Security Games

Tuesday, September 28, 2021 1:42 PM

Defender (leader)

n targets to defend $= \{1 \dots n\} = [n]$

$\mathcal{E}$ = set of possible defence strategies

$\mathcal{E} \subseteq \{0,1\}^n \qquad e \in \mathcal{E}$

$e_i = 1$ if target i is defended

$= 0$ o.w.

r_i = reward if i is defended while attacked. ≥ 0

c_i = cost o.w. ≤ 0

Attacker (Follower)

Can only attack "one" target $i \in [n]$.

s_i = reward if i was not defended when he/she attacked it ≥ 0

ξ_i = cost o.w. ≤ 0

→ Pure play $(e, \underline{i})$, $e \in \mathcal{E}$, $i \in [n]$

$\underline{r_i e_i + c_i(1-e_i)}$ | $\underline{s_i(1-e_i) + \xi_i e_i}$

★ Mixed (randomized) play

$p \in \Delta(\mathcal{E}) \qquad y \in \Delta([n])$

$$\text{Defender's payoff } (p, y) = \sum_{e \in \mathcal{E}} \sum_{i \in [n]} p_e y_i \left(r_i e_i + c_i (1-e_i) \right)$$

$$= \sum_{i \in [n]} y_i \Big[r_i \underbrace{\Big(\sum_{e \in \mathcal{E}} p_e e_i \Big)}_{x_i} + c_i \underbrace{\Big(1 - \sum_{e \in \mathcal{E}} p_e e_i \Big)}_{(1-x_i)} \Big]$$

marginal probability with which target i is defended.

$$x \in \mathcal{D} = \Big\{ x \in [0,1]^n \;\Big|\; x = \sum_{e \in \mathcal{E}} p_e e \Big\}$$

$$= \sum_{i\in[n]} y_i \left(r_i x_i + c_i(1-x_i)\right)$$

Q: Suppose, we fix y for the attacker
Then what x is Best for the defender

$$\arg\max_{x\in P} \underbrace{\sum_{i\in[n]} x_i \underbrace{\left(y_i(r_i-c_i)\right)}_{\geq 0}}_{w_i} + \sum_{i\in[n]} y_i c_i$$

independent of x_i's.

equiv!

$$\max_{e\in \mathcal{E}} \sum_{i\in[n]} e_i . w_i = \boxed{\max_{e\in \mathcal{E}} \langle e, w\rangle}$$

combinatorial set

Combinatorial Problem. → ☆.

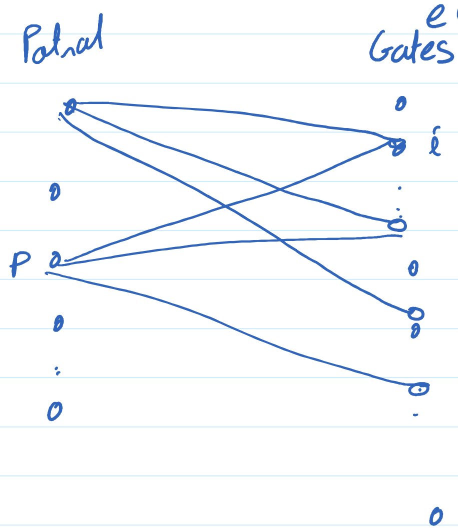
☆ ORD : defend gates of ORD
n-gates k- police patrol.
$n >> k$

$$\mathcal{E} = \left\{ e \in \{0,1\}^n \;\middle|\; \sum_{i\in[n]} e_i \leq k \right\} \leftarrow$$

$$x \in P = \left\{ x \in [0,1]^n \;\middle|\; \sum_{i\in[n]} x_i \leq k \right\}$$

"Constraints"

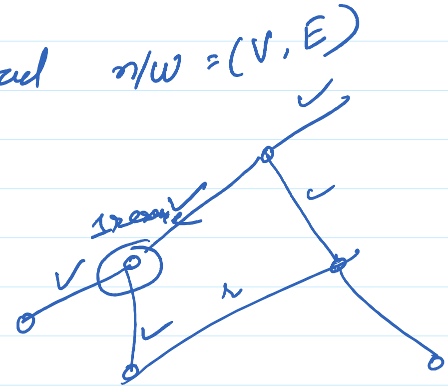
$$x_i = \sum_{e\in\mathcal{E}} p_e . e_i \Rightarrow \sum_{i\in[n]} x_i = \sum_{e\in\mathcal{E}} p_e \underbrace{\left(\sum_{i\in[n]} e_i\right)}_{\leq k} \leq k \cdot \underbrace{\sum_e p_e}_{=1} = k$$



$$\mathcal{E} = \{\text{matchings of size} = k\}$$

☆ Problem is the max-weight-matching

★ Protecting road n/w $= (V, E)$



$\mathcal{E} = \{$subset of vertices of size $\leq k\}$

★ Problem is max-weight vertex cover

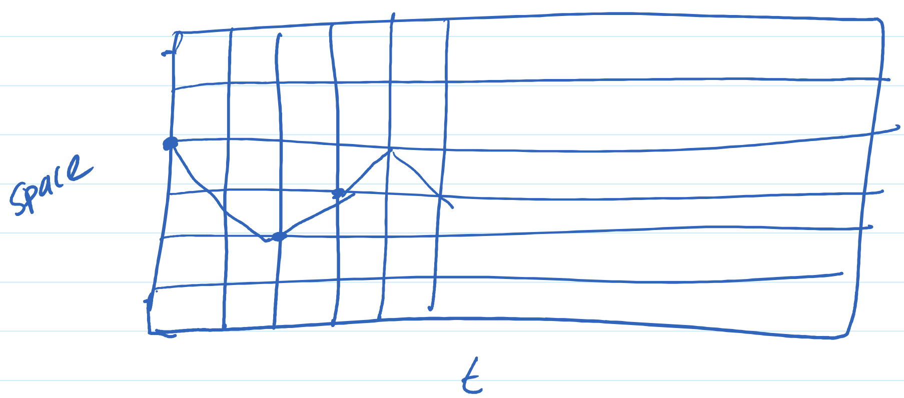
→ Air marshal's problem:

↳ can defend flight A → flight B if destination of A = source of B

$S_1, \ldots, S_m$ are all possible feasible subsets of flight.

K-A.M. ⇒ we may pick k subsets that maximizes the coverage.

★ Defend Ports / Forests for poaching



Goal: defend as many targets as many times.

Theorem: If the DBR(★) problem can be solved in polynomial-time then SE can be computed in poly-time.

Pf: CS '06 : $\forall K \in [n]$

LP_K max: $r_K x_K + c_K (1 - x_K)$

Dual var   s.t.

$y_i \leftarrow \quad s_K(1-x_K) + \xi_K x_K \geq s_i(1-x_i) + \xi_i x_i \quad \forall i \neq K, \; i \in [n]$

$w_i \leftarrow \quad \Rightarrow x_i = \sum_{e \in \mathcal{E}} p_e \cdot e_i \quad \forall i \in [n]$

$\mathcal{E}$ ← complex & exponential in size

$v \leftarrow \quad \sum_{e \in \mathcal{E}} p_e = 1$

$p_e \geq 0 \quad \forall e \in \mathcal{E}.$

$\downarrow$ Dual

DLP_K:

$$\min: \sum_{i \neq K} (s_K - s_i) y_i - v$$

s.t.

$$w_K = (\xi_K - s_K) - \Big(\sum_{i \neq K} y_i\Big)(s_K - \xi_K) \leftarrow$$

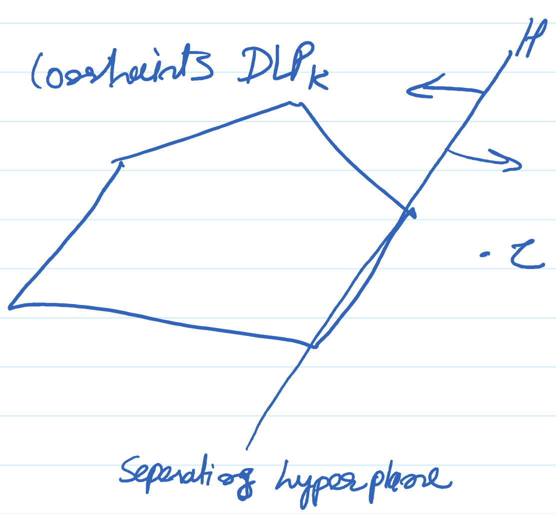
$$\forall i \neq K \quad w_i = s_i - \xi_i \leftarrow$$

} Easy to check.

$$\forall e \in \mathcal{E}, \quad v \geq \langle e, w \rangle \leftarrow \text{ exponentially many!}$$

$$y \geq 0$$

solve DLP_K using Ellipsoid method if $\exists$ separation oracle



$\downarrow$ Algorithm

$z = (v^*, w^*, y^*)$

$\downarrow$

Check if $\forall e \in \mathcal{E}$

$$v^* \geq \langle e, w^* \rangle$$

$$\equiv \quad v^* \geq \boxed{\max_{e \in \mathcal{E}} \langle e, w^* \rangle} \; \bigstar$$

DBR.

if $v^* \geq \max$ then $\tau \in DLP_K$ polytope

o.w. $e^* = \arg\max_{e \in \mathcal{E}} \langle e, w^* \rangle$

$v^* < \langle e^*, w^* \rangle$ $\Leftarrow$ seperating hyperplane..