A Generic Metaheuristic Approach to Sequential Security Games

Extended Abstract

Adam Żychowski
Warsaw University of Technology
Warsaw, Poland
a.zychowski@mini.pw.edu.pl

Jacek Mańdziuk
Warsaw University of Technology
Warsaw, Poland
j.mandziuk@mini.pw.edu.pl

ABSTRACT

The paper introduces a generic approach to solving Sequential Security Games (SGs) which utilizes Evolutionary Algorithms (EAs). Formulation of the method (named *EASG*) is general and largely game-independent, which allows for its application to a wide range of SGs with just little adjustments addressing game specificity. Experiments performed on 3 different types of games (with 300 instances in total) demonstrate robustness and stability of EASG, manifested by repeatable achieving optimal or near-optimal solutions in the vast majority of the cases. The main advantage of EASG is time efficiency. The method scales better than state-of-the-art approaches and can be applied to sequential SGs with bigger numbers of steps compared to the existing methods. Due to *anytime* characteristics, EASG is very well suited for time-critical applications.

KEYWORDS

Evolutionary algorithm; Security Games; Stackelberg equilibrium

ACM Reference Format:
Adam Żychowski and Jacek Mańdziuk. 2020. A Generic Metaheuristic Approach to Sequential Security Games. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 STACKELBERG SECURITY GAMES

We consider n step games with two players: the Defender (D) and the Attacker (A). In each time step $(\tau_1, \ldots, \tau_n)$ player p chooses action a_i^p ($p \in \{D, A\}, i = 1, \ldots, n$) from the set of available actions $M(s_i^p)$ where s_i^p is a state of player p in step τ_i. State s_i^p is determined by previous player's actions, his initial position and the opponent's actions. Players are not aware of the opponent's actions. For each state s there are four predefined payoffs $U^k(s)$ ($k \in \{A+, A-, D+, D-\}$) representing the Attacker's reward ($U^{A+}(s)$), their penalty ($U^{A-}(s)$), the Defender's reward ($U^{D+}(s)$) and their penalty ($U^{D-}(s)$), resp. Some of the states (usually those with high $U^{A+}(s)$ values) are distinguished as *targets*. If in any step $\tau_i (i = 1, \ldots, n)$ the Attacker and the Defender move to the same state (say s_k), then the game ends (the Attacker is intercepted) and players receive payoffs $U^{A-}(s_k)$ and $U^{D+}(s_k)$, resp. If the Attacker reaches any of the targets (say s_j) and is not intercepted, then the game ends with the respective payoffs equal $U^{A+}(s_j)$ and $U^{D-}(s_j)$. Otherwise, the game ends with neutral payoffs after n steps.

A *pure strategy* of a player is an assignment of one action to each *potentially reachable* state of the game. Let's denote a set of all pure strategies of player p by Σ^p. A *mixed strategy* π^p is a probability distribution over Σ^p. The game model employs Stackelberg Game (StG) principles: *first the Defender commits to their mixed strategy π^D and then the Attacker, being aware of π^D, determines their strategy π^A*. Let's denote by $U^p(\pi^D, \pi^A)$ an expected utility value of player p as a result of the game played according to mixed strategies π^D and π^A. Strong Stackelberg Equilibrium (SStE) [7] is defined as a pair (π^{D*}, π^{A*}) satisfying the following equations: $\pi^{D*} = \arg\max_{\pi^D} U^D(\pi^D, \pi^{A*})$ where $\pi^{A*} = \arg\max_{\pi^A} U^A(\pi^D, \pi^A)$. The second one defines the Attacker's best (optimal) response to Defender's strategy π^D and the first one selects the best Defender's strategy against the optimal Attacker's response. Additionally, it is assumed that the Attacker breaks ties in favor of the Defender. Solving StG means finding Defender's SStE strategy. EASG method is proposed to accomplish this task.

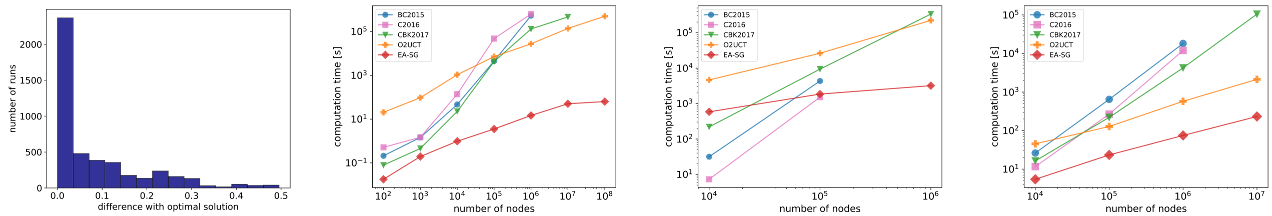
2 EVOLUTIONARY ALGORITHM FOR SGS

We consider a classical EA definition in which population of individuals of size p_{size} is maintained through generations until one of the stopping conditions is met: either the limit for generation number is reached or there is no solution improvement in a certain number of generations. Each ***chromosome*** (individual) represents some Defender's mixed strategy (a candidate SStE solution) in the form of a vector of pure strategies π_i^q and their respective probabilities p_i^q: $CH_q = \{(\pi_1^q, p_1^q), \ldots, (\pi_{l_q}^q, p_{l_q}^q)\}$, $\sum_{i=1}^{l_q} p_i^q = 1$, where l_q is the number of pure strategies included in the mixed strategy represented by that chromosome (l_q varies between individuals). A particular form of a pure strategy depends on game specificity. In the most common case, a pure strategy is represented as a list of Defender's actions in consecutive time steps. Each chromosome in the initial is composed of one pure strategy, randomly sampled.

Crossover. First, a subset of $p_c \cdot p_{size}$ individuals are randomly selected from the population, where p_c is crossover rate. Then, individuals from this subset are randomly paired and from each pair one new *offspring* chromosome is created in the following way. All pure strategies from the *parent* chromosomes are merged into one mixed strategy with their probabilities halved. Next, each pure strategy π_i^q in this newly created chromosome, except for the one with the highest probability, is removed with probability $(1 - p_i^q)^2$. Afterwards, probabilities of the remaining pure strategies are normalized.

Mutation is applied to each chromosome independently with some probability p_m. First, one pure strategy in the chromosome is randomly chosen. Then iteratively, starting from a randomly selected time step t_i up to the last time step t_n, an action in a

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.



(a) Accuracy. **(b) Warehouse Games.** **(c) Search Games.** **(d) FlipIt Games.**

Figure 1: Left: histogram of differences between the optimal Defender's payoffs and the payoffs obtained by EASG (for all games with known optimal solutions). Next: EASG time scalability vs state-of-the-art methods (logarithmic scales).

considered time step $t_j, i \leq j \leq n$ is changed to an action uniformly chosen among all actions feasible in that state.

Evaluation and Selection. Chromosome fitness equals a Defender's payoff when they play a mixed strategy encoded in that chromosome. Following [3] it is sufficient to iteratively check all pure Attacker's strategies and select the one with the highest Attacker's payoff (breaking ties in favor of the Defender - SStE condition). The Defender's payoff against the above best Attacker's response is a chromosome fitness value.

In selection, first e highest-fitted individuals (including the offsprings) are promoted. Next, a *binary tournament* is repeatedly executed until the next generation reaches p_{size} individuals. In the tournament two individuals are sampled with return from the current generation and the offsprings. A higher-fitted one is promoted with probability p_s, otherwise a lower-fitted one is promoted.

3 EXPERIMENTAL EVALUATION

Benchmark games. Properties of EASG are tested on 3 sets of multi-step games with variable characteristic: **Warehouse Games (WHG)** [4], **Search Games (SEG)** [1], and **FlipIt Games (FIG)** [9].

In WHG there is one Defender's unit which in a given turn can either move to an adjacent vertex or stay. Games of $T = 3 - 8$ steps were considered which led to game trees of $10^2 - 10^8$ nodes. For each T, 25 games downloaded from [8] were tested. In SEG the Defender controls several units and mobility of each of them is restricted to a subset of vertices. Furthermore, the Attacker leaves traces which makes them partially observable. In total, 90 games with $T = 4 - 6$, played on 3 different graph structures [1] were used. In FIG the Attacker attempts to infect certain nodes and the Defender may take actions to restore their control on the infected units. 60 FIG instances played on 3 different graph structures [2] with $T = 3 - 6$ were used. For each graph, 5 different payoffs structures were randomly drawn. Games were played in *No-Info* variant [2] (players did not know the results of their actions).

Benchmark methods. EASG was tested against four state-of-the-art methods for sequential general-sum extensive-form SGs BC2015 [1], C2016 [10], CBK2018 [2] and O2UCT [5, 6]. The first two are exact approaches, the remaining two yield approximate solution. BC2015, C2016 and CBK2018 are MILP-based. O2UCT relies on guided Monte-Carlo simulations.

Experimental setup. EASG is evaluated from three perspectives: accuracy, stability, and scalability. All results are obtained in 30 independent runs per game instance. In total EASG assessment is based on 9 000 trials (150 WHG, 90 SEG, 60 FIG, each tested 30 times) run on Intel Xeon Silver 4116 @ 2.10GHz with 256GB RAM.

Accuracy. A histogram of the differences between optimal and EASG solutions in all runs, across all game instances with known optimal solutions is presented in Fig. 1a. In the case of WHG, both exact methods were able to calculate the SStE for 100 games with $3 - 6$ time steps. In all tests involving larger games ($T = 7, 8$) the solution could not be reached due to extensive time requirements. For 72 out of these 100 games EASG obtained optimal solutions. The mean difference between EASG best results and the optimal ones was equal to 0.0013. For SEG, optimal solutions are known for 60 games with ($T = 4, 5$), out of which EASG found optimal strategies in 28 cases (47%). The average divergence from the optimal results equaled 0.0253. In FIG (which are recognized as highly challenging for SGs methods due to extensive search space) EASG managed to achieve optimal solutions in 73% of the cases (exact methods were capable of finding solutions for 45 test games, out of which EASG yielded the same solutions for 33 games). The average divergence from the optimal results equaled 0.0087.

Stability. Since EASG is highly non-deterministic, the ability to repeatedly reproduce good results is of paramount importance. For 45% of games, standard deviation was equal to 0. The mean standard deviation equaled 0.0059 with the maximal value 0.1629.

Time scalability. Figure 1 compares time efficiency of EASG vs four state-of-the-art algorithms summarized above. First, all games of a given type (separately WHG, SEG, FIG) were divided into subsets of instances with pairwise comparable game tree sizes (pairwise equal after rounding to the nearest power of 10). Then, for each subset the running times of all game instances belonging to that subset were averaged and plotted. Due to exceeding time limit of 200 hours per trial, for biggest games the results of exact methods (BC2015 and C2016) are not plotted.

Summary. EASG has proven to be a robust method which scales in time visibly better than state-of-the-art approaches while providing optimal or close-to-optimal solutions. It can be regarded as a MILP alternative when calculation of an exact solution is infeasible.

ACKNOWLEDGMENTS

This work was supported by the National Science Centre, grant number 2017/25/B/ST6/02061.

REFERENCES

[1] Branislav Bošanský and Jiří Čermak. 2015. Sequence-Form Algorithm for Computing Stackelberg Equilibria in Extensive-Form Games. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 805–811.

[2] Jakub Černỳ, Branislav Boỳanskỳ, and Christopher Kiekintveld. 2018. Incremental strategy generation for Stackelberg equilibria in extensive-form games. In *Proceedings of the 2018 ACM Conference on Economics and Computation*. ACM, 151–168.

[3] Vincent Conitzer and Tuomas Sandholm. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*. ACM, 82–90.

[4] Jan Karwowski and Jacek Mańdziuk. 2019. A Monte Carlo Tree Search approach to finding efficient patrolling schemes on graphs. *European Journal of Operational Research* 277, 1 (2019), 255–268.

[5] Jan Karwowski and Jacek Mańdziuk. 2019. Stackelberg Equilibrium Approximation in General-Sum Extensive-Form Games with Double-Oracle Sampling Method. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, Montreal, Canada, 2045–2047.

[6] Jan Karwowski and Jacek Mańdziuk. 2020. Double-oracle sampling method for Stackelberg Equilibrium approximation in general-sum extensive-form games. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, New York, NY, USA, 7 pages.

[7] George Leitmann. 1978. On generalized Stackelberg strategies. *Journal of optimization theory and applications* 26, 4 (1978), 637–643.

[8] Jacek Mańdziuk, Jan Karwowski, and Adam Żychowski. 2019. Simulation-based methods in multi-step Stackelberg Security Games in the context of homeland security. (2019). https://sg.mini.pw.edu.pl

[9] Marten Van Dijk, Ari Juels, Alina Oprea, and Ronald L Rivest. 2013. FlipIt: The game of stealthy takeover. *Journal of Cryptology* 26, 4 (2013), 655–713.

[10] Jiří Čermák, Branislav Bošanský, Karel Durkota, Viliam Lisý, and Christopher Kiekintveld. 2016. Using Correlated Strategies for Computing Stackelberg Equilibria in Extensive-Form Games. In *Proceedings of the Thirty AAAI Conference on Artificial Intelligence*. 439–445.