due to memory limitations.*

ERASER (-C) performed significantly better than DOBSS on real data sets (in addition to the randomly generated data sets discussed before) .

Discussion Questions

The paper makes several assumptions involving payoff calculations, independence of targets, resource allocations, etc. Are these valid assumptions? Are there any that we may be hesitant to accept?

What different factors do you think might be considered in determining the payoff (i.e. how do you think flight risk is evaluated)?

How would coordinating multiple attackers affect the game? Does this increase the complexity of the game?