When Security Games Go Green

Suraj Nair

Outline

- Green Security Game
- Planning algorithms
- Planning and learning
- Results

Green Security Domains: Protecting Fish and Wildlife

Features

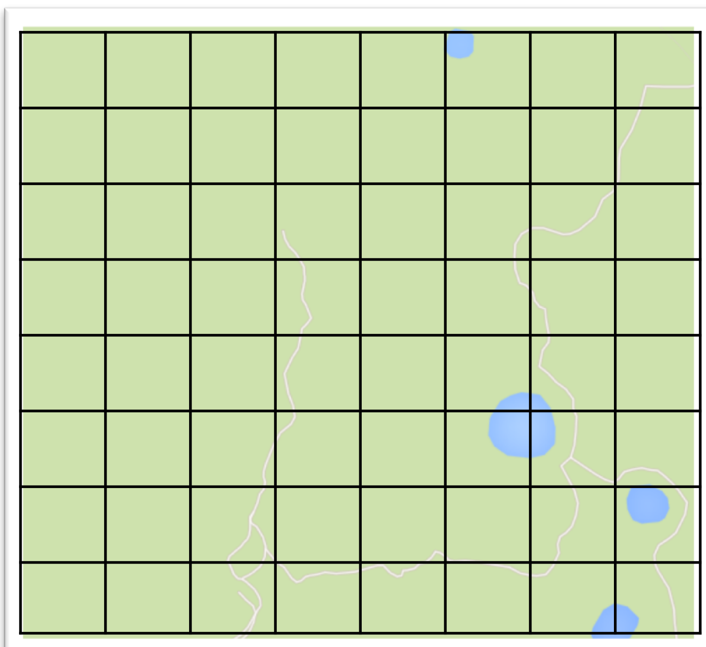
Green security games

- *Generalized Stackelberg assumption*
- *Repeated and frequent attacks*
  - Significant amounts data
- *Attacker bounded rationality*
  - Limited surveillance/planning

Green Security Game Model

- T round game, K defenders, N targets where $N \geq K$
- Coverage vector $c = \langle c_i \rangle$ where
  - c_i denotes probability that target i is covered
  - c^t denotes the defender strategy profile for round t

Jan	Feb	Mar	Apr	May



0.3	0.4	0.1	0.2	0.5	0	0.1	0.7
0.2	0.2	0.3	0.3	0.4	0.5	0.6	0.1
0.6	0.7	0.1	0.2	0.3	0.2	0	0.9
0.4	0.5	0.6	0.3	0.1	0.2	0.3	0
0.1	0.2	0.3	0.4	0.5	0.3	0.4	0.1
0.3	0.4	0.5	0.5	0.5	0.1	0.1	0.1
0.2	0.2	0.2	0.3	0.3	0.3	0.4	0.5
0.1	0.1	0.6	0.3	0.4	0.2	0.4	0.4

Green Security Game Model

- L attackers who respond to convex combination of defender strategy in recent rounds
  - η^t denotes the strategy of attacker for round t

Jan	Feb	Mar	Apr	May
c^1	c^2	c^3=?		

$\boxed{\eta^3 = 0.3c^1 + 0.7c^2}$

- Payoff values for target i $P_i^a, R_i^a, P_i^d, R_i^d$
  - Where P stands for Penalty, R for reward
  - a for attacker, d for defender
- Expected utility for defender d if attacker targets i
  - $U_i^d(c) = c_i R_i^d + (1 - c_i) P_i^d$

Green Security Game Model

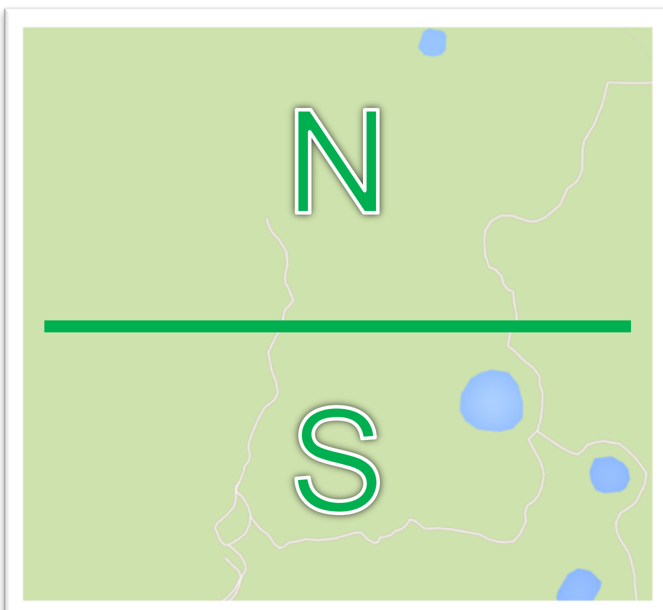
- Attacker chooses target with bounded rationality
  - Following the SUQR model
    - Choose <u>more promising targets</u> with higher probability
  - Probability that an attacker attacks target i is
    - $q_i(\omega, \eta) = \frac{e^{\omega_1 \eta_i + \omega_2 R_i^a + \omega_3 P_i^a}}{\sum_j e^{\omega_1 \eta_i + \omega_2 R_i^a + \omega_3 P_i^a}}$
  - Create a defender strategy profile $\lfloor c \rfloor = \langle c^1, \ldots, c^T \rangle$
  - Expected utility of defender in round t
    - $E^t(\lfloor c \rfloor) = \sum_l \sum_i q_i\left(\omega^l, \eta^t\right) U_i^d(c^t)$

Outline

- Green Security Game Model
- ***Planning algorithms***
- Planning and Learning
- Results

Planning

- Exploit attackers' delayed observation ($\eta^t = c^{t-1}$)
- A simple example:
  - Patrol Plan A: always uniformly random
  - Patrol Plan B: change her strategy deliberately, detect more snares overall

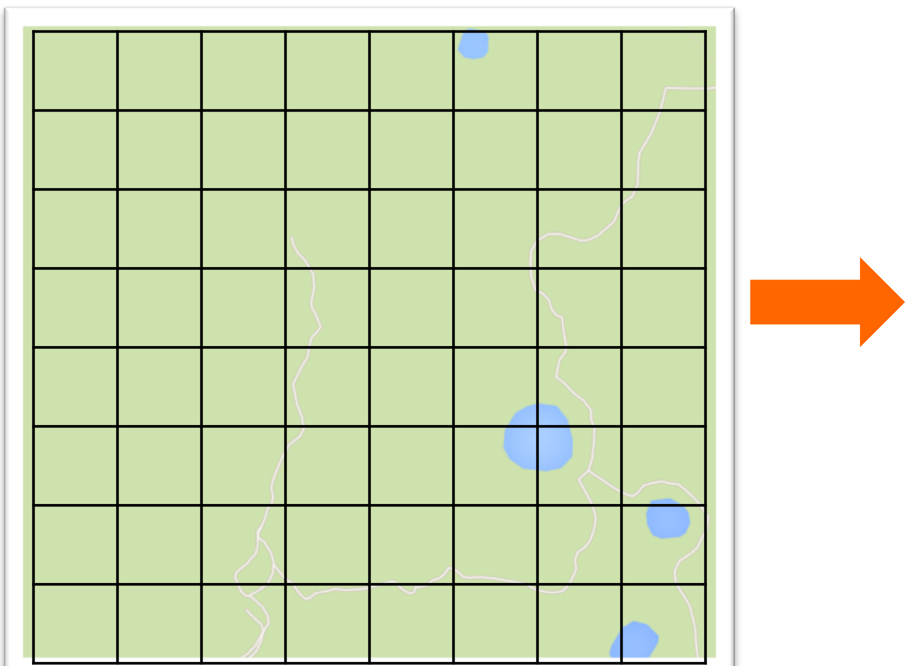


	Jan	Feb
N	80%	20%
S	20%	80%

Planning

- Solve directly X
  - Optimize over all rounds → computationally expensive

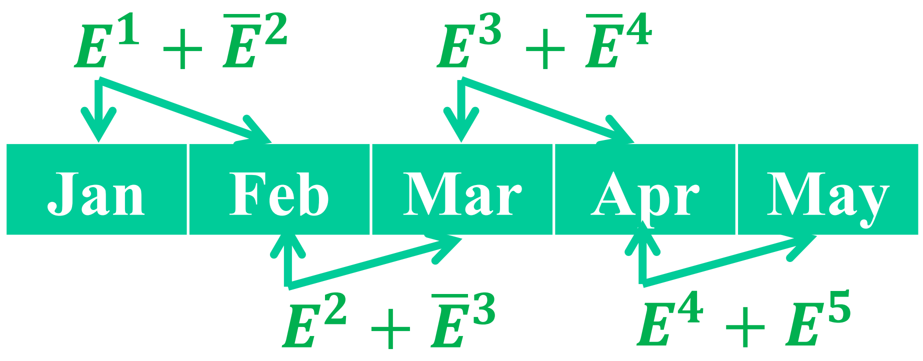
Jan	Feb	Mar	Apr	May



?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?
?	?	?	?	?	?	?	?

PlanAhead-M

- PlanAhead-M
  - Look ahead M steps: find an optimal strategy for current round as if it is the M[th] last round of the game
  - Sliding window of size M. Example with M=2



- Add discount factor γ to compensate the over-estimation

PlanAhead-M

Algorithm 1 Plan Ahead(ω, M)

Output: a defender strategy profile $[c]$

1: **for** t=1 to T **do**

2: $\quad c^t = \text{f-PlanAhead}(c^{t-1}, \omega, \min\{T-t+1, M\})$

- Mathematical program

$$\max_{c^t, c^{t+1}, \ldots c^{t+m-1}} \sum_{\tau=0}^{m-1} E^{t+\tau} \quad (2)$$

$$s.t \quad E^\tau = \sum_l \sum_i q_i(\omega^l, \eta^\tau) U_i^d(c^\tau), \tau = t, .., t+m-1 \quad (3)$$

$$\eta^\tau = c^{\tau-1}, \tau = t, .., t+m-1 \quad (4)$$

$$\sum_i c_i^\tau \le K, \tau = t, .., t+m-1 \quad (5)$$

FixedSequence-M

- Require the defender to execute the sequence of length M repeatedly

- Example with M=2: find best strategy A and B

Jan	Feb	Mar	Apr	May
A	B	A	B	A

- Theoretical guarantee: $\left(1-\frac{1}{M}\right)$ approximation of the optimal strategy profile

FixedSequence-M

Algorithm 2 Fixed Sequence

Output: defender strategy profile $[c]$

1: $(a^1, ..., a^M) = \text{f-FixedSequence}(\omega, M)$.
2: **for** t=1 to T **do**
3: $\quad c^t = a^{(t \bmod M)+1}$

$$\max_{a^1,...,a^M} \sum_{t=1}^{M} E^t \tag{7}$$

$$s.t \quad E^t = \sum_l \sum_i q_i(\omega^l, \eta^t) U_i^d(a^t), t = 1, ..., M \tag{8}$$

$$\eta^1 = a^M \tag{9}$$

$$\eta^t = a^{t-1}, t = 2, ..., M \tag{10}$$

$$\sum_i a_i^t \leq K, t = 1, .., M \tag{11}$$

Outline

- Green Security Game Model
- Planning algorithms
- ***Planning and Learning***
- Results

Planning and Learning

- Learn parameters in attackers' bounded rationality model from attack data

- Previous work
  - Apply Maximum Likelihood Estimation (MLE)
  - May lead to highly biased results

- Proposed learning algorithm
  - Calculate posterior distribution for each data point

Planning and Learning

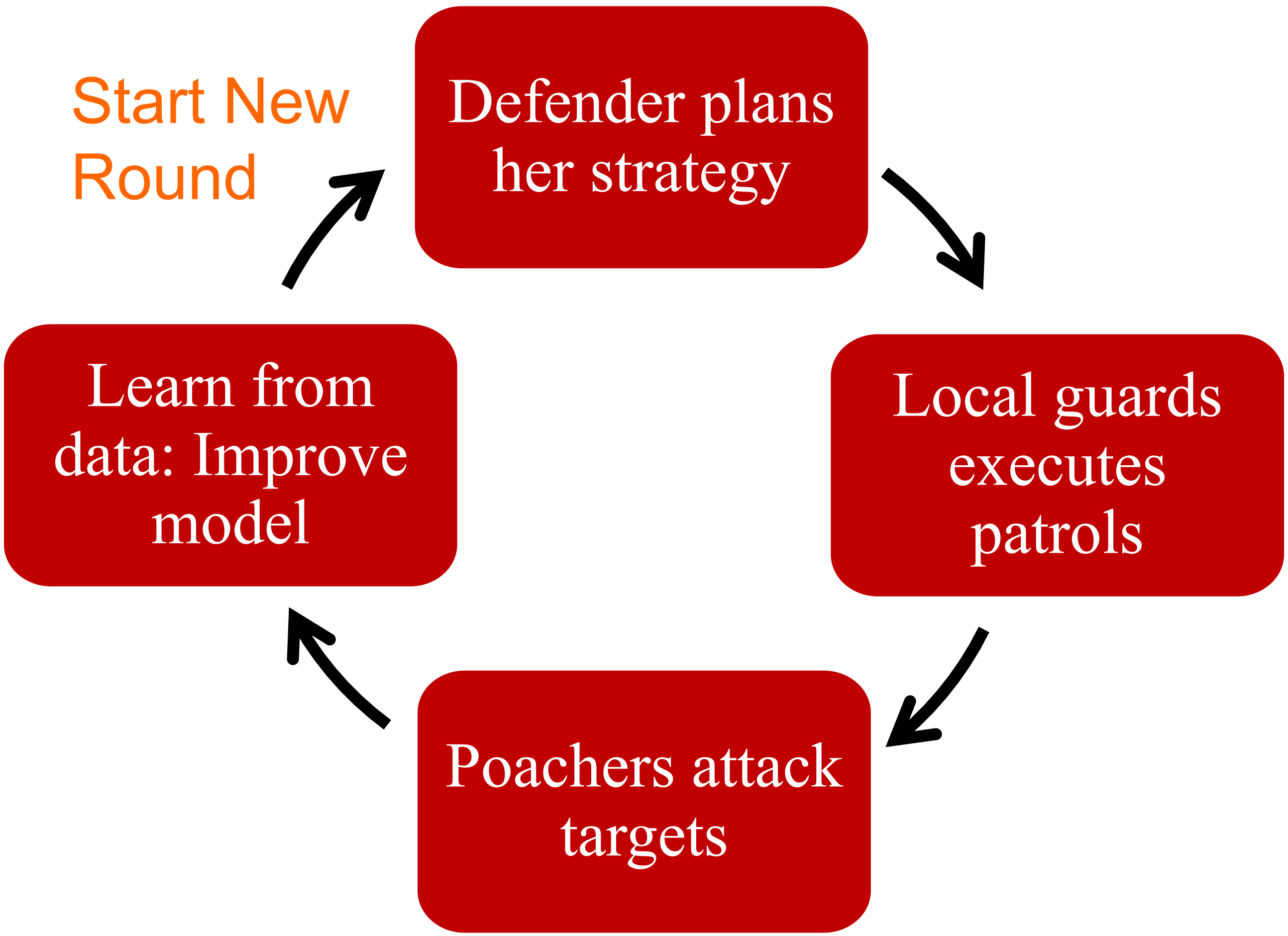
Algorithm 3 Learn-BU (η, χ,$\{\hat{\omega}\}$, p)

Output: $\bar{p}$ – a probability distribution over $\{\hat{\omega}\} = \{\hat{\omega}^1, ..., \hat{\omega}^S\}$.

1: **for** i=1 to N **do**
2:     **for** s=1 to S **do**
3:         $\bar{p}_i(s) = \frac{p(s)q_i(\hat{\omega}^s,\eta)}{\sum_r p(r)q_i(\hat{\omega}^r,\eta)}$
4: **for** s=1 to S **do**
5:     $\bar{p}(s) = \frac{\sum_i \chi_i \bar{p}_i(s)}{\sum_i \chi_i}$

χ_i - number of attacks on target i
discrete set $\{\widehat{\omega}\}$ - $\{\widehat{\omega}^1, \dots, \widehat{\omega}^S\}$
prior p - $\langle p_1, \dots, p_S \rangle$

General Framework of Green Security Game

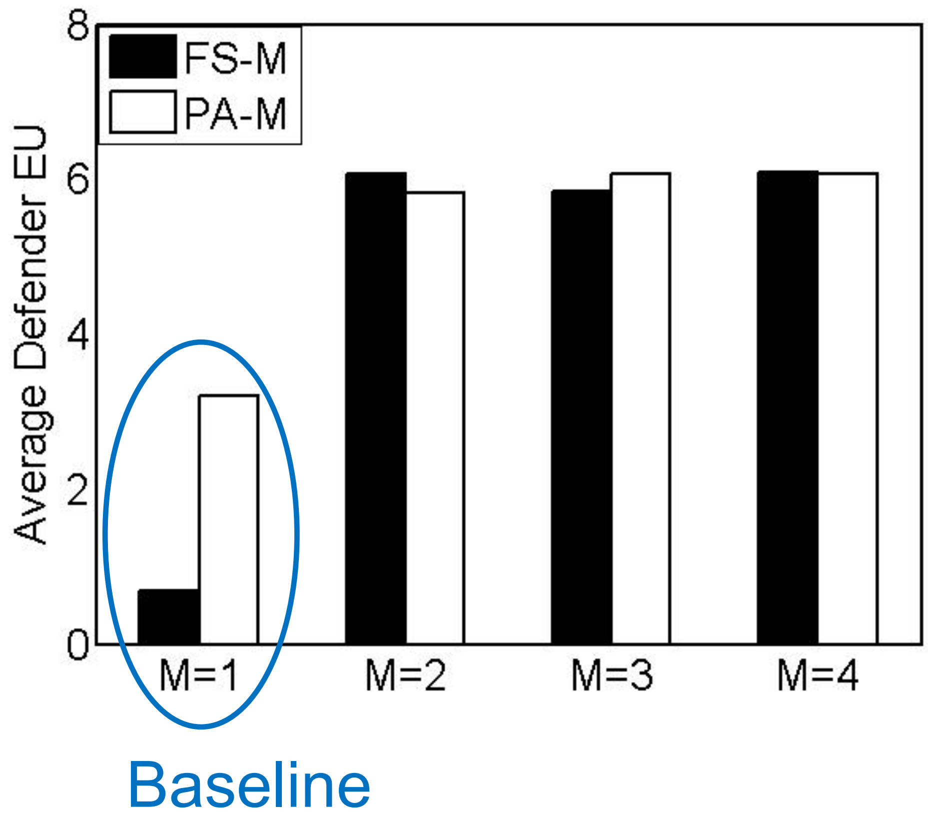


Outline

- Green Security Game Model
- Planning algorithms
- Planning and Learning
- ***Results***

Experimental Results
Planning

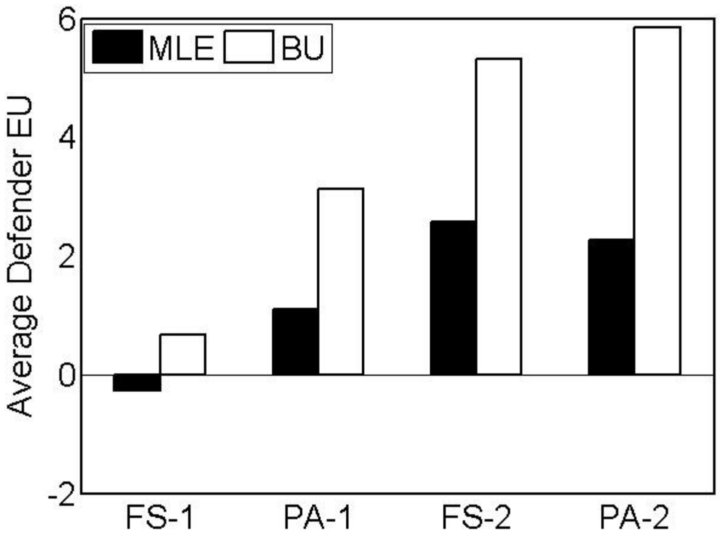
- Baseline: FS-1 (Stackelberg), PA-1 (Myopic)
- Attacker respond to last round strategy, 10 Targets, 4 Patrollers



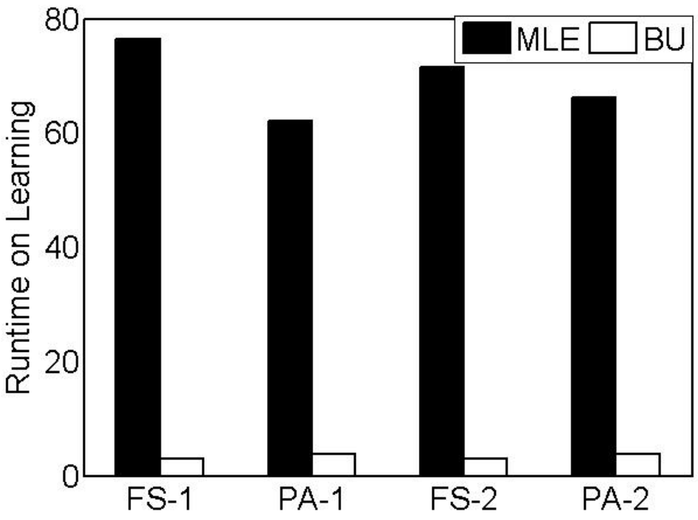
Experimental Results
Planning and Learning

- Baseline: Maximum Likelihood Estimation (MLE)

Solution Quality



Runtime



Thank you!