4th Polish Conference on Artificial Intelligence

Optimized mutation operator in evolutionary approach to Stackelberg Security Games

Adam Żychowski, Jacek Mańdziuk

Warsaw University of Technology, Faculty of Mathematics and Information Science

24-26 April 2023

Abstract

We introduce several **mutation modifications in Evolutionary Algorithm for finding Strong Stackelberg Equilibrium in sequential Security Games**. The mutation operator used in the state-of-the-art evolutionary method is extended with several greedy optimization techniques. Proposed mutation operators are comprehensively tested on three types of games with different characteristics (totally over **300 test games**). The experimental results show that application of some of the **proposed mutations yields Defender's strategies with higher payoffs**. A trade-off between the results quality and the computation time is also discussed.

Federal Air Marshal Service

US Coast Guard in Boston Harbor

Los Angeles Airport

Poaching in Uganda

Tickets control in Los Angeles

Stackelberg Security Games (SSGs)

- Two asymmetrical players: **Defender and Attacker**
- Each game is composed of m time steps.
- Each player chooses an action to be performed in each time step.
- A player's *pure strategy* σ_P ($P \in \{D, A\}$) is a sequence of their actions in consecutive time steps: $\sigma_P = (a_1, a_2, \ldots, a_m)$.
- Many real-life applications: e.g. cybersecurity, scheduling canine patrols, protecting Boston Harbor, preventing poaching.

D

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

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

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

$$\pi_D^* = \text{argmax}_{\pi_D \in \Pi_D} U_D(\pi_D, R(\pi_D))$$
$$R(\pi_D) = \text{argmax}_{\pi_A \in \Pi_A} U_A(\pi_D, \pi_A)$$

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

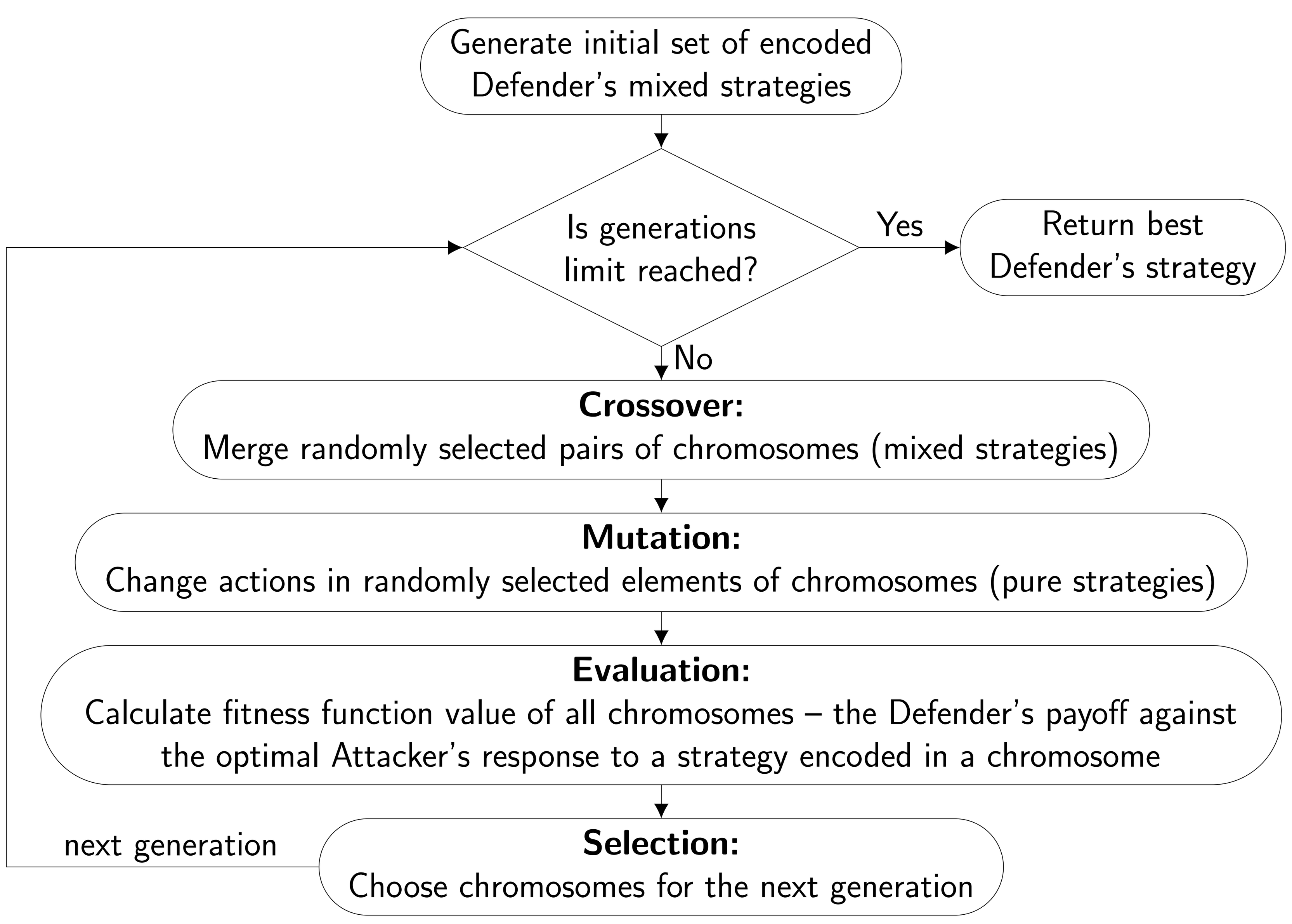

Figure: An overview of the EASG method.

Evolutionary Algorithm for Stackelberg Games (EASG)

EASG [1] aims to **optimize the Defender's payoff** by evolving a population of Defender's mixed strategies. Initially, EASG creates a population of pure Defender's strategies selected at random. The population evolves over successive generations until the stopping criterion is met. Four operations are applied in each generation: crossover, mutation, evaluation, and selection.

Defender's strategy encoding:

$$CH_q = \{(\sigma_1, p_1), \ldots, (\sigma_l, p_l)\}, \quad \sum_{i=1}^{l} p_l = 1$$

Mutation operator randomly selects a pure strategy encoded in the chromosome and modifies it, starting from a randomly selected time step. New actions are drawn from the set of all feasible actions in a given game state.

Mutation enhancements

- **EASG$_n$** - EASG algorithm with repeated mutation.
- **MANPS$_1$, MANPS$_n$** - *mutation adds new pure strategy* - a uniformly selected pure strategy is added with a uniformly sampled probability.
- **MCP$_1$, MCP$_n$** - *mutation changes probability* - a probability of randomly selected pure strategy is uniformly changed.
- **MSP$_1$, MSP$_n$** - *mutation switches probability* - probabilities of two randomly chosen pure strategies are switched.
- **MDPS$_1$, MDPS$_n$** - *mutation deletes pure strategy* -a randomly chosen pure strategy is removed.
- **MCWPS** - *mutation changes the weakest pure strategy* - mutation is applied only to a pure strategy with the lowest payoff.
- **MDWPS** - *mutation deletes the weakest pure strategy* - pure strategy with the lowest payoff is deleted.

Results

Table: The average and standard deviation values of the Defender's payoff and the computation time for various mutation operators. The best results are **bolded**. Results that are better than the baseline version of the algorithm (EASG) are underlined. In cases where the difference between the baseline version (EASG) and a given variation is statistically significant (according to the Wilcoxon test with p-value < 0.05), the result is highlighted with a gray background.

	Defender's payoff			Computation time [s]		
	WHG	SEG	FIG	WHG	SEG	FIG
EASG	**0.017**	0.108	0.031	152	2534	328
EASG$_n$	**0.017**	<u>0.135</u>	**<u>0.037</u>**	1206	21913	3051
MANPS$_1$	0.014	0.059	0.031	156	2548	313
MANPS$_n$	0.016	**<u>0.139</u>**	<u>0.036</u>	1366	21892	2988
MCP$_1$	0.015	0.074	0.030	148	2422	336
MCP$_n$	0.016	<u>0.131</u>	**<u>0.037</u>**	1285	22651	3008
MSP$_1$	0.013	0.099	0.024	156	2583	316
MSP$_n$	0.016	0.108	**<u>0.037</u>**	1332	21447	2931
MDPS$_1$	0.013	0.052	0.029	147	2620	313
MDPS$_n$	0.013	0.053	0.026	1283	22026	2900
MCWPS	0.013	0.046	0.030	148	2612	321
MDWPS	0.008	0.058	0.018	139	2361	299

Conclusions

- **Repetition of mutation operation leads to improvement of SSGs outcomes**, though at the expense of significant increase in computation time.
- The proposed modifications offer a **viable alternative to the base EASG formulation** for cases when computational cost is less important.

References

[1] Żychowski A., Mańdziuk J. Evolution of Strategies in Sequential Security Games. In Proceedings of the 20th AAMAS conference, pages 1434-1442. 2021.