MECHANISM DESIGN

JOHN P DICKERSON & MARINA KNITTEL

Lecture #12 – 03/07/2022
Lecture #13 – 03/09/2022

CMSC498T
Mondays & Wednesdays
2:00pm – 3:15pm

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

PROJECT PROPOSALS

We'd like you to submit a 1-2 pager covering an initial plan for your course project by the end of next week (Fri March 18).

How to submit:

- We'll open up an ELMS assignment
- Feel free to create groups (highly encouraged!)
- Group finding: in class, Piazza, etc

You will get 100% for this if you submit something "okay" – this is just to kickstart (i) movement and (ii) discussion between us

PROJECT PROPOSALS: A SUGGESTION

Consider a 75%/100%/125% set of goalposts:

Project Plan:

75% goals

- Create and train 3 regressor system for electrical energy consumption dataset.
- Design the adaptive learning algorithm.

100% goals

- Implement the adaptive learning algorithm.
- Apply the algorithm to forecasting electrical energy consumption in the United States problem.
- Compare its performance with baselines which are:
  - Single regressor agent.
  - Multi-agents with equal weights.

125% goals

- Compare this algorithm performance against other techniques used to improve long horizon forecast.
- Test this algorithm performance on other forecasting problems including a forecasting brain ventricular volume as a biomarker for neurodegenerative disease progression.
- Test performance on other decision making problems that are unrelated to forecasting.

[Thanks, Aya Ismail! S2018 CMSC828M]

THIS CLASS: STACKELBERG & SECURITY GAMES

Thanks to: AGT book, Conitzer (VC), Gupta (NG), Procaccia (AP)

SIMULTANEOUS PLAY

Previously, assumed players would play simultaneously

- **Two drivers simultaneously decide to go straight or divert**
- **Two prisoners simultaneously defect or cooperate**
- **Players simultaneously choose rock, paper, or scissors**
- **Etc …**

No knowledge of the other players' chosen actions

What if we allow sequential action selection ...?

LEADER-FOLLOWER GAMES

Heinrich von Stackelberg

Two players:

- **The leader commits to acting in a specific way**
- **The follower observes the leader's mixed strategy**

NE, iterated strict dominance

Commit to "Bottom"	
0, 0	2, 1

What is the Nash equilibrium ????????

- **Social welfare: 2**
- **Utility to row player: 1**

Row player = leader; what to do ????????

- **Social welfare: 3**
- **Utility to row player: 2**

ASIDE: FIRST-MOVER ADVANTAGE (FMA)

From the econ side of things …

- **Leader is sometimes called the Market Leader**
- **Some advantage allows a firm to move first:**
    - Technological breakthrough via R&D
    - Buying up all assets at low price before market adjusts

By committing to a strategy (some amount of production), can effectively force other players' hands.

Things we won't model:

- Significant cost of R&D, uncertainty over market demand, initial marketing costs, etc.

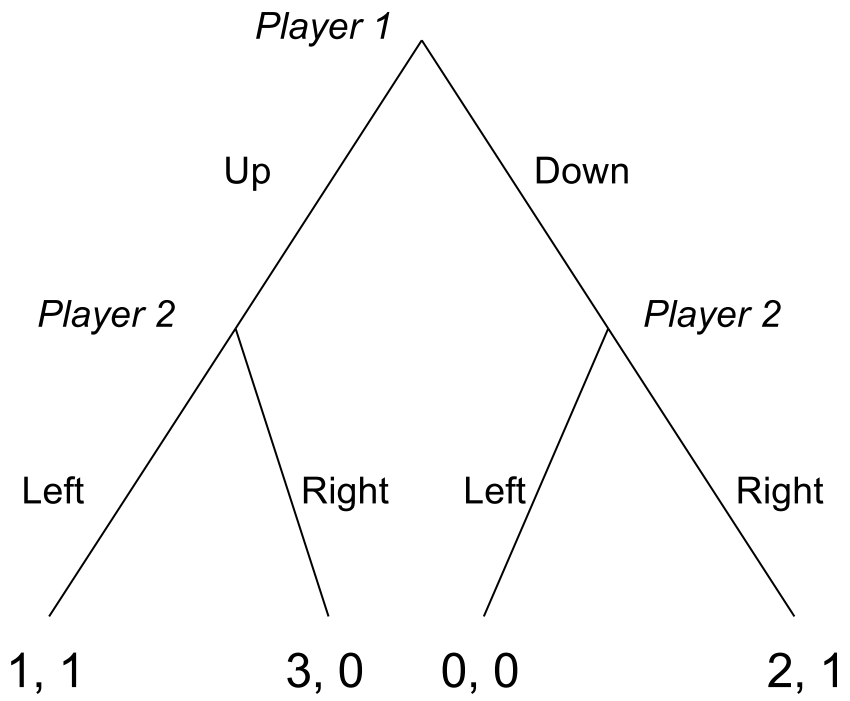
These can lead to Second-Mover Advantage

- **Atari vs Nintendo, MySpace (or earlier) vs Facebook**

COMMITMENT AS AN EXTENSIVE-FORM GAME

1, 1	3, 0
0, 0	2, 1

For the case of committing to a pure strategy:



COMMITMENT TO MIXED STRATEGIES

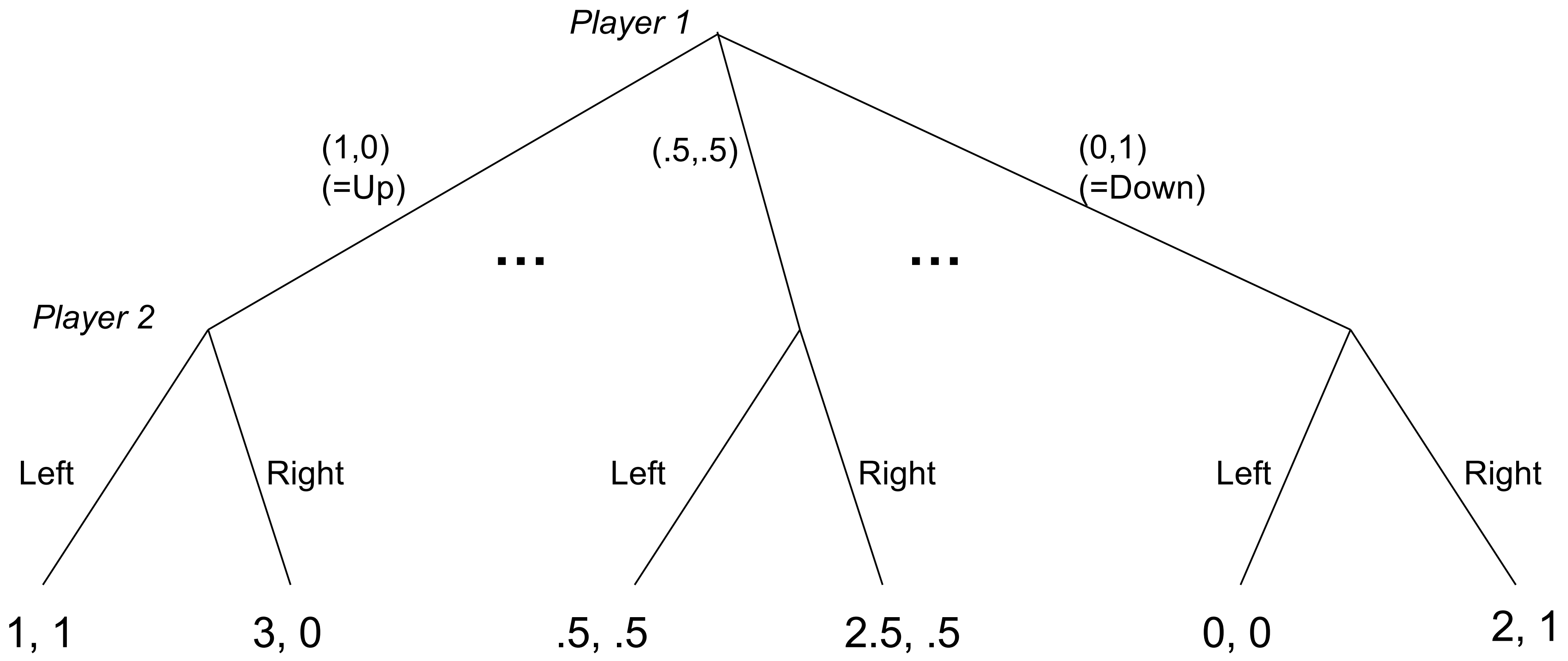
	0	1
.49	1, 1	3, 0
.51	0, 0	2, 1

What should Column do ????????

Sometimes also called a Stackelberg (mixed) strategy

COMMITMENT AS AN EXTENSIVE-FORM GAME…

For the case of committing to a mixed strategy:



- **Economist: Just an extensive-form game …**
- **Computer scientist: Infinite-size game! Representation matters**

VC

WHAT SHOULD THE LEADER COMMIT TO?

2-P
Z-S

Special case: 2-player zero-sum normal-form games

Recall: Row player plays Minimax strategy

- **Minimizes the maximum expected utility to the Col**
- **Minimax utility: $\min_{\sigma_{-i}} \max_{s_i} u_i(s_i, \sigma_{-i})$**

Doesn't matter who commits to what, when

Minimax strategies = Nash Equilibrium

= Stackelberg Equilibrium

(not the case for general games)

Polynomial time computation via LP – earlier lectures

WHAT SHOULD THE LEADER COMMIT TO?

2-P
G-S

Strong Stackelberg Equilibrium (SSE): follower breaks ties in favor of the leader

Theorem [Conitzer & Sandholm]: In 2-player, general-sum normal-form games, an SSE can be found in polytime

- **?????????????**

Idea:

- **Iterate over every follower pure strategy aka column c**
- **Compute a mixed strategy r for leader such that playing pure strategy c is a best response for follower**
- **Choose r*, the best (aka highest value for leader) mixed strategy amongst those strategies!**

[Conitzer & Sandholm, Computing the optimal strategy to commit to, EC-06]

WHAT SHOULD THE LEADER COMMIT TO?

2-P
G-S

Separate LP for every column c^*:

$$\textit{maximize } \Sigma_r\, p_r\, u_R(r, c^*)$$

s.t.

for all c, $\Sigma_r\, p_r\, u_C(r, c^*) \geq \Sigma_r\, p_r\, u_C(r, c)$

$\Sigma_r\, p_r = 1$

for all r, $p_r \geq 0$

Row utility

Column optimality
aka Col best response

Distributional constraints

Choose strategy from LP with highest objective

[Conitzer & Sandholm, Computing the optimal strategy to commit to, EC-06]

RUNNING EXAMPLE

x	1, 1	3, 0
y	0, 0	2, 1

maximize $1x + 0y$

s.t.

$1x + 0y \geq 0x + 1y$

$x + y = 1$

$x \geq 0$

$y \geq 0$

maximize $3x + 2y$

s.t.

$0x + 1y \geq 1x + 0y$

$x + y = 1$

$x \geq 0$

$y \geq 0$

IS COMMITMENT ALWAYS GOOD FOR THE LEADER?

Yes, if we allow commitment to mixed strategies

- Always weakly better to commit [von Stengel & Zamir, 2004] **??????**
- If (r*, c) is Nash, then Row can always commit to r* → Col will play c*, can achieve value of that equilibrium

What about only pure strategies?

	Rock	Paper	Scissors
Rock			
Paper	+1,-1	0,0	-1,+1
Scissors			

Expected utility to Row by playing mixed Nash: ???????????

$$E_R[\langle 1/3,1/3,1/3 \rangle] = 0$$

Expected utility to Row by any pure commitment: ???????????

$$E_R[\langle 1,0,0 \rangle] = -1$$
$$E_R[\langle 0,1,0 \rangle] = -1$$
$$E_R[\langle 0,0,1 \rangle] = -1$$

WHAT SHOULD THE LEADER COMMIT TO?

Bayesian 2-P G-S

Bayesian games: player *i* draws type θ_i from Θ

Special case: follower has only one type, leader has type θ

Like before, solve a separate LP for every column c*:

$$\textit{maximize } \Sigma_\theta\, \pi(\theta)\ \Sigma_r\, p_{r,\theta}\, u_{R,\theta}(r, c^*)$$

$$\textit{s.t.}$$

$$\textit{for all } c,\ \Sigma_\theta\, \pi(\theta)\, \Sigma_r\, p_{r,\theta}\, u_C(r, c^*) \geq \Sigma_\theta\, \pi(\theta)\, \Sigma_r\, p_{r,\theta}\, u_C(r, c)$$

$$\textit{for all } \theta,\ \Sigma_r\, p_{r,\theta} = 1$$

$$\textit{for all } r, \theta,\ p_{r,\theta} \geq 0$$

Choose strategy from LP with highest objective

WHAT SHOULD THE LEADER COMMIT TO?

Bayesian
N-P G-S

So, we showed polynomial-time **methods for:**

- **2-Player, zero-sum**
- **2-Player, general-sum**
- **2-Player, general-sum, Bayesian with 1-type follower**

In general, NP-hard **to compute:**

- **2-Player, general-sum, Bayesian with 1-type leader**
  - Arguably more interesting (“I know my own type”)
- **2-Player, general-sum, Bayesian general**
- ***N*-Player, for *N* > 2:**
  - 1st player commits, *N*-1-Player leader-follower game, 2nd player commits, recurse until 2-Player leader-follower

STACKELBERG SECURITY GAMES

Leader-follower → Defender-attacker

- Defender is interested in protecting a set of targets
- Attacker wants to attack the targets

The defender is endowed with a set of resources

- Resources protect the targets and prevent attacks

Utilities:

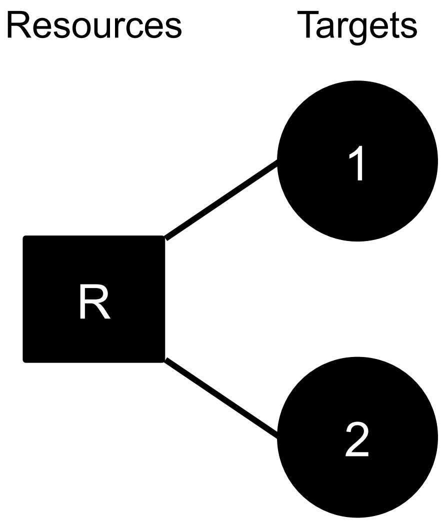
- Defender receives positive utility for preventing attacks, negative utility for "successful" attacks
- Attacker: positive utility for successful attacks, negative otherwise
- Not necessarily zero-sum

SECURITY GAMES: A FORMAL MODEL

Defined by a 3-tuple (N, U, M):

- **N: set of *n* targets**
- **U: utilities associated with defender and attacker**
- **M: all subsets of targets that can be simultaneously defended by deployments of resources**
  - A schedule $S \subseteq 2^N$ is the set of target defended by a single resource *r*
  - Assignment function $A : R \rightarrow 2^S$ is the set of all schedules a specific resource can support
- **Then we have *m* pure strategies, assigning resources such that the union of their target coverage is in M**
- **Utility $u_{c,d}(i)$ and $u_{u,d}(i)$ for the defender when target i is attacked and is covered or defended, respectively**

SIMPLE EXAMPLE



Targets	Defender		Attacker Type θ_1		Attacker Type θ_2	
i	**$u_{c,d}(i)$**	**$u_{u,d}(i)$**	**$u_{c,a}(i)$**	**$u_{u,a}(i)$**	**$u_{c,a}(i)$**	**$u_{u,a}(i)$**
1	0	-1	0	+1	0	+1
2	0	-2	0	+5	0	+1

[Blum, Haghtalab, Procaccia, Learning to Play Stackelberg Security Games, 2016]

REAL-WORLD SECURITY GAMES

Lots of deployed applications!

- Checkpoints at airports
- Patrol routes in harbors
- Scheduling Federal Air Marshalls
- Patrol routes for anti-poachers

Typically solve for strong Stackelberg Equilibria:

- Tie break in favor of the defender; always exists
- Can often “nudge” the adversary in practice

Two big practical problems: computation and uncertainty

USC University of Southern California

Duke University

Bar-Ilan University

University of Maryland

Carnegie Mellon

OVERVIEW OF AN IMPACTFUL PAPER IN THIS SPACE

[Kiekintveld et al. 2009]

Computing Optimal Randomized Resource Allocations for Massive Security Games (linked on course webpage)

- Motivated first by resource assignment for checkpoints at LAX, e.g., multiple canine units assigned to cover multiple terminals …
- … and later by much larger games such as Federal Air Marshals Service assignments and port inspection.

m resources to cover n targets, m < n

Defender (leader) commits to a mixed strategy

Attacker (follower) observes the probabilities for each coverage set

- Surveillance, insider threat, etc – maybe not perfectly realistic

Attacker chooses a pure strategy

Equilibrium concept not ex post

OVERVIEW OF AN IMPACTFUL PAPER IN THIS SPACE [Kiekintveld et al. 2009]

Initially assume interchangeable resources (extended in paper, will cover in a few slides by introducing "types")

Assume players are risk neutral

One type of follower (attacker)

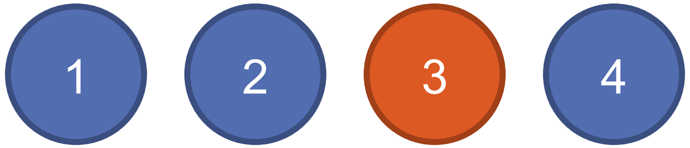
- Recall: one type of follower → PTIME solvable, one LP solved for each pure strategy of follower …
- … but the number of pure strategies in some games might be large, e.g., with 100 targets and 10 resources, 1.7×10^{13}!

RUNNING EXAMPLE

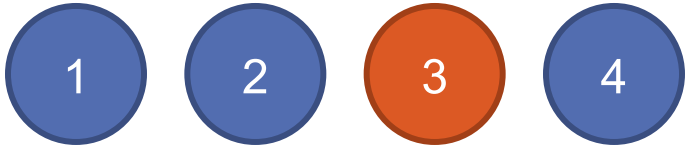
4 targets, 2 resources

Qualitatively:

- Defender values all 4 targets equally (and prefers a covered attack to an uncovered attack).
- Attacker gets twice as much utility for successful attack on target 3. All failed attacks get the same (lower) utility.



MOTIVATION AND INTRODUCTION



"Utility for leader θ if the target 3 is attacked and it is covered (c) or uncovered (u)"

$u^c_\Theta(3)$ $u^u_\Theta(3)$

Targets {1, 2, 4}		
	Covered	Uncovered
Defender	4	1
Attacker	0	1

Target 3		
	Covered	Uncovered
Defender	4	1
Attacker	0	2

"Utility for follower Ψ if attacks target 3 and it is covered (c) / uncovered (u)"

$u^c_\Psi(3)$ $u^u_\Psi(3)$

COMPACT REPRESENTATIONS OF SECURITY GAMES—EXTENSIVE FORM IS TOO BIG!

Defender commits to a mixed strategy (one of uncountably many, i.e., EFG tree will be infinite size)

$$\left.\begin{aligned} \Delta = (\delta_{12}, \delta_{13}, \delta_{14}, \delta_{23}, \delta_{24}, \delta_{34}) \\ \forall i, j \; 0 \leq \delta_{ij} \leq 1 \end{aligned}\right\} \text{In general, size } \binom{n}{m}$$

$$\sum_{i,j} \delta_{ij} = m$$

Attacker strategy is an efficient algorithm, which given any mixed strategy, Δ, computes target

$$\arg \max_{t \in \Gamma(\Delta)} U_{\Theta}(\Delta, t)$$

Where optimization is taken over the attack set Γ(Δ), the set of targets yielding max expected payoff for attacker given Δ

$$\Gamma(\Delta) = \{t : t \in \arg\max U_{\Psi}(\Delta, t)\}$$

COMPACT REPRESENTATIONS OF SECURITY GAMES

Key insight: the only information needed to represent the defender strategy is the probabilities a target is covered

$$\delta_{\Theta}^{1,2} + \delta_{\Theta}^{1,3} + \delta_{\Theta}^{1,4} = c_1$$
$$\delta_{\Theta}^{1,2} + \delta_{\Theta}^{2,3} + \delta_{\Theta}^{2,4} = c_2$$
$$\delta_{\Theta}^{1,3} + \delta_{\Theta}^{2,3} + \delta_{\Theta}^{3,4} = c_3$$
$$\delta_{\Theta}^{1,4} + \delta_{\Theta}^{2,4} + \delta_{\Theta}^{3,4} = c_4$$

In our 2 resources, 4 targets example: probability c_1 that target 1 is covered is sum of all pure strategies that cover 1

This gives us a coverage vector C

- Running example: C = [c_1, c_2, c_3, c_4]

ERASER (Efficient Randomized Allocation of SEcurity Resources) **takes security game & computes C that is SSE for defender**

ERASER FORMULATION

$$\begin{aligned}
\max \quad & d \\
a_t \in \quad & \{0,1\} \quad && \forall t \in T \\
\sum_{t \in T} a_t = \quad & 1 \\
c_t \in \quad & [0,1] \quad && \forall t \in T \\
\sum_{t \in T} c_t \leq \quad & m \\
d - U_\Theta(t,C) \leq \quad & (1-a_t)\cdot Z \quad && \forall t \in T \\
0 \leq k - U_\Psi(t,C) \leq \quad & (1-a_t)\cdot Z \quad && \forall t \in T
\end{aligned}$$

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

Attacker can assign mass to exactly one target

Defender applies valid (aka at most m) probability mass over targets

↓

(Theorem in paper states how to convert coverage vector to mixed strategy)

ERASER FORMULATION

$$
\begin{aligned}
\max \quad & d \\
a_t \in \quad & \{0,1\} \quad \forall t \in T \\
\sum_{t \in T} a_t = \quad & 1 \\
c_t \in \quad & [0,1] \quad \forall t \in T \\
\sum_{t \in T} c_t \leq \quad & m \\
d - U_\Theta(t,C) \leq \quad & (1 - a_t) \cdot Z \quad \forall t \in T \\
0 \leq k - U_\Psi(t,C) \leq \quad & (1 - a_t) \cdot Z \quad \forall t \in T
\end{aligned}
$$

Expected utility to leader given attack on t and coverage vector with coverage c_t

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

Determine the defender's expected payoff d, given the target attacked (a_t)

- **For unattacked targets (a_t=0), RHS is huge (i.e., Z)**
- **For attacked target (a_t=1), RHS is 0 → d = utility of defender given t attacked, and coverage vector C**

Objective: maximize d

ERASER FORMULATION

$$
\begin{aligned}
\max \quad & d \\
a_t \in \quad & \{0,1\} \quad && \forall t \in T \\
\sum_{t \in T} a_t = \quad & 1 \\
c_t \in \quad & [0,1] \quad && \forall t \in T \\
\sum_{t \in T} c_t \leq \quad & m \\
d - U_\Theta(t, C) \leq \quad & (1 - a_t) \cdot Z \quad && \forall t \in T \\
0 \leq k - U_\Psi(t, C) \leq \quad & (1 - a_t) \cdot Z \quad && \forall t \in T
\end{aligned}
$$

Two bottom sets of constraints imply that defender's coverage vector C is best response to attack vector A, & vice versa

→ Strong Stackelberg Equilibrium

"Big M" (or in this case "Big Z") style of constraints are a common way to encode if statements

ERASER: RUNNING EXAMPLE (2 RESOURCES, 4 TARGETS)

$$
\begin{aligned}
\max d \\
s.t. \\
\\
a_1 + a_2 + a_3 + a_4 = 1 \\
c_1 + c_2 + c_3 + c_4 \leq m \\
d - 4c_1 + (c_1 - 1) \leq (1 - a_1)Z \\
d - 4c_2 + (c_2 - 1) \leq (1 - a_2)Z \\
d - 4c_3 + (c_3 - 1) \leq (1 - a_3)Z \\
d - 4c_4 + (c_4 - 1) \leq (1 - a_4)Z \\
0 \leq k + c_1 - 1 \leq (1 - a_1)Z \\
0 \leq k + c_2 - 1 \leq (1 - a_2)Z \\
0 \leq k + 2c_3 - 2 \leq (1 - a_3)Z \\
0 \leq k + c_4 - 1 \leq (1 - a_4)Z \\
c_t \in [0, 1] \\
a_t \in \{0, 1\}
\end{aligned}
$$

NG

ERASER: RUNNING EXAMPLE (2 RESOURCES, 4 TARGETS)

```
Elapsed time = 0.01 sec. (0.26 ticks, tree = 0.01 MB, solutions = 3)

Root node processing (before b&c):
  Real time             =    0.01 sec. (0.26 ticks)
Parallel b&c, 4 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.01 sec. (0.26 ticks)

Solution status =  101 : MIP_optimal
Solution value  =  3.14285714286
Row 0:  Slack =   0.000000
Row 1:  Slack =   0.000000
Row 2:  Slack =  99.142857
Row 3:  Slack =  99.142857
Row 4:  Slack =   0.000000
Row 5:  Slack =  99.142857
Row 6:  Slack =   0.000000
Row 7:  Slack =   0.000000
Row 8:  Slack =   0.000000
Row 9:  Slack =   0.000000
Row 10:  Slack = 100.000000
Row 11:  Slack = 100.000000
Row 12:  Slack =   0.000000
Row 13:  Slack = 100.000000
Column 0:  Value =   3.142857
Column 1:  Value =  -0.000000
Column 2:  Value =  -0.000000
Column 3:  Value =   1.000000
Column 4:  Value =   0.000000
Column 5:  Value =   0.428571
Column 6:  Value =   0.428571
Column 7:  Value =   0.714286
Column 8:  Value =   0.428571
Column 9:  Value =   0.571429
Coverage vector: [0.428571428571, 0.428571428571, 0.714285714286, 0.428571428571]
Adversary attack vector: [-0.0, -0.0, 1.0, 0.0]
mb_pro_umd:mech ngupta$
```

$$c_1 = c_2 = c_4 = 3/7$$
$$c_3 = 5/7$$

ERASER – RUNNING EXAMPLE

Problem: we need mixture over pure strategies (i.e., placements of resources on targets), not just coverage vector

$$\delta_{12} + \delta_{13} + \delta_{14} = 3/7$$
$$\delta_{12} + \delta_{23} + \delta_{24} = 3/7$$
$$\delta_{13} + \delta_{23} + \delta_{34} = 5/7$$
$$\delta_{14} + \delta_{24} + \delta_{34} = 3/7$$
$$0 \leq \delta_{12} \leq 1$$
$$0 \leq \delta_{13} \leq 1$$
$$0 \leq \delta_{14} \leq 1$$
$$0 \leq \delta_{23} \leq 1$$
$$0 \leq \delta_{24} \leq 1$$
$$0 \leq \delta_{34} \leq 1$$

⇒

$$\delta_{12} = \delta_{14} = \delta_{24} = 2/21$$
$$\delta_{13} = \delta_{23} = \delta_{34} = 5/21$$

NG

ERASER-C(ONSTRAINED)

Can generalize to a setting where resources have a type drawn from some type space Ω

- Type ω in Ω determines feasible coverage schedules, i.e., subsets of targets coverable by that resource

Yields a very similar compact IP, similar solution of probability mass placed on each resource and schedule

$$
\begin{array}{rcl}
\max & d & \\
a_t \in & \{0,1\} & \forall t \in T \\
c_t \in & [0,1] & \forall t \in T \\
q_s \in & [0,1] & \forall s \in S \\
h_{s,\omega} \in & [0,1] & \forall s,\omega \in S \times \Omega \\
\sum_{t \in T} a_t = & 1 & \\
\sum_{\omega \in \Omega} h_{s,\omega} = & q_s & \forall s \in S \\
\sum_{s \in S} q_s M(s,t) = & c_t & \forall t \in T \\
\sum_{s \in S} h_{s,\omega} Ca(s,\omega) \leq & \mathcal{R}(\omega) & \forall \omega \in \Omega \\
h_{s,\omega} \leq & Ca(s,\omega) & \forall s,\omega \in S \times \Omega \\
d - U_\Theta(t,C) \leq & (1-a_t) \cdot Z & \forall t \in T \\
0 \leq k - U_\Psi(t,C) \leq & (1-a_t) \cdot Z & \forall t \in T
\end{array}
$$

HOW TO COMPUTE THE ACTUAL MIXED STRATEGY TO FOLLOW?

Kiekintveld paper proved feasible solutions (i.e., coverage vectors) to their MIPs corresponded to mixed strategies

Did not show how to compute them quickly ($\binom{n}{m}$ variables $\delta_{\omega,t}$)

First idea: for each target t*:

- Solve separate compact LP under the constraint that the attacker is incentivized to attack t*
- Pick LP with best defender utility
- Just like last lecture!

$$
\begin{aligned}
&\text{maximize } U_d(t^*, \mathbf{c}) \\
&\text{subject to} \\
&\forall \omega \in \Omega, \forall t \in A(\omega) : 0 \leq c_{\omega,t} \leq 1 \\
&\forall t \in T : c_t = \sum_{\omega \in \Omega : t \in A(\omega)} c_{\omega,t} \leq 1 \\
&\forall \omega \in \Omega : \sum_{t \in A(\omega)} c_{\omega,t} \leq 1 \\
&\forall t \in T : U_a(t, \mathbf{c}) \leq U_a(t^*, \mathbf{c})
\end{aligned}
$$

Problem: this still gives marginal probabilities over targets

We need probability mixture over pure strategies!

A TOOL: BIRKHOFF-VON NEUMANN THEOREM

Every doubly stochastic n x n matrix can be represented as a convex combination of n x n permutation matrices

.1	.4	.5
.3	.5	.2
.6	.1	.3

= .1

1	0	0
0	0	1
0	1	0

+.1

0	1	0
0	0	1
1	0	0

+.5

0	0	1
0	1	0
1	0	0

+.3

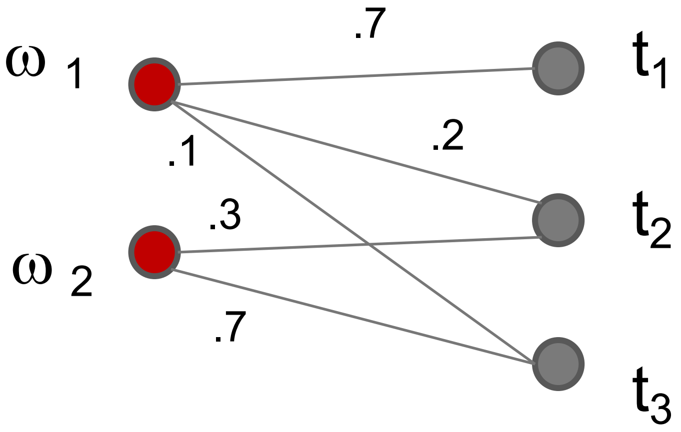
0	1	0
1	0	0
0	0	1

Decomposition can be found in polynomial time $O(n^{4.5})$, and the size is $O(n^2)$ [Dulmage and Halperin, 1955]

Can be extended to rectangular doubly substochastic matrices

SCHEDULES OF SIZE 1 USING BVN

"Schedule of size 1" → resource is assigned to exactly one target



	t_1	t_2	t_3
ω_1	.7	.2	.1
ω_2	0	.3	.7

.1

0	0	1
0	1	0

.2

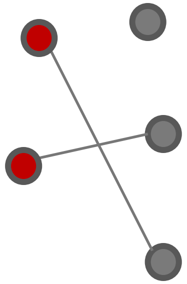
0	1	0
0	0	1

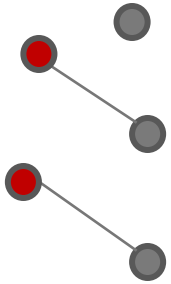
.2

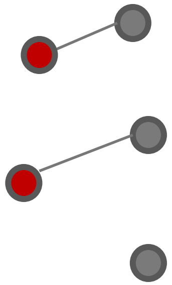
1	0	0
0	1	0

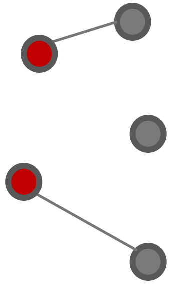
.5

1	0	0
0	0	1

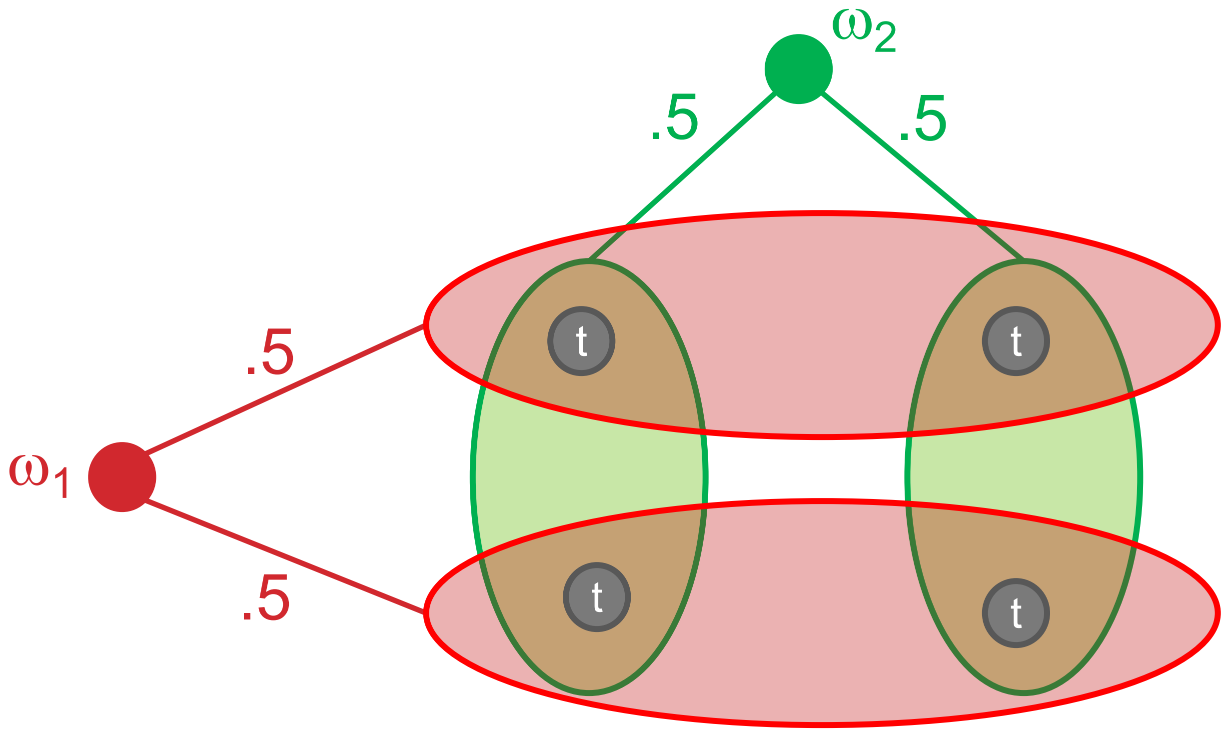








COUNTER-EXAMPLE TO THE COMPACT LP



2 resources ω_1 & ω_2, schedules of size 2

LP suggests: we can cover every target with probability 1 ?????

… but in fact we can cover at most 3 targets at a time → for general schedule sizes, it is not always possible to find feasible mixture

VC

ALGORITHMS & COMPLEXITY

[Korzhyk, Conitzer, Parr, "Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games]

		Homogeneous Resources	Heterogeneous resources
Schedules	Size 1	P	P (BvN theorem)
	Size ≤2, bipartite	P (BvN theorem)	NP-hard (SAT)
	Size ≤2	P (constraint generation)	NP-hard
	Size ≥3	NP-hard (3-COVER)	NP-hard