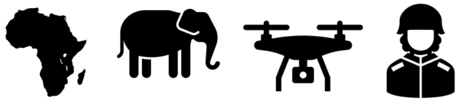
Evolutionary Approach to Security Games with Signaling



Adam Żychowski[1]**,** Jacek Mańdziuk[1], Elizabeth Bondi[2], Aravind Venugopal[3], Milind Tambe[2], Balaraman Ravindran[3,4]

[1]Faculty of Mathematics and Information Science, Warsaw University of Technology

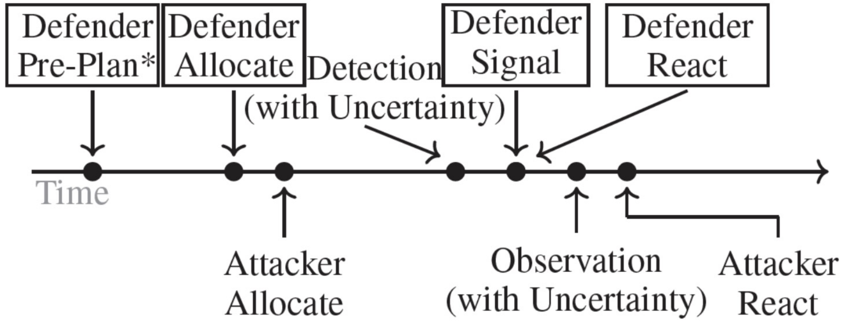
[2]Center for Research on Computation and Society, Harvard University

[3]Robert Bosch Centre for Data Science and AI, IIT Madras

[4]Department of Computer Science and Engineering, IIT Madras

Problem definition

- Inspiration: prevent poaching in Africa
- 2 players: **Defender** and **Attacker**
- Defender's units: patrollers, drones
- Drone can send one of the following signals:
  - weak – sending information to patrollers about attack detection
  - strong – sending information about attack and lauch sound/light signals to deter Attacker
- Games on graph – each vertex is target with set of payoffs
- Defender's strategy: assigning patrollers and drones to targets, signaling strategy
- Attacker's strategy: choose target to attack, signaling reaction
- Players' uncertainties considered



Problem definition – Stackelberg equilibrium

- Defender commits strategy first
- Attacker, knowing the Defender's strategy, chooses his/her strategy
- Defender always commits to a mixed strategy

- **Stackelberg equilibrium**: pair of players' strategies, for which strategy change by any of players leads to his/her result deterioration.

$$(\pi_D^*, R(\pi_D^*)) \in \Pi_D \times \Pi_A$$

$$\pi_D^* = \mathrm{argmax}_{\pi_D \in \Pi_D} U_D(\pi_D, R(\pi_D))$$

$$R(\pi_D) = \mathrm{argmax}_{\pi_A \in \Pi_A} U_A(\pi_D, \pi_A)$$

$G \in \{D, A\}$ – *players (Defender, Attacker)*
Π_G – a set of player's G all mixed strategies
U_G – payoff of player G

Evolutionary algorithm (EASGS) - solutions encoding

$$CH_j = \{(e_1^j, q_1^j), \ldots, (e_i^j, q_i^j), \ldots, (e_{d_j}^j, q_{d_j}^j), \boldsymbol{\Psi_j^\theta}, \boldsymbol{\Phi_j^\theta}\}$$

$e_i^j \;=\; (V_p, V_s, V_r)$ - pure strategy

- V_p - a set of vertices with assigned patrollers
- V_s - a set of vertices with assigned drones
- V_r - reallocation plan, a set of vertices (connected with V_p), to which each patroller moves if no adversaries are observed

q_i^j - the probability of playing strategy e_i^j

$\theta \in \{\bar{s},\, s^+,\, s^-\}$ - drones allocation states:

- $\bar{s}$ - no patroller is in the drone's neighbourhood
- s^+- a patroller is planned to visit drone's vertex in the reaction stage
- s^- - no patroller will visit drone's vertex in the reaction stage but there is at least one patroller in neighbourhood who can respond

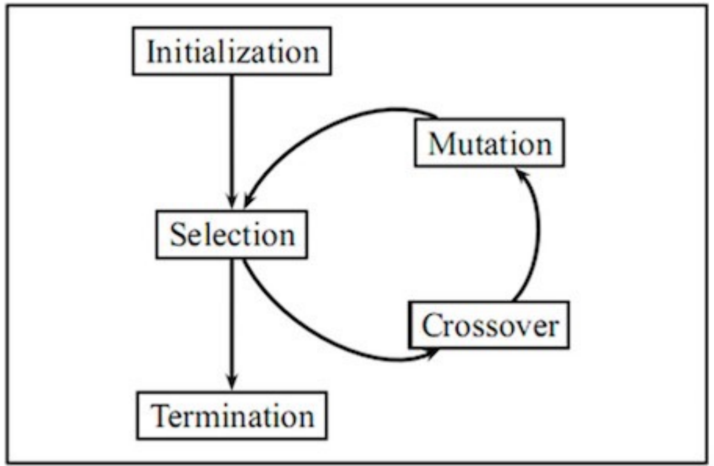
$\boldsymbol{\Psi_j^\theta} \;=\; [\Psi_{j,1}^\theta, \Psi_{j,2}^\theta, \ldots, \Psi_{j,\mathcal{N}}^\theta]$ - signaling strategy in case of attack detection

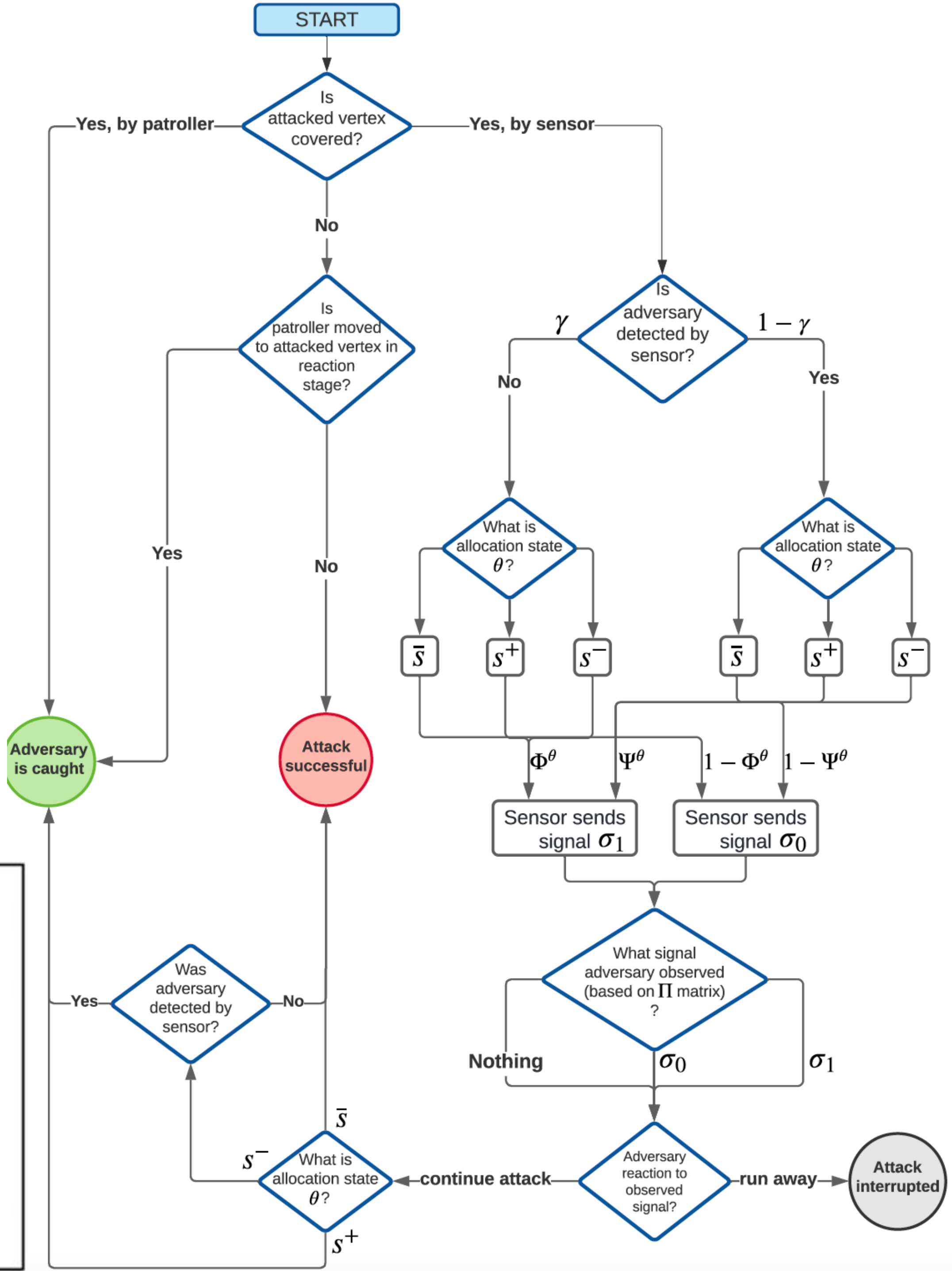
$\boldsymbol{\Phi_j^\theta} \;=\; [\Phi_{j,1}^\theta, \Phi_{j,2}^\theta, \ldots, \Phi_{j,\mathcal{N}}^\theta]$ - signaling strategy in case of no attack detection

EASGS operators

- 3 **mutation** types:
  - random allocation/reallocation modification
  - random singaling probability modification
  - random pure strategy improvement

- **Crossover**: combining pure strategies with halved probabilities, averaging signaling probabilities
- **Evaluation** based on game rules (including detection and observational uncertainties)

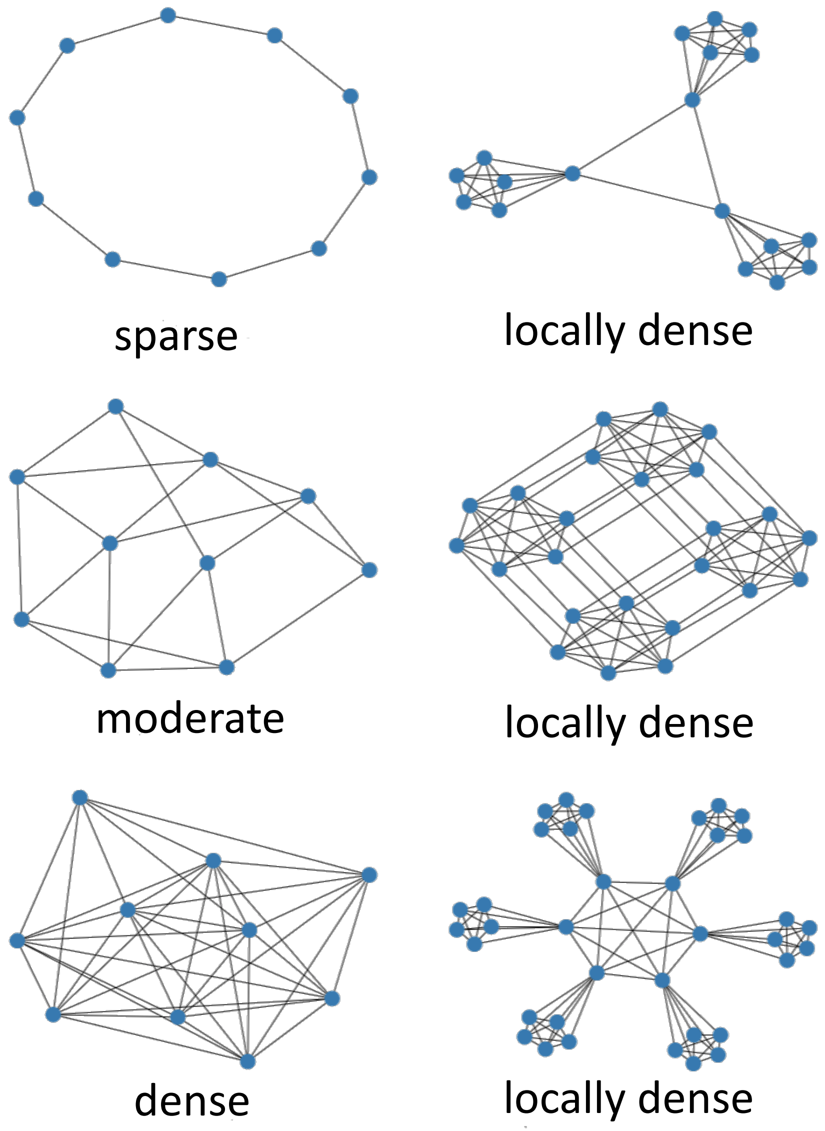




Benchmark games

- **342 games** with different graph topologies:
  - sparse (avg deg = 2) – 50 games
  - moderate (avg deg = $n/2$) – 50 games
  - dense (avg deg = n-2) – 50 games
  - locally dense (connected cliques) – 192 games

- number of vertices: $n \in [10, 100]$
- number of patrollers: $k_s = \sqrt{\frac{n}{2}}$
- number of drones: $k_d = \frac{2}{3}n - k_s$

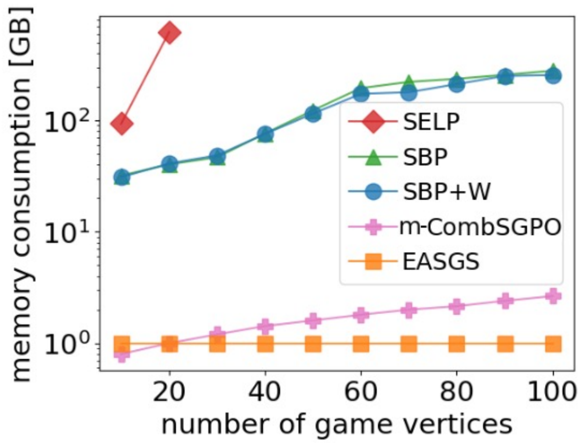


Results

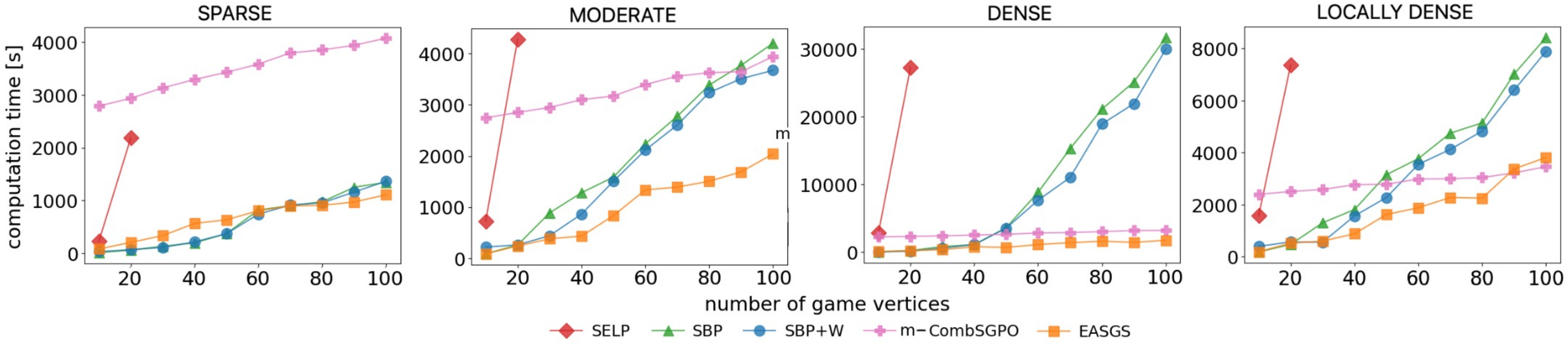
EASGS obtained the best result for 200 out of 342 games

	SBP	SBP+W	m-CombSGPO	EASGS
sparse	-86.68 (84%)	**-86.01 (92%)**	-419.86 (0%)	-91.32 (6%)
moderate	-75.01 (2%)	-72.75 (36%)	-255.73 (0%)	**-69.92 (62%)**
dense	-58.72 (2%)	-57.98 (34%)	-149.14 (0%)	**-51.47 (64%)**
locally-dense	-60.68 (4%)	-57.80 (26%)	-340.65 (0%)	**-54.36 (70%)**

Averaged payoffs for all games



Memory consumption



Time scalability

Thank you