Sequential Stackelberg Games with Bounded Rationality

Jan Karwowski*, Jacek Mańdziuk, Adam Żychowski

Faculty of Mathematics and Information Science, Warsaw University of Technology, Koszykowa 75, 00-662 Warsaw, Poland

Abstract

Stackelberg Games (SGs) assume the perfect rationality of players. However, in real-life situations modeled by SGs, the followers may act not perfectly rationally, as their decisions may be affected/bounded by biases of various kinds, reflecting human behavior in the real world. Anchoring Theory (AT) is one of the popular bounded rationality (BR) models. It postulates that humans have a tendency to flatten the probabilities of the available options, i.e. their probability distribution is perceived as more uniform than is actually the case. This paper proposes a formulation of AT in sequential extensive-form SGs (*ATSG*) and its linearized approximate version (ATSGL) suitable for Mixed-Integer Linear Program (MILP) solution methods. ATSGL is implemented in three MILP/LP state-of-the-art methods for solving sequential SGs and compared with two recent non-MILP metaheuristic approaches based on the original non-simplified ATSG formulation, which rely on Monte Carlo sampling (*O2UCT*) and Evolutionary Algorithms (*EASG*), respectively. Experimental evaluation indicates that non-MILP heuristic approaches provide better solutions and scale better in time than MILPs in the AT setting. The efficacy of ATSG is further evaluated in experiments involving humans as followers, which show that it is more advantageous to use the ATSG leader's strategy than the Stackelberg Equilibrium strategy, which assumes the perfect rationality of the follower. The results confirm the existence of the human follower's AT-bias and the possibility to exploit

*Corresponding author

it by the leader. An additional advantage of heuristic methods is the flexibility of the potential BR formulation they are able to incorporate.

Keywords: Sequential Games, Stackelberg Games, Bounded Rationality, Anchoring Theory, MILP

1. Introduction

Stackelberg Games (SGs) [1, 2] are a game-theoretic model which has attracted considerable interest in recent years, in particular in the Security Games area [3]. In its simplest form, a Stackelberg Security Game (SSG) has two players: a *leader* who commits to a (mixed) strategy first, and a *follower* who makes their commitment knowing the decision of the leader. The above asymmetry of the players occurs in many practical situations (e.g. in commerce [4, 5, 6]) or strongly corresponds to the interactions between law enforcement (leader) and smugglers, terrorists or poachers (followers) modeled by SSGs [7, 8, 9, 10].

A fundamental assumption in SGs is that the follower makes an optimal, perfectly rational decision exploiting knowledge of the leader's commitment. However, in real-life scenarios followers may suffer from cognitive biases, unwillingness to cooperate (in the case of multiple leaders / followers) [11] or bounded rationality, which leads them to suboptimal decisions [12, 13, 7].

On a general note, *bounded rationality* (BR) [14] in problem-solving refers to the limitations of decision-makers that result in suboptimal actions. Except for limited cognitive abilities, BR can be attributed to partial knowledge about the situation/problem, limited resources, or imprecisely defined goals [15, 16]. The most popular models of BR are: Prospect Theory (PT) [17], Anchoring Theory (AT) [18], Quantal Response (QR) [19] and Framing Effect (FE) [20]. Each of these models makes specific problem-related assumptions and each of them has some degree of experimental justification, though none of them could be regarded as widely agreed-upon or a leading BR formulation.

The concept of BR plays an important role in SSGs, as in real-world applications of SSGs the follower's role is performed by humans, e.g. terrorists, poachers or criminals, who usually suffer from BR limitations. However, due to the inherent non-linearity of BR models the implementations proposed so far in the literature refer to single-step games only.

1.1. Contribution

In this paper, the AT formulation for single-step normal-form games is extended to the case of *sequential extensive-form games* (abbreviated as ATSG). Furthermore, an approximation of ATSG is proposed that avoids non-linear constraints (ATSGL), suitable for a wide range of MILP/LP (Mixed Integer Linear Program/Linear Program) approaches. ATSGL is implemented for three state-of-the-art methods for solving extensive-form SSGs [21, 22, 23]. Furthermore, two other non-MILP heuristic methods for solving SSG are adequately modified to incorporate ATSG: one that relies on Monte Carlo sampling [24, 25] and another one which employs Evolutionary Algorithms [26]. All five methods are experimentally evaluated on three sets of test games: Warehouse Games (WHG) [27, 28], Warehouse Games with more diverse payoffs (WNZ) [25, 28] and Search Games (SEG) [29, 23].

Additionally, the efficacy of the proposed AT implementation is tested online on a dedicated game-playing portal [28] by volunteers (university students) who assume the role of the follower.

In summary, the main contributions of this paper can be listed as follows:

(i) Proposition of ATSG, i.e. an Anchoring Theory formulation in sequential (multi-step) Stackelberg Games;

(ii) Efficient MILP-suitable linearized simplification of ATSG (ATSGL);

(iii) Implementation of ATSGL in three MILP/LP state-of-the-art methods for solving sequential Stackelberg Games (two exact and one approximate);

(iv) Implementation of ATSG in two approximate non-MILP approaches to Stackelberg Games relying on Monte Carlo sampling [25] and Evolutionary Algorithms [26], respectively;

(v) Experimental evaluation with respect to the quality of payoffs and time efficiency of the five above-mentioned approaches in BR settings, on three sets of benchmark games;

(vi) Evaluation of the efficacy of ATSG in online experiments involving humans playing the role of the follower.

This paper substantially extends the extended abstract published in [30].

1.2. Related work

To the best of our knowledge, the concept of BR has been addressed in SSGs only in the context of single-step games and this work is the first to consider a BR implementation in sequential SSGs.

For single-step SSGs, one of the main BR implementations is the COBRA [13] method, which modifies DOBSS MILP [31] to address AT with ϵ-optimality models. A similar approach was taken by Yang et al. [12] who proposed BR models relying on PT and QR, respectively, and demonstrated their suitability in SSGs based on experiments with human players. The SHARP system [32] considers certain game-related aspects (e.g. past performance and similarity of game conditions) which are taken into account in repeated SSGs played against human adversaries. MATCH [33] optimizes the leader's strategy against a worst-case outcome within some error bound (i.e. assuming certain deviations from the follower's optimal strategy). Another approach – BRQR [34], proposed by Yang et al., refers to the idea of QR. The method is further improved in the SU-BRQR system [35], which introduces a subjective utility function for the follower with parameters tuned in experiments involving humans. QR was also used to model bounded rationality for the optimal defense resources allocation for power systems [36]. All the above-mentioned works are implementations of BR models in MILP/LP formulations of single-step SSGs.

More recent BR approaches are more practically-oriented and focused on real-life SSG applications in particular domains. The work of [37] uses a logistic regression model for optimizing the signaling strategy in the poaching domain against boundedly rational poachers. In [38] a predictive framework for wildlife protection is proposed that accounts for imperfect crime information and uncertainty in wildlife data. The work of [39] presents a solution based on PT modeling with regret minimization related to past data, with an application in cybersecurity - in the so-called Cyber Camouflage Games [40]. A reinforcement learning approach to discovering strategies for poaching prevention that considers information uncertainty is presented in [41]. Two other papers [42, 43] adopt a convolutional neural network to learn the adversary's behavior in network security games, which can be applied in smuggling prevention. Both methods rely on using historical data and machine learning techniques to approximate and predict the follower's behavior, whereas in our method no historical data is used to define or tune the model.

In summary, while the above-mentioned models represent quite a wide range of BR approaches, none of them could be applied directly to sequential SSGs. Furthermore, to the best of our knowledge, this work is the first BR implementation in sequence-form games.

1.3. Motivation

This paper combines the following two concepts which are generally considered separately in the literature: (1) BR models in Security Games and (2) efficient solution methods for sequential SGs.

In both areas significant progress has been observed in recent years. However, to our knowledge, the concept of BR has been addressed in Security Games exclusively in the context of single-step games. An example of this is the recently emerged, fast-growing field of Green Security Games [44, 7], in which game-theoretical models exploit the non-rational behavior of the followers (e.g. poachers or illegal forest extractors) to maximize the effectiveness of protection activities.

At the same time, several algorithms have been proposed for large-scale sequential SGs. These algorithms utilize different techniques which yield an exact solution, e.g. sequence-form [23], correlated equilibria [21]. Another class of approaches employs heuristics to improve the equilibrium computation time. Notable contributions in the area include, first of all, a method that folds large parts of the game tree into small structures called gadgets, calculates an equilibrium in the abstracted game using LP, and iteratively unfolds gadgets that have the highest impact on the equilibrium strategy [22].

A new line of approximate methods based on Monte Carlo Tree Search (MCTS) combined with UCT sampling [45] of the game tree to iteratively improve the leader's strategy has been recently proposed in [46]. Subsequent works have been developed along two main directions: (1) the Mixed-UCT method [47, 27], in which imperfect-information UCT is applied to sample gradually stronger follower in order to derive an approximation of the optimal leader's mixed strategy, and (2) O2UCT [24, 25], which interleaves the update phases of the sampled leader's strategy with finding the follower's best response.

A distinct group of methods use Evolutionary Algorithms for SSE computation in different areas of SGs [48, 26, 49]. Yet, another powerful recent approach applies finite-state machines as a compact representation of the abstracted strategy [50].

Successful research on the crossroads of the two above-described research directions may allow for tackling practical, large-scale problems arising in real-life scenarios.

The majority of methods described in the literature assume that the follower chooses an optimal response strategy, i.e. behaves in a perfectly rational way. However, in real-life scenarios in which humans are involved, this assumption may not hold, due to the imperfections of the players' senses, cognitive biases, partial knowledge about the problem, or imprecisely defined goals. Decision-making limitations of this type are of paramount importance in practical security scenarios modeled by Stackelberg Games, in which the role of the follower is "played" by terrorists, thieves, poachers, or hackers. Defenders often have little knowledge about such opponents, their perception, or cognitive abilities. Hence, in practice, playing the SE strategy by the leader may – in fact – be suboptimal. Furthermore, human-follower BR biases can potentially be exploited by the leader and may increase their payoff in Stackelberg Security Games.

Among several BR models introduced in the literature, the AT approach has been chosen since it has been already successfully applied to single-step SGs [13] and is furthermore intuitively justified in real-life cases when only limited observation of the leader's strategy is possible.

There is no scientific consensus on which of the BR models best mimics human decision making, however, AT is claimed by psychologists to accurately reflect human strategies of revenue/expense allocation [51], which is the essence of SSGs. Furthermore, AT has been successfully applied in real-world systems, e.g. structured clinical judgment [52] or retirement investing planing [53]. Finally, our choice of AT as a BR model is supported by promising results of the application of AT to single-step SGs in experiments involving humans [13].

Generally speaking, AT [18] assumes the existence of a bias of a person who observes some events (for instance, surveils the opponent's strategy in SSG), where their assessment of the probability distribution is skewed towards a uniform distribution. Formally, for any probability distribution over a finite set X, let us denote the probability of $x \in X$ as q_x. The observer believes that this probability is equal to $q'_x = q_x(1-\alpha) + \alpha/|X|$, where $0 < \alpha < 1$ is a parameter of the AT bias and $|X|$ is the cardinality of X.

The underlying claim of this paper is that in SGs, the leader, being aware of the follower's AT bias, can *effectively exploit* this knowledge in their own mixed strategy formulation.

1.4. Definitions

Throughout this paper the notation from [21] is used so as to easily refer to the method proposed therein. Sequential games are represented as *Extensive-Form Games (EFGs)*, i.e. tuples $G = (\mathcal{N}, \mathcal{H}, \mathcal{Z}, \mathcal{A}, \rho, u, \mathcal{I})$, where $\mathcal{N} = \{l, f\}$ is a set of players, the leader and the follower, respectively. $\mathcal{H}$ is a set of game nodes that compose a game tree with the root node representing the initial game position. $\mathcal{Z} \subset \mathcal{H}$ is a set of leaves representing terminal game states. $\mathcal{A}$ is a family of sets A_h which $\forall h \in \mathcal{H} \setminus \mathcal{Z}$ define possible actions from each nonterminal node h. $\rho : \mathcal{H} \setminus \mathcal{Z} \rightarrow \mathcal{N}$ is a function that defines the acting player for a given node. $u = \{u_l, u_f\},\ u_i : \mathcal{Z} \rightarrow \mathbb{R}, i \in \mathcal{N}$ is a family of utility functions that for a terminal node map from a player to a game outcome for this respective player. $\mathcal{I}$ is a family of Information Sets (ISs); each $I \in \mathcal{I}$ defines states that are indistinguishable to the acting player. $\mathcal{I}$ satisfies the following conditions:

- $\mathcal{I}$ partitions $\mathcal{H} \setminus \mathcal{Z}$,
- $\forall I \in \mathcal{I} \quad \forall h_1, h_2 \in I \quad \rho(h_1) = \rho(h_2)$ – all nodes in a given IS have the same acting player,
- $\forall I \in \mathcal{I} \quad \forall h_1, h_2 \in I \quad A_{h_1} = A_{h_2}$ – for a given IS, the set of available actions is the same in all nodes.

Additionally, $A(I)$ denotes the set of actions available in I and $\mathcal{I}_i, i \in \mathcal{N}$ signifies a family of ISs with acting player i ($\mathcal{I} = \mathcal{I}_l \cup \mathcal{I}_f$).

Moreover, the games are assumed to satisfy the *perfect recall* property, i.e. throughout the game each player is fully aware of previous ISs visited by him/her and actions taken by him/her in that ISs.

In EFG, a *pure strategy* of a player assigns to each IS in which the player is an acting player a particular action to be played in that IS. A set of all pure strategies of player i will be denoted by Π_i. A *mixed strategy* of a player is a probability distribution over pure strategies of that player. Δ_i will be used to denote a set of mixed strategies of player i. Elements of Π_i and Δ_i will be denoted by π and δ with adequate indices (l-leader, f-follower), respectively.

A *behavior strategy* is an assignment of a probability distribution of actions for each IS, that a player can reach during the game. It can be viewed as a tree with nodes representing player's ISs and edges representing actions (labeled with their probabilities). The notions of *mixed strategy* and *behavior strategy* will be used interchangeably as they are equivalent in games with perfect recall.

The notation of u functions will be overloaded, so that $u_i(\pi_l, \pi_f)$ will denote the i-th player's utility after the pure strategy profile (π_l, π_f) has been played. Similarly, $u_i(\delta_l, \delta_f)$ will denote the expected utility of the i-th player for the mixed strategy profile (δ_l, δ_f).

Each node in a game tree is uniquely defined by a pair of sequences: the leader's actions and the follower's actions, both of which lead to that node. These sequences will be denoted by σ_l and σ_f, respectively. A pair of sequences (σ_l, σ_f) is *compatible* if it leads to a terminal node in the game tree. Utility values in terminal nodes for a compatible pair of sequences will be denoted by $u_i(\sigma_l, \sigma_f), i \in \mathcal{N}$. Following [21], for any pair (σ_l, σ_f) an auxiliary function $g_i(\sigma_l, \sigma_f)$ is defined which yields a value of $u_i(\sigma_l, \sigma_f)$ if the sequences are compatible and 0 otherwise. Finally, $\sigma_i(h), i \in \mathcal{N}, h \in \mathcal{H}$ will denote a sequence of actions of the i-th player which led to node h and $I_i(\sigma_i)$ is the IS in which the last action from σ_i was played.

The goal of SG is to find a *Stackelberg Equilibrium (SE)*, i.e. a strategy profile (δ_l^*, δ_f^*) which is a solution of the following set of equations:

$$\begin{cases} \delta_l^* = \arg\max_{\delta_l \in \Delta_l} u_l\left(\delta_l, OR(\delta_l)\right) \\ \delta_f^* = OR(\delta_l^*) \\ OR(\delta_l) = \arg\max_{\delta_f \in \Delta_f} u_f(\delta_l, \delta_f) \end{cases} \tag{1}$$

In the above, $OR(\delta_l)$ denotes the *unique* optimal follower's response to the leader's strategy δ_l. It is of importance that OR may not be well-defined when there is more than one optimal follower's response δ_f^*. For this reason, SE is often extended to the form of a *Strong Stackelberg Equilibirum (SSE)* [54], defined below. Note that OR is a set now:

$$\begin{cases} \arg\max_{\delta_l \in \Delta_l, \delta_f \in OR(\delta_l)} u_l(\delta_l, \delta_f) \\ OR(\delta_l) = \left\{ \delta_f \middle| (\forall \delta_f') u_f(\delta_l, \delta_f) \geq u_f(\delta_l, \delta_f') \right\}. \end{cases} \tag{2}$$

In SSE, in addition to (1), a specific provision is considered. In the case of a tie among the follower's best response strategies, the one of them that maximizes the leader's utility is selected (if there are more such strategies any one of them is chosen). In this paper the SSE version of SE is considered.

An important property of SE (and SSE) in finite games is that there always exists an equilibrium strategy profile in which the follower's strategy is in the form of a pure strategy [55]. This observation is a cornerstone of the methods used to compute SE (SSE).

1.5. Structure

The rest of the paper is organized as follows. Sections 2 and 3 present straightforward (ATSG) and approximate (ATSGL) implementations of Anchoring Theory for Stackelberg Games, the latter of which is specifically tailored to MLIP methods. The next section presents an experimental setup and evaluation results of approximate AT implementations for five state-of-the art approaches to SGs: three based on MILP, which employ a linear ATSGL formulation, and two relying on metaheuristics. Section 5 describes online experiments involving humans in the role of the follower against three different methods for the calculation of the leader's strategy: *fully rational* SSE (described in this section), SSE biased by ATSG (described in Section 2) and SSE biased by ATSGL (described in Section 3). These online tests show that humans are indeed affected by the AT-bias and that considering this bias in the leader's strategy improves his/her expected outcomes. The paper is summarized in Section 6. Appendix A lists the basic notation and abbreviations used throughout.

2. Anchoring Theory in Sequential Games

As stated above, the implementations of AT in SGs presented in the literature are limited to single-step games only. There are two straightforward ways to generalize AT to sequential games. The first one is to transform an extensive-form game to its normal-form where each action of a player is equivalent to a pure strategy. Such an approach, however, would introduce a *global distortion* of probabilities and is, therefore, inaccurate if the opponent's behavior is considered separately at each decision point. The other possibility is to apply the AT distortion *locally* - i.e. directly to the probability distribution for each IS that forms a behavior strategy of the player. This approach seems to be more intuitive, though, due to non-linear constraints, poses problems for sequence-form MILP methods.

In the remainder of this section, two state-of-the-art exact MILP methods for SSE computation in sequential games are first presented (Sections 2.1 and 2.2). Then, in Section 2.3 a further analysis of the above-mentioned problem with the *local* AT implementation in MILP methods (ATSG) is performed, followed by an introduction of the local ATSG approximation (ATSGL). ATSGL allows for the AT distortion in every node of the behavior strategy similar to ATSG, albeit avoids non-linear constraints.

2.1. Sequence-form based MILP

The first method (which will henceforth be referred to as *BC2015*), introduced in [23], uses a sequence-based representation of the extensive-form game. The method assigns a probability to each possible move sequence of each player and then calculates utility values using functions $g_i, i \in \{l, f\}$ introduced in Section 1.4. The following MILP is formulated and solved to obtain the SSE of the game:

$$\max_{p,r,v,s} \sum_{z \in Z} p(z) u_l(z), \text{ s.t.} \tag{3}$$

$$v_{I_f(\sigma_f)} = s_{\sigma_f} + \sum_{I' \in \mathcal{I}_f | \sigma_f(I') = \sigma_f} v_{I'} \quad + \sum_{\sigma_l \in \Sigma_l} r_l(\sigma_l) g_f(\sigma_l, \sigma_f) \tag{4}$$

$$r_i(\emptyset) = 1 \quad \forall i \in \mathcal{N} \tag{5}$$

$$r_i(\sigma_i) = \sum_{a \in A_i(I_i)} r_i(\sigma_i a) \quad \forall i \in N \; \forall I_i \in \mathcal{I}_i, \sigma_i = \sigma_i(I_i) \tag{6}$$

$$0 \leq s_{\sigma_f} \leq (1 - r_f(\sigma_f)) \cdot M \quad \forall \sigma_f \in \Sigma_f \tag{7}$$

$$0 \leq p(z) \leq r_i(\sigma_i(z)) \quad \forall i \in N \quad \forall z \in \mathcal{Z} \tag{8}$$

$$1 = \sum_{z \in Z} p(z) \tag{9}$$

$$r_f(\sigma_f) \in \{0, 1\} \quad \forall \sigma_f \in \Sigma_f$$

$$0 \leq r_l(\sigma_l) \leq 1 \quad \forall \sigma_l \in \Sigma_l$$

In the above, the r_l and r_f families of variables denote the probabilities of playing a given move sequence by the leader and the follower, respectively. r_f

are binary variables, as there always exists an SSE profile in which the follower plays a pure strategy [55]. Equations (5) and (6) tie together the probability of playing a particular move sequence with the sum of probabilities of playing that sequence extended by one move. Variables p are defined for terminal nodes of the game and each of them denotes a probability of reaching the respective node. Equations (8) and (9) establish a connection between variables p and r, so that a probability of reaching a terminal node is equal to the probability of the move sequence leading to this node. Variables v are used to calculate the follower's utility after playing particular move sequences and variables s are slacks. Equations (4) and (7) ensure that the follower's strategy defined by r_f is an optimal response to the leader's strategy.

Please note that each sequence-form variable $r_l(a_1 a_2 \cdots a_n) = q(a_1)q(a_2)\cdots q(a_n)$ is a product of probabilities of subsequent actions in the behavior strategy.

2.2. SEFCE-based approach

The other exact approach for calculating SSE in sequential games [21] considered in this study, referred to as *C2016*, is an iterative method which alternates between two phases: solving MILP/LP to find a Stackelberg Extensive-Form Correlated Equilibrium (SEFCE) in the sequence-form game representation, and a SEFCE refinement with a dedicated procedure relying on an LP modification towards SSE. Since the equilibrium refinement part is not affected by the implementation of AT, it will not be discussed here. The SEFCE part of the method, defined by a set of equations (10)-(16), is built around variables that define the probabilities of particular sequences being played by the players [21]. The following LP definition employs the notion of relevant sequence pairs (rel) – formally introduced in Definition 3 in [21].

$$\max_{p,v} \sum_{\sigma_l \in \Sigma_l} \sum_{\sigma_f \in \Sigma_f} p(\sigma_l, \sigma_f) g_l(\sigma_l, \sigma_f) \tag{10}$$

$$\text{s.t.} \qquad p(\emptyset, \emptyset) = 1; 0 \leq p(\sigma_l, \sigma_f) \leq 1 \tag{11}$$

$$p(\sigma_l(I), \sigma_f) = \sum_{a \in A(I)} p(\sigma_l(I)a, \sigma_f) \;\; \forall I \in \mathcal{I}_l, \forall \sigma_f \in rel(\sigma_l(I)) \tag{12}$$

$$p(\sigma_l, \sigma_f(I)) = \sum_{a \in A(I)} p(\sigma_l, \sigma_f(I)a) \;\; \forall I \in \mathcal{I}_f, \forall \sigma_l \in rel(\sigma_f(I)) \tag{13}$$

$$v(\sigma_f) = \sum_{\sigma_l \in rel(\sigma_f)} p(\sigma_l, \sigma_f) g_f(\sigma_l, \sigma_f) + \sum_{I \in \mathcal{I} | \sigma_f(I) = \sigma_f} \sum_{a \in A_f(I)} v(\sigma_f a) \tag{14}$$

$$v(I, \sigma_f) \geq \sum_{\sigma_l \in rel(\sigma_f)} p(\sigma_l, \sigma_f) g_f(\sigma_l, \sigma_f(I)a) + \sum_{I' \in \mathcal{I}_f | \sigma_f(I') = \sigma_f(I)a} v(I', \sigma_f),$$
$$\forall I \in \mathcal{I}_f, \forall \sigma_f \in \bigcup_{h \in I} rel(\sigma_l(h)), \forall a \in A(I) \tag{15}$$

$$v(\sigma_f(I)a) = v(I, \sigma_f(I)a) \forall I \in \mathcal{I}_f, \forall a \in A(I) \tag{16}$$

The main variables in the above LP are $p(\sigma_l, \sigma_f)$ which describe the correlation plan and represent a probability that the correlation device will give a suggestion of the respective sequences of moves (σ_l, σ_f) being played by the players. Implicitly, they define the players' strategies. Objective (10) maximizes the leader's utility. Constraints (11) – (13) are conceptually similar to constraints (5) – (6) and they ensure that the correlation plan is correct, i.e. the probability of playing a given sequence is the sum of the probabilities of playing the sequences that extend this given sequence by one action. v are auxiliary variables which guarantee that the suggested σ_f is the optimal follower's response.

The crucial constraints (from the AT perspective) are (14) and (15), which assure that the selected follower's strategy yields utility not worse than any other strategy. An implementation of AT requires changing the perception of $p(\sigma_l, \sigma_f)$ variables, so as to include the anchoring bias - the details are presented in the following section.

The scheme to solve the above LP alternates iteratively with the mentioned refinement procedure, which can be summarized as follows. First, all p variables whose values are not binary, i.e. the probabilities of playing the respective actions are between 0 and 1, are identified for the current LP solution. Next, the LP is extended with constraints that force those values to be 0 or 1, which leads to various possible refinements of the LP. The refined LPs are solved and among the feasible ones, the one that yields the highest leader's utility is chosen.

No modifications to this procedure, compared to its original formulation [21], are required in our method. Similarly to *BC2015*, variables p used in (10)–(16) are products of probabilities q from the respective behavior strategy.

2.3. Anchoring Theory modification

ATSG is implemented as a distorted follower's perception of the leader's behavior strategy. Let's denote by $q(i)$ the probability of action i being chosen by the leader in a given IS, stemming from the behavior strategy. The most straightforward implementation of AT (though, non-linear in sequence-form games) is to change the probability of taking this action to $q'(i) = (1 - \alpha q(i)) + \alpha/M$, where M is the number of actions available in this IS. However, in sequence-form games, for a given sequence of leader's actions $\sigma_l = a_1, a_2, a_3, \ldots, a_n$, a probability of playing it, based on the behavior strategy, would be $p(\sigma_l) = q(a_1)q(a_2)\cdots q(a_n)$ and the distorted AT probability would become:

$$\begin{aligned} p'(\sigma_l) = ((1-\alpha)q(a_1) + \alpha/M_1)((1-\alpha)q(a_2) + \alpha/M_2)\cdots \\ ((1-\alpha)q(a_n) + \alpha/M_n), \end{aligned} \tag{17}$$

where M_i is the number of actions available for the IS in which a_i is played.

Let us observe again that variables r in the MILP formulation (3)–(9) are products of $q(a_i)$ values presented in (17) and as such cannot be expressed in linear form with respect to $q(a_i)$. A similar remark applies to LP (10)–(16) where variables p are also products of probabilities in the behavior strategy. Consequently, a straightforward application of the above AT modification to those programs would end-up with non-linear constraints, unsuitable for the MILP/LP formulation.

Consequently, this paper proposes to simplify the above ATSG formulation (17) by dropping the distortion coefficients from all probabilities except the last one:

$$\begin{aligned} p''(\sigma_l) &= q(a_1)q(a_2)\cdots q(a_{n-1})((1-\alpha)q_{a_n} + \alpha/M_n) \\ &= q(a_1)q(a_2)\cdots q(a_{n-1})\cdot\alpha/M_n + (1-\alpha)q(a_1)q(a_2)\cdots q(a_{n-1})q(a_n) \\ &= p(init(\sigma_l))\alpha/M_n + (1-\alpha)p(\sigma_l), \end{aligned} \tag{18}$$

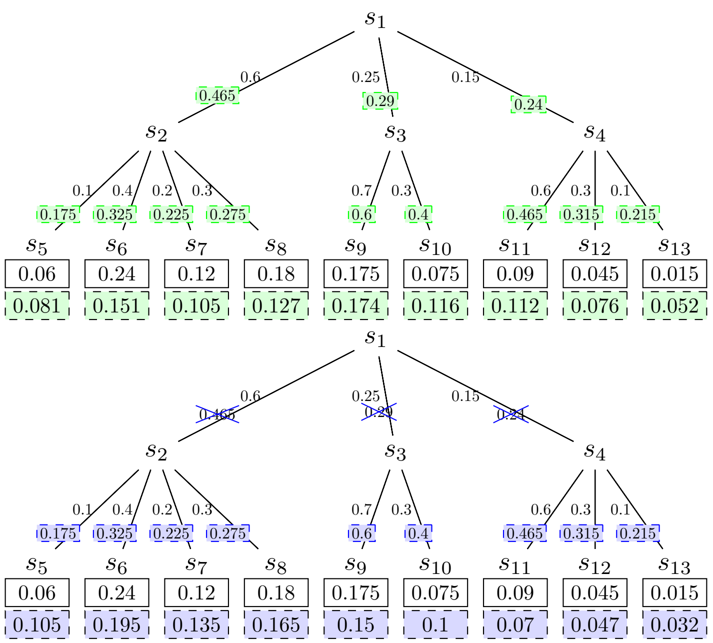


Figure 1: An illustrative example of the difference between the exact (upper tree) and simplified (lower tree) Anchoring Theory formulations.

where $init(\cdot)$ is a function which outputs a sequence of moves without the last one. ATSGL (18) is a simplified version of ATSG (17), well suited to MILP/LP formulations of sequence-form games. The difference between ATSG and ATSGL is visualized in Figure 1.

The relations among the probabilities of the leader's actions within a single IS are the same for both eqs. (17) and (18), i.e. $\forall \sigma_l^1, \sigma_l^2 \quad I(\sigma_l^1) = I(\sigma_l^2) \Rightarrow p'(\sigma_l^1)/p'(\sigma_l^2) = p''(\sigma_l^1)/p''_{(}\sigma_l^2)$, where $p'(\sigma), p''(\sigma)$ represent the probabilities of a sequence σ in a given IS calculated according to (17) and (18), respectively. Furthermore, for a given sequence σ_l, for small values of α the difference $|p''(\sigma_l) - p'(\sigma_l)|$ is also small.

The resulting p'' values do not represent a proper probability distribution since they do not sum up to one. Normalization is not needed though, as they are used only to make comparisons between the distorted utility of various follower strategies in the above-mentioned MILP/LP programs. The results of such comparisons are independent of p'' normalization.

2.4. Modification of MILP/LP based methods

The ATSGL formulation (18) was implemented for both state-of-the-art exact methods for sequential SGs: *BC2015* and *C2016*. The changes imposed by the incorporation of ATSGL are presented in eqs. (19)–(21). The parts of the equations which are removed from the baseline MILP formulations are crossed out and replaced with the respective parts emphasized in bold. The new formulation arises directly from eq. (18).

2.4.1. Sequence-form method

BC2015 directly utilizes the sequence-form game representation and, unlike *C2016*, is not an iterative method, i.e. it relies on solving a single MILP instance with a larger number of variables to obtain the game solution. The implementation of ATSGL (according to eq. (18)) requires replacing part of equation (4) with a variant that uses the ATSGL distortion in the calculation of the follower's utility:

$$v_{I_f(\sigma_f)} = s_{\sigma_f} + \sum_{I' \in \mathcal{I}_f | \sigma_f(I') = \sigma_f} v_{I'} \quad + \cancel{\sum_{\sigma_l \in \Sigma_l} r_l(\sigma_l) g_f(\sigma_l, \sigma_f) +}$$
$$+ \sum_{\boldsymbol{\sigma_l \in \Sigma_l}} \boldsymbol{r_l(\sigma_l) g_f(\sigma_l, \sigma_f)} + \boldsymbol{\alpha / M_{I(\sigma_l)} g_f(\sigma_l, \sigma_f) r_l(init(\sigma_l))} \tag{19}$$

The remaining part of MILP does not change.

2.4.2. SEFCE method

In *C2016*, ATSGL is implemented according to eq. (18) through the modification of constraints (14) and (15), which are replaced with constraints (20) and (21) presented below:

$$\begin{aligned} v(\sigma_f) = {} & \xcancel{\sum_{\sigma_l\in rel(\sigma_f)} p(\sigma_l,\sigma_f) g_f(\sigma_l,\sigma_f)} \\ & \boldsymbol{\sum_{\sigma_l\in rel(\sigma_f)} g_f(\sigma_l,\sigma_f)\,(p(\sigma_l,\sigma_f)+\alpha/M_I\cdot p(init(\sigma_l),\sigma_f))}+ \\ & + \sum_{I\in\mathcal{I}|\sigma_f(I)=\sigma_f}\ \sum_{a\in A_f(I)} v(\sigma_f a), \quad \forall\sigma_f\in\Sigma_f \end{aligned} \tag{20}$$

$$\begin{aligned} v(I,\sigma_f) \geq {} & \xcancel{\sum_{\sigma_l\in rel(\sigma_f)} p(\sigma_l,\sigma_f) g_f(\sigma_l,\sigma_f(I)a)} \\ & \boldsymbol{\sum_{\sigma_l\in rel(\sigma_f)} g_f(\sigma_l,\sigma_f(I)a)\,(p(\sigma_l,\sigma_f)+\alpha/M_I\cdot p(init(\sigma_l),\sigma_f))}+ \\ & + \sum_{I'\in\mathcal{I}_f|\sigma_f(I')=\sigma_f(I)a} v(I',\sigma_f), \quad \forall I\in\mathcal{I}_f, \forall\sigma_f\in\bigcup_{h\in I} rel(\sigma_l(h)), \forall a\in A(I) \end{aligned} \tag{21}$$

The LP in *C2016* does not encompass the variables describing the probabilities of playing σ_l alone (p_{σ_l}), but instead refers to a correlation plan which provides suggestions on playing pairs (σ_f,σ_l). Moreover, $p(\sigma_l,\sigma_f)$ equals $p(\sigma_l)$ only if marginal probabilities satisfy

$$p(\sigma_f)\in\{0,1\}, \tag{22}$$

i.e. the correlation plan suggests the follower to play a pure strategy. In the above ATSGL version of *C2016*, defined by equations (20)–(21), the condition (22) may not initially hold for all σ_f, but must be fulfilled for all of them at the completion of *C2016*, since (22) constitutes a stopping condition for this method.

2.4.3. Game abstraction method

In 2018 a new approach to extensive-form SSGs that folds game subtrees into nodes called *gadgets* and then incrementally unfolds them to refine the solution was proposed [22]. The method (henceforth referred to as *CBK2018*) internally employs *C2016* to solve the abstracted (smaller) games. *CBK2018* was formulated by its authors in two variants: as an exact method and as a heuristic time-optimized approach, with an experimental evaluation provided only for the latter variant [22]. Consequently, this study also focuses on the heuristic formulation of *CBK2018* and following a recommendation from [22] sets the internal method's parameters to $\epsilon = 0.3, \sigma = 0.4$ which assures fast convergence, albeit at the cost of some deviation from the optimal results. In the ATSGL modification of *CBK2018* original *C2016* constraints (14)–(15) are replaced with their ATSGL versions (20)–(21).

3. Heuristic Approximations of ATSG

The above-discussed three ATSGL modifications of MILP/LP methods are compared with two heuristic non-MILP approaches to solving sequential extensive-form SSGs (*O2UCT* and *EASG*, summarized in Sections 3.1 and 3.2, respectively) with adequate ATSG adjustments (Sections 3.1.1 and 3.2.1, resp.). Contrary to the MILP/LP methods, the heuristic approaches are capable of dealing with both the non-linear ATSG formulation and its linearized ATSGL version.

3.1. A summary of O2UCT *method*

The first approach (referred to as *O2UCT* — double-oracle UCT sampling) [24, 25] relies on the guided sampling of the follower's strategy space interleaved with finding a feasible leader's strategy using the double-oracle method.

In the first step, the follower's pure strategy (π_f^r) is obtained using the *Upper Confidence bounds applied to Trees* (UCT) algorithm [45] - a variant of guided Monte Carlo sampling. Then, for the sampled follower's strategy, a process of building the leader's strategy (δ_l) is performed. δ_l must satisfy the following conditions: (1) π_f^r is the optimal response strategy against δ_l; (2) δ_l provides the highest possible leader's utility against the best follower's response. An algorithm for finding the requested leader's strategy δ_l is presented in detail in [25] and outlined in Figure 2.

In the first step in Figure 2, the optimal follower's response (π_f^b) is calculated (†) against δ_l. Then, the algorithm checks if $\pi_f^b = \pi_f^r$. If so, then a procedure for adjusting δ_l to obtain better utility against π_f^b (compared with π_f^r) is applied

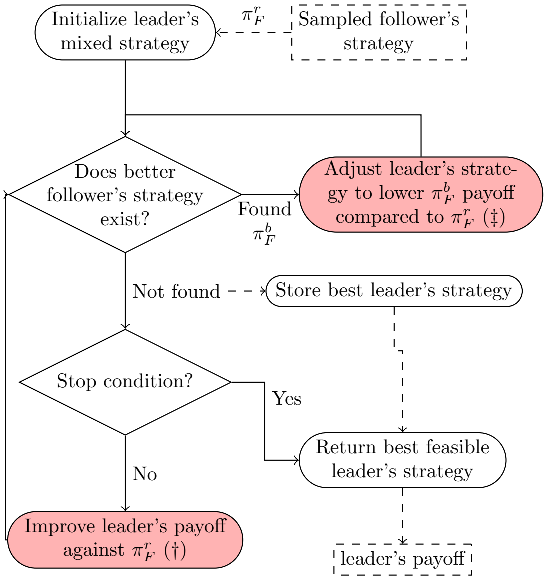

Initialize leader's mixed strategy
π_F^r
Sampled follower's strategy
Does better follower's strategy exist?
Found π_F^b
Adjust leader's strategy to lower π_F^b payoff compared to π_F^r (‡)
Not found
Store best leader's strategy
Stop condition?
Yes
Return best feasible leader's strategy
No
Improve leader's payoff against π_F^r (†)
leader's payoff

Figure 2: *O2UCT*: An overview of the method of finding the leader's mixed strategy corresponding to the requested follower's strategy. Procedures marked in red adjust the current leader's strategy.

(‡). Otherwise, when $\pi_f^r \neq \pi_f^b$, an adjustment to δ_l is made so as to increase the leader's utility against π_f^r.

In *O2UCT* the two above-mentioned phases: the sampling of the follower's strategy π_f^r (against the current leader's strategy δ_l) and the adjustment of δ_l are iteratively alternated until the stopping conditions are met. Consult [24, 25] for a detailed description of the method.

3.1.1. ATSG implementation

The implementation of ATSG in *O2UCT* requires two changes. First of all, in the follower's best response oracle (†), which exhaustively searches through all possible pure strategies in *O2UCT*, the procedure that calculates the follower's utility is modified so as to use distorted probabilities (17) when calculating the expected value. Similarly, in the procedure that calculates the difference between the follower's utility for two strategies (‡), the way the expected utility is calculated is adapted so as to use a distorted strategy (perceived by the follower in ATSG).

In the case of *O2UCT*, contrary to the MILP/LP ATSG implementations, the potential existence of non-linearities in the formulas defining the distorted follower's probabilities is not harmful, and – in principle – any other BR modification could be used instead of eq. (17).

3.2. A summary of EASG *method*

The other heuristic method applicable to sequential SGs considered in this paper utilizes Evolutionary Algorithms (EA) to find the leader's mixed strategy [26] and, to our knowledge, is the first generic evolutionary approach proposed in this domain. The authors are not aware of any other EA-based applications to sequential SGs except for our previous approach [48], which, however, is tailored to a specific game scenario in which targets are moving along predefined trajectories.

Algorithm 1: Pseudocode of *EASG*.

```
$\mathcal{P}$ - population
$\mathcal{P} \leftarrow$ randomly selected leader's pure strategies
while (generations limit not reached) do
    $E \leftarrow n_e$ chromosomes with highest fitness function values
    $\mathcal{P}_c \subseteq \mathcal{P}$ /* random population subset for crossover */
    /* crossover merges pairs of chromosomes */
    $\mathcal{P} = \mathcal{P} \cup Crossover(\mathcal{P}_c)$
    $\mathcal{P}_m \subseteq \mathcal{P}$ /* random population subset for mutation */
    /* mutation changes actions in a randomly selected element of a
       chromosome */
    $\mathcal{P} = (\mathcal{P} \setminus \mathcal{P}_m) \cup Mutation(\mathcal{P}_m)$
    $Evaluate(\mathcal{P})$ /* calculate fitness function value - leader's
       payoff against optimal follower's response to a strategy
       encoded in a chromosome */
    $\mathcal{P} = E \cup Selection(\mathcal{P})$ /* choose strategies for next generation
       based on fitness evaluation */
end
return best leader's strategy
```

EASG follows a standard evolutionary algorithm scheme as presented in Algorithm 1. A population of individuals evolves for a fixed number of generations. In each generation crossover and mutation operators are applied with certain probabilities and the population for the next generation is created by a selection procedure, based on the fitness value computed for each individual.

Population. Each chromosome $CH_q, q = 1, \ldots, population\_size$ represents some leader's mixed strategy in the form of a vector of pure strategies π_i^q with their probabilities p_i^q:

$$CH_q = \{(\pi_1^q, p_1^q), \ldots, (\pi_{l_q}^q, p_{l_q}^q)\}, \quad \sum_{i=1}^{l_q} p_i^q = 1, \tag{23}$$

where l_q is the length of CH_q. A strategy π_i^q is a list of the leader's actions in consecutive rounds. Each chromosome in the initial population includes one randomly selected pure strategy with a probability equal to 1.

Crossover. A crossover operator combines two randomly chosen chromosomes by aggregating all pure strategies they contain and halving their probabilities (if a given strategy belongs to both chromosomes, the resulting probabilities are summed up). Crossover is applied to a chosen pair of chromosomes with a specified probability.

Mutation. In the mutation operation a pair (pure strategy, round number) is uniformly selected in a chromosome. Then, starting from the selected round until the last one, a leader's action is uniformly chosen in each round (among all actions available in this round) to replace the existing action. The mutation affects each individual with a specified probability.

The role of the mutation operation is to boost the exploration of the leader's strategy space while crossover combines existing solutions and has a more exploitative nature.

Selection. Selection is a two-step procedure. First, n_e individuals with the highest fitness values (called the *elite*) from the union of the current population and the set of offspring chromosomes are directly (unconditionally) promoted to the next generation population. Next, a binary tournament with a specified selection pressure P is iteratively executed until the next generation is filled with $population\_size$ individuals, i.e. in each tournament among two randomly chosen chromosomes the higher-rated one (in terms of the fitness value) is promoted with probability P and the lower-rated one with probability $1 - P$. Chromosomes for the tournaments are sampled from the union of the current population and the offspring.

Evaluation. The fitness function is defined as the leader's utility obtained when playing a strategy encoded by a chromosome. This utility is calculated by computing game payoffs against all possible follower's strategies and choosing the one that yields the highest value for the follower while breaking ties in favor of the leader (the SSE condition).

3.2.1. ATSG implementation

Similarly to *O2UCT*, an important advantage of the *EASG* formulation is its flexibility, understood as the ease of adaptation to various SG formulations. In the context of BR, various types of perturbations to the optimal follower's response can be implemented in *EASG* by adjusting the chromosome evaluation procedure.

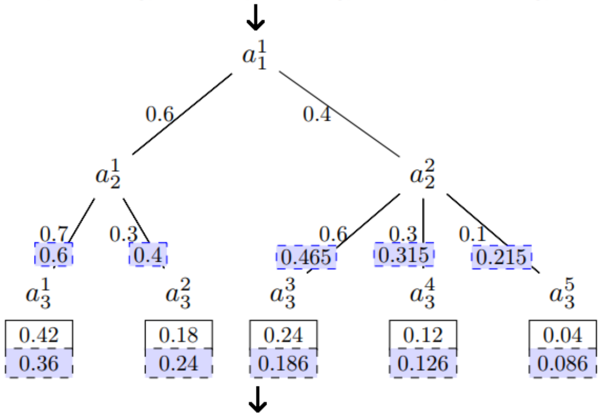
The incorporation of ATSG into *EASG* relies on considering a distorted version of the leader's mixed strategy when calculating the best follower's response. This distorted leader's strategy is obtained in the three following steps.

1. First, in order to directly apply eq. (17), a strategy encoded by a chromosome is transformed into a tree with nodes and edges representing game states and moves (actions) between states, respectively. Formally, if $\sigma_l = a_1, a_2, \ldots, a_l$ is a list of consecutive actions in the first l rounds and $P_{pref}(\sigma_l)$ is a sum of probabilities of all pure strategies in the chromosome that begin with the actions σ_l, then the probability of an edge (move) in a game tree between nodes corresponding to $\sigma_{l-1} = init(\sigma)$ and σ_l is computed as $\frac{P_{pref}(\sigma_l)}{P_{pref}(\sigma_{l-1})}$.
2. Next, all probabilities in the above game tree are modified in line with eq. (17).
3. Finally, the tree (with modified probabilities) is transformed back to a list of pure strategies with assigned probabilities through a reversed procedure, i.e. each unique path from the root to a leaf node corresponds to a pure strategy encoded by the list of the respective actions (nodes) with a probability equal to the product of probabilities on all edges on this path.

A set of all such pairs (encoded pure strategy and its probability) form a new chromosome (see eq. (23)), which represents the ATSG-distorted leader's strategy.

An example of how the leader's strategy encoded in a chromosome is transformed into its distorted version is presented in Figure 3.

$$CH = \{([a_1^1, a_2^1, a_3^1], 0.42), ([a_1^1, a_2^1, a_3^2], 0.18), \\ ([a_1^1, a_2^2, a_3^3], 0.24), ([a_1^1, a_2^2, a_3^4], 0.12), ([a_1^1, a_2^2, a_3^5], 0.04)\}$$



$$CH' = \{([a_1^1, a_2^1, a_3^1], 0.36), ([a_1^1, a_2^1, a_3^2], 0.24), \\ ([a_1^1, a_2^2, a_3^3], 0.186), ([a_1^1, a_2^2, a_3^4], 0.126), ([a_1^1, a_2^2, a_3^5], 0.086)\}$$

Figure 3: An example of how the ATSG-distorted leader's strategy is computed in the modified EASG method.

Once the ATSG leader's strategy is defined, the optimal follower's response strategy is computed by enumerating over all possible follower's pure strategies and choosing the one with the highest follower's payoff.

Next, this follower's response strategy is used to calculate the utility of the players, but this time using the original, unmodified strategy extracted from a chromosome (without the distortion of probabilities). In other words, a distorted strategy is used only in the calculation of the follower's utility. The leader is assumed to be perfectly rational and therefore his/her utility (chromosome fitness value) is calculated with no distortion.

Similarly to *O2UCT*, any other BR model could be used instead of eq. (17).

4. Experimental evaluation

Proposed ATSG/ATSGL modifications are evaluated from two perspectives:

- **time** – computation times of the same methods with and without AT an implementation are compared to see the influence of AT on computational efficiency,
- **payoffs** – several approximations of the basic ATSG formulation are considered: ATSGL implemented in *BC2015* and *C2016* (exact MILP methods), ATSGL implemented in *CBK-2018* (heuristic MILP method), and ATSG implemented in *EASG* and *O2UCT* (non-MILP metaheuristic methods). For each method, the performance of the obtained leader's strategy was tested against an ATSG follower (i.e. a follower with an AT perception distortion) and against a fully rational follower (with no AT distortion).

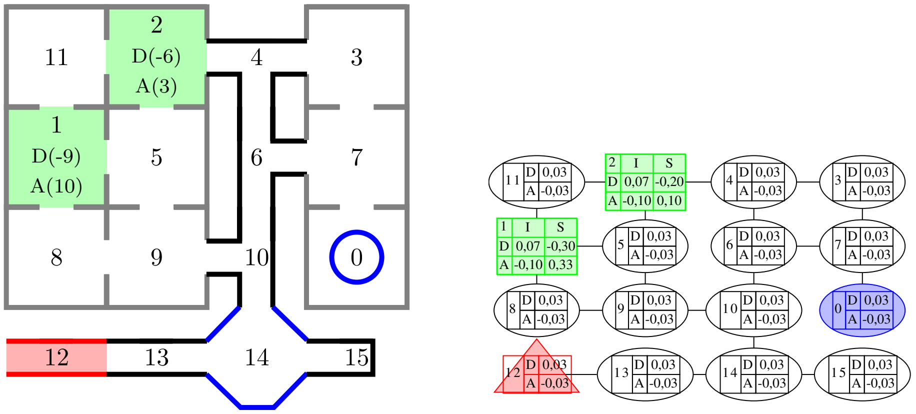
The ATSG versions of the five considered methods will be referred to with the prefix *AT-*:

- *AT-BC2015*, *AT-C2016*, *AT-CBK2018* – ATSGL implementations of the respective MILP methods,
- *AT-O2UCT* and *AT-EASG* – ATSG implementations of the respective metaheuristic methods.

4.1. Benchmark games

An experimental evaluation is performed on three sets of benchmark games: WHG [27], WNZ [25] and SEG [23]. All WHG and WNZ instances can be downloaded from our project website [28].

Each WHG/WNZ game refers to a scenario of patrolling a warehouse or an office building. The game area is modeled in the form of a graph with some vertices containing valuable resources (referred to as *targets*). Each player (leader, follower) possesses a single unit, located in one of the vertices (warehouse spaces). In each round, each unit can either stay in the currently occupied



(a) An example of a warehouse layout: the narrow black path denotes the main corridor, squares are storage spaces. Room numbers correspond to the vertex labels in the resulting game graph presented in the right figure.

(b) The corresponding game graph. Rectangular vertices are targets, the triangle vertex is the follower's starting point, the blue shaded circle vertex is the leader's starting point.

Figure 4: An example game from the Warehouse Games benchmark. Values in the right figure denote the payoffs for the follower and the leader, respectively, in the case of an interception of the follower in a given vertex. Additional utility values, relevant in the case of a successful attack, are assigned to targets (right column). All games are defined on a 4×4 grid.

vertex or move to an adjacent vertex (change the room). If the units meet in a common vertex, an interception occurs and the leader receives a reward while the follower receives a penalty. If the follower reaches any of the target vertices (rooms) without being intercepted by the leader, he/she is rewarded and the leader is penalized. In either of the above cases, the game ends. Otherwise, the game is played for a fixed number of rounds T. Once the round limit T is reached, both players are assigned a neutral utility of 0.

The WHG benchmark set consists of 25 game layouts generated on a 4×4 grid, with general-sum utility. An example game layout created by the warehouse generator is presented in Figure 4a (this is an auxiliary game representation). The corresponding game graph (the actual game representation) is depicted in Figure 4b. A detailed description of the game generator settings is presented in [27]. In this paper, each of the 25 WHG games is considered with $T = 3, \ldots, 7$, albeit for $T = 7$ exact methods were unable to compute solutions within the allotted time and memory. This leads to 125 test games, in total.

The WNZ instances admit exactly the same graph structure as the WHG

Table 1: Payoff ranges of the Warehouse Games generator for WHG and WNZ game instances. Actual values for the games were uniformly drawn from the interval $[x_{min}, x_{max}]$.

Parameter	WHG		WNZ	
	x_{min}	x_{max}	x_{min}	x_{max}
follower's penalty for being caught in a target	-0.1	-0.1	-1	0.2
leader's reward for catching the follower in a target	0.03	0.03	0.2	0.2
follower's penalty for being caught in a vertex other than a target	-0.03	-0.03	-1	0
leader's reward for catching the follower in a vertex other than a target	0.06	0.06	0.1	0.1
follower's reward in the case of a successful attack	0.03	0.67	-0.2	1
leader's penalty in the case of a successful attack	-0.67	-0.03	-1	0.2

ones, differing in the payoff structure, which is more diverse in WNZ. In effect, WNZ games are less correlated and "further away" from zero-sum ones than their WHG counterparts. The average Pearson's correlation coefficient (PCC) between the leader's and the follower's rewards in WHG and WNZ instances is equal to -0.82 and -0.57, respectively. For comparison, for zero-sum games $PCC = -1.00$. In total, 125 test games are considered – 25 for each value of $T = 3, \dots, 7$. The payoff ranges used in both settings of the game generator are presented in Table 1.

SEG instances are played on directed graphs and according to different rules. Game instances are defined on three variants of a graph proposed in [29] and presented in Figure 5. The graph variants differ in their connection topologies. The game graphs are directed and there is no possibility to retreat from some vertices.

The leader has 2 units and the mobility of each of them is restricted to one of the node subsets denoted by oval shapes in the figure. The follower has a single unit at his/her disposal. In each round, each unit can move to one of the adjacent vertices. Additionally, the follower can choose to stay in the currently occupied vertex.

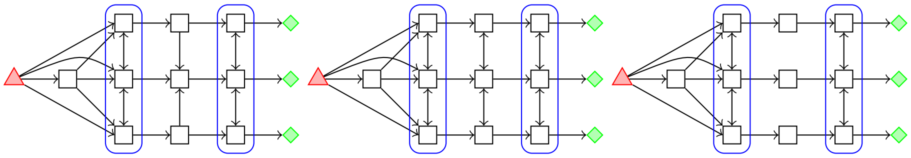


Figure 5: Game layouts of SEG instances.

Furthermore, when moving, the follower leaves traces which are visible to the leader when he/she enters a vertex in which a follower was previously present. The trace in a given node disappears (is erased by the follower) if the follower chooses to stay in this vertex for another round (time step).

The follower starts the game in the leftmost red triangle vertex of the graph and receives a reward on reaching one of the rightmost vertices. If the leader and the follower meet in a common vertex they both receive a payoff of 1 (leader) and -1 (follower). If the follower does not reach any of the destination nodes, the payoff for both of them equals 0. The follower's rewards in the destination (rightmost) nodes are drawn uniformly from the range $[1, 2]$. The leader's penalty in case the follower reaches a target equals -1. For each game graph two game variants are considered – in the first one the follower is able to erase traces and in the other one staying in a vertex (and erasing traces) is not allowed. For each game setting (one of the three graphs presented in Figure 5 and one of the two above-mentioned variants) 5 game instances with various follower's rewards were generated leading to 30 test games. Each of these games is played with a time limit of $T = 4, 5, 6$ steps. For smaller values of T it is impossible for the follower to reach the destination whereas larger games are too complex to be solved by MILP methods as discussed in Section 4.2.

4.2. Experimental setup

The performance of both heuristic methods was analyzed along two dimensions: the quality of the results (the expected leader's payoff) and time efficiency. Results for all games were grouped based on the order of magnitude of the number of game nodes in the extensive form of the game. Formally, the grouping follows the formula (24):

$$bucket = 10^{round(\log_{10}|\mathcal{H}|)}, \tag{24}$$

where *round* rounds a number to the nearest integer. Such a grouping combines two aspects of game complexity: one stemming from the underlying game graph structure and the other one resulting from the game length. For the remainder of this paper $B_i, i = 2, \dots, 7$ will denote the *i-th bucket of games*, i.e. the one which contains all games for which $round(\log_{10}|\mathcal{H}|) = i$. In order to streamline the notation, $B_{\geq i}, i = 2, \dots, 7$ will denote the *union of buckets* $B_i, B_{i+1}, \dots, B_7$ and $B_{\leq i}, i = 2, \dots, 7$, the *union of buckets* $B_2, B_3, \dots B_i$.

Tests were run on an Intel Xeon Silver 4116 @ 2.10GHz with 256GB RAM. Experiments with *O2UCT* and *EASG* were run in parallel, each with 8GB RAM assigned. Tests with *BC2015*, *C2016*, *CBK2018* were executed sequentially with all 256GB RAM available in each trial. Each run was limited to 200 hours and was forcibly terminated if not completed within the allotted time. The same settings were applied to the respective *AT*-modified versions.

Table 2 shows the number of calculated game instances for each of the methods in both fully-rational and AT settings. *O2UCT* and *EASG* were able to complete every game instance. For each game instance *(AT-)O2UCT* and *(AT-)EASG* were run 10 times and for each other method (deterministic MILP) a single trial was performed.

The *AT-BC2015* method is parameterless (this also holds for *BC2015* [23]). In *AT-C2016*, the SI-LP variant of *C2016* [21] is considered. For *AT-CBK2018*, the fast-converging variant of *CBK2018* [22] (with $\epsilon = 0.3$ and $\sigma = 0.4$) is implemented.

The parameters for metaheuristic methods are selected based on a limited number of preliminary simulations. *AT-O2UCT* is parameterized by the following 3 stopping conditions (cf. Figure 2): either the maximum number of executions of the *positive pass* (step † in the figure) exceeds $5,000$, or the improvement of the leader's payoff in 500 subsequent iterations is less than 10^{-5}, or the number of subsequent executions of the *feasibility pass* (step ‡ in the figure)

Table 2: Number of instances solved within time limit.

	WHG					WNZ					SEG		
	B_3	B_4	B_5	B_6	B_7	B_3	B_4	B_5	B_6	B_7	B_5	B_6	B_7
AT-O2UCT	26	25	24	25	23	25	23	21	24	22	30	30	30
AT-EASG	26	25	24	25	23	25	23	21	24	22	30	30	30
AT-BC2015	26	25	24	25	11	25	23	21	24	6	30	30	0
AT-C2016	26	25	24	25	0	25	23	20	22	0	30	30	0
AT-CBK2018	26	25	24	25	15	25	23	21	23	5	30	30	0
O2UCT	26	25	24	25	23	25	23	21	24	22	30	30	30
EASG	26	25	24	25	23	25	23	21	24	22	30	30	30
BC2015	26	25	24	25	3	25	23	21	24	7	30	30	0
C2016	26	25	24	25	2	25	23	20	18	0	30	30	0
CBK2018	26	25	24	25	21	25	23	21	23	18	30	30	0

without the execution of the *positive pass* (step †) exceeds $10,000$ (*infeasible strategy*).

In *AT-EASG* the following values for the steering parameters are selected: population size - 30, mutation probability - 0.5, crossover probability - 0.8, selection pressure $P = 0.9$, the number of elitist chromosomes - $n_e = 2$. The algorithm is run either for $1,000$ generations or until no improvement of the leader's strategy is observed in 20 subsequent generations (whichever occurs first).

In ATSG/ATSGL formulations (17), (18), $\alpha = 0.5$ was applied.

4.3. Payoffs

In order to evaluate the efficacy of the proposed AT implementation in SSGs (ATSG/ATSGL), the leader's payoffs obtained while playing the calculated strategy against an AT-biased follower are compared for the 5 tested methods. The average expected leader's utility obtained by each method for WHG, WNZ and SEG are presented in Figures 6a, 6b and 6c, respectively.

As reported in Table 2, some methods were not able to calculate solutions for larger game instances. Consequently, if for a given bucket fewer than 75% of the games were calculated by a given method, the respective data point was omitted from the plot. Otherwise, each data point presents a mean result across all game instances calculated by a given method.

Since both *AT-C2016* and *AT-BC2015* are exact methods, their results are the same and are presented as ATSGL in the plot. Additionally, the results of playing the Stackelberg Equilibrium leader's strategy (the variant with a perfectly-rational opponent) are presented as a baseline (no-AT). *AT-EASG* and *AT-O2UCT* are metaheuristic methods that implement a non-simplified ATSG model of the follower's behavior (17) and *AT-BC2015*, *AT-C2016*, *AT-CBK2018* rely on the ATSGL model (18).

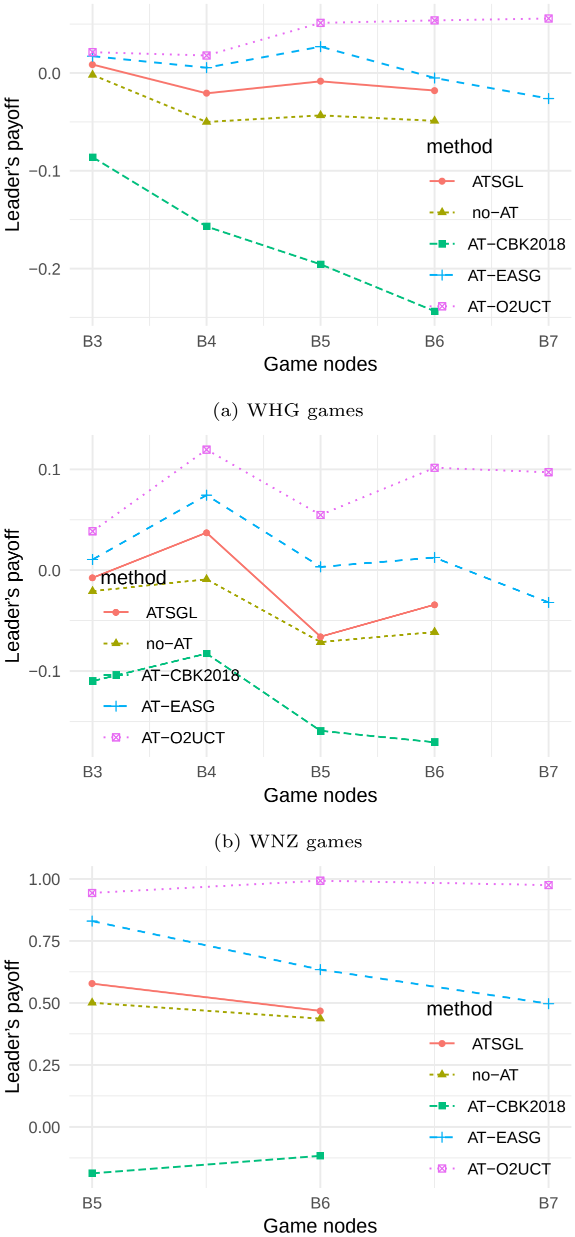
AT-CBK2018 performs poorly in all game genres, though it should be pointed out that this method yields weak results for these games also in perfect rationality settings [25]. The second worst result in all three figures is that of no-AT, which is not surprising since no-AT does not consider the assumed opponent's AT bias in any way. *AT-O2UCT* is the best among all methods, followed by *AT-EASG*. The MILP solutions (*AT-BC2015*, *AT-C2016*) that use the simplified follower's model (ATSGL) provide lower payoffs, though still higher than the no-AT baseline.

Particular differences between methods depend on the game set. In the case of more demanding game genres: WNZ and SEG, the relative gap (compared with the baseline) between ATSGL and *AT-O2UCT* or *AT-EASG* is bigger than in the case of relatively simpler WHG instances.

In all three game sets the gap between *AT-O2UCT* and *AT-EASG* grows for larger games, though at the cost of increased computation time for *AT-O2UCT*, as discussed in the next section.

4.4. Time scalability

The time analysis of the compared methods needs to be done with care since the time required to compute a strategy depends on both the method itself (which is irrelevant to the AT modifications) and the changes introduced by the follower's AT bias implementation. For this reason, the running times of the original methods (with a fully-rational follower) on the three benchmark sets are presented first, so as to establish the model-specific baselines. Next, the results for the AT-modified methods are presented and discussed with respect



(a) WHG games

(b) WNZ games

(c) SEG games

Figure 6: The average expected leader's utility obtained by each of the tested methods. Since both *AT-C2016* and *AT-BC2015* are exact methods, their results are the same and are presented as ATSGL in the plot. no-AT results refer to the case of playing the Stackelberg Equilibrium leader's strategy (the variant with a perfectly-rational opponent) - a baseline result.

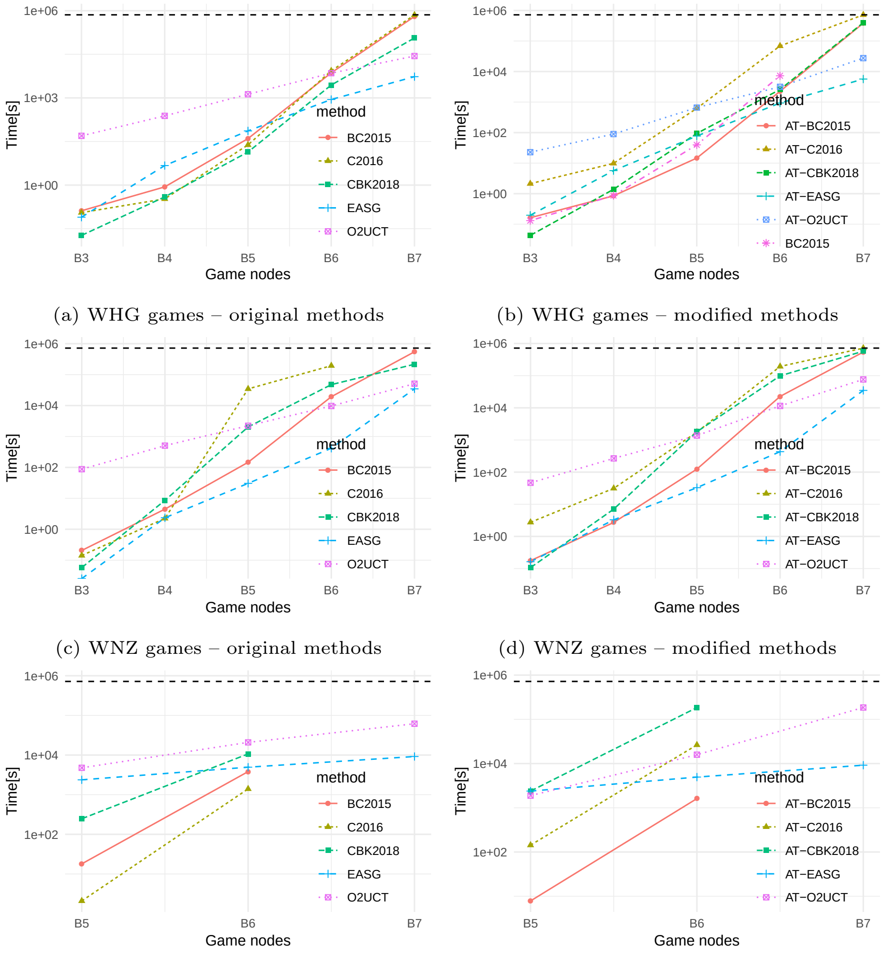
to the baseline. As stated earlier, all experiments were executed with a time limit of 200 hours, after which each trial was forcibly terminated. The dashed line in the plots denotes the time limit. For every trial that exceeds the time limit, a threshold value of 200 hours is used as the computation time estimate, which makes the average computation times for such cases smaller than the actual values (without time limits imposed).

Figures 7a, 7c and 7e present the average running times of unmodified methods for the respective benchmark sets.

EASG is the fastest method except for some small games. *O2UCT* is the slowest when applied to small games but starts to outperform *BC2105*, *C2016* and *CBK2018* for larger instances. MILP-based methods suffer from hitting the time limit for some games in B_6 and the vast majority of B_7 instances for WNZ, for all games in B_7 for SEG, and for about half of B_7 WHG instances. Moreover, initial trials (not presented) have shown that for larger games (B_7 and beyond) MILP methods require more than 256GB of RAM while *EASG* and *O2UCT* still work fine with an 8GB memory limit.

The exact differences between the methods depend on the characteristics of the particular game set. For WNZ, which has a more complicated payoff structure, *O2UCT* starts to outperform the other methods earlier, while for WHG, where the payoff structure is closer to zero-sum, MILP methods are still reasonably fast for B_6. In the case of SEG, which uses fixed game graph, and therefore less diverse extensive-form game trees, only two data buckets for MILP-based methods could be gathered, which made a more detailed comparison impossible. The general conclusion is that MILP approaches are usable only with smaller games, while metaheuristic methods perform better on larger instances. This comes at the cost of the metaheuristic methods not guaranteeing convergence to the optimal solution.

Figures 7b and 7d and 7f present computation times for AT-modified methods for the respective game sets. The times presented in the plots follow the trends identified for the baseline. Generally, *AT-EASG* is still the fastest method, although for smaller games from WHG, it is outperformed by *BC2015*



(a) WHG games – original methods

(b) WHG games – modified methods

(c) WNZ games – original methods

(d) WNZ games – modified methods

(e) SEG games – original methods

(f) SEG games – modified methods

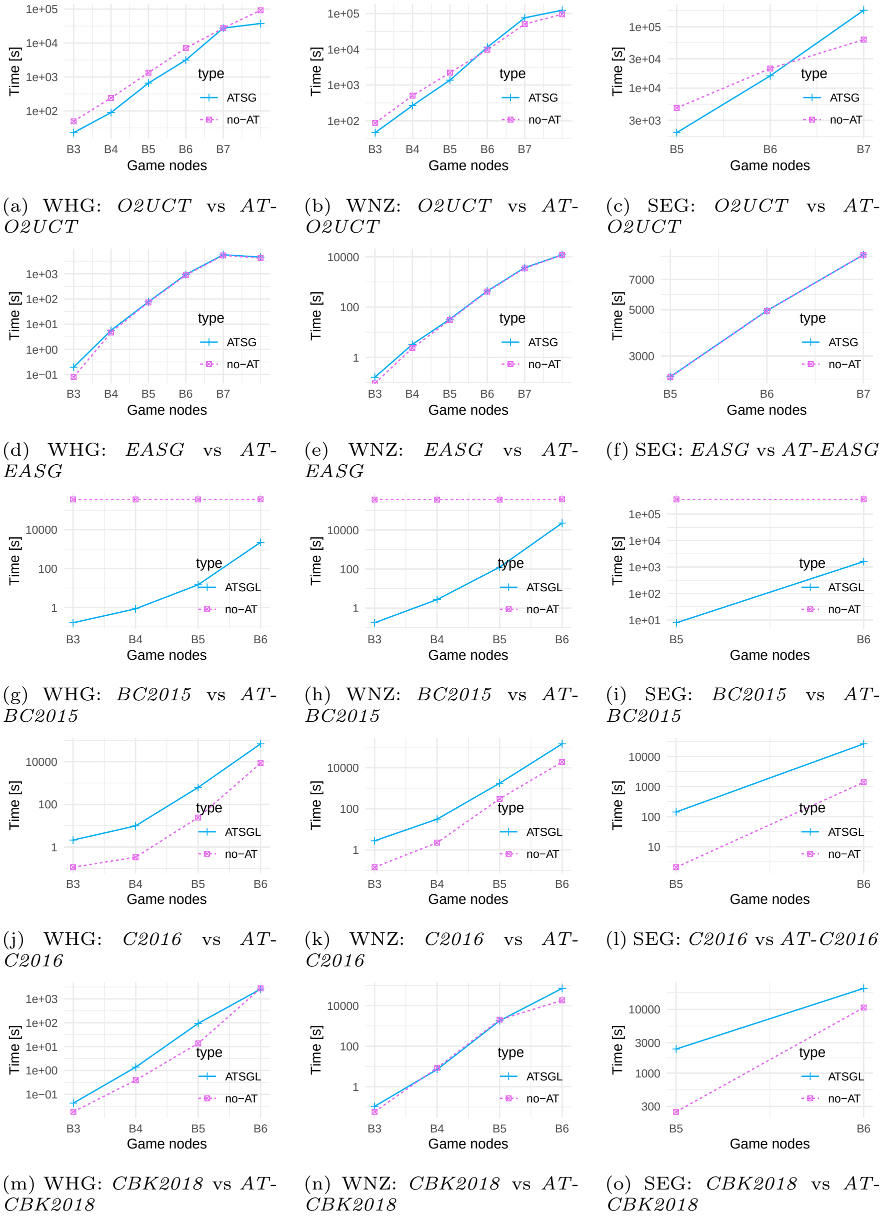
Figure 7: On the left: average computation times for original methods without an AT implementation - a baseline. On the right: average computation times for the ATSG/ATSGL-modified methods.

and *CBK2018*. *O2UCT* again is the slowest approach for small games but scales better with respect to the game's size. The difference that stands out the most is that *AT-C2016* is significantly slower than *C2016*. *AT-CBK2018*, which internally uses *AT-C2016*, gives mixed results. For SEG it is slower than the unmodified variant. For other games the times are very similar.

The differences between the remaining methods are not clearly visible in the general plot. Therefore, for each method, a comparison of the computation times before and after the introduction of the AT modification is provided. This way insights about the impact of the AT implementation on computation times of the respective methods can be gleaned. Figure 8 presents a comparison of computation times for the not-modified and AT-modified versions of the respective methods, for the analyzed game sets. In the case of *O2UCT*, for simpler games (the WHG set, which has a simpler payoff structure and smaller games than the other sets), the AT-modified version is slightly faster, but for more complicated games it scales worse and results in longer computation times. No significant differences in running time can be observed for *EASG*. *BC2015* in Figures 8g and 8i is slightly faster in the ATSGL version. For *C2016*, the AT implementation has the largest impact among all tested methods. The ATSGL implementation causes a major slowdown, making the method more than 10 times slower than the original one. *CBK2108*, which internally uses *C2016*, is slower for SEG (Figure 8o).

In summary, it seems reasonable to conclude that beyond a certain level of game complexity exact methods become infeasible and in such cases both metaheuristic approaches present a viable alternative. On the other hand, it should not be forgotten that the main advantage of *AT-C2016* and *AT-BC2015* is their convergence to the optimal solution for the ATSGL game formulation.

Based on a direct comparison, it can be concluded that for the set of most complex games *AT-EASG* and *AT-O2UCT* are clearly faster than the state-of-the-art heuristic MILP methods for solving extensive-form games. Furthermore, thanks to their capability to handle non-simplified AT formulations, they provide much better leader's payoffs. Among the two metaheuristic approaches,



(a) WHG: *O2UCT* vs *AT-O2UCT*

(b) WNZ: *O2UCT* vs *AT-O2UCT*

(c) SEG: *O2UCT* vs *AT-O2UCT*

(d) WHG: *EASG* vs *AT-EASG*

(e) WNZ: *EASG* vs *AT-EASG*

(f) SEG: *EASG* vs *AT-EASG*

(g) WHG: *BC2015* vs *AT-BC2015*

(h) WNZ: *BC2015* vs *AT-BC2015*

(i) SEG: *BC2015* vs *AT-BC2015*

(j) WHG: *C2016* vs *AT-C2016*

(k) WNZ: *C2016* vs *AT-C2016*

(l) SEG: *C2016* vs *AT-C2016*

(m) WHG: *CBK2018* vs *AT-CBK2018*

(n) WNZ: *CBK2018* vs *AT-CBK2018*

(o) SEG: *CBK2018* vs *AT-CBK2018*

Figure 8: Comparison of computation times of the original method and its ATSG/ATSGL version for each game set.

AT-O2UCT generally yields better payoffs for more complex games than *AT-EASG*, albeit at the cost of inferior time scalability.

5. Experiments involving human players

The underlying assumption of AT is that humans' cognitive biases cause them to perceive a slightly different probability distribution of possible action than the actual (true) distribution. This distortion may in turn lead them to take non-optimal actions. The idea behind considering AT in SG is to modify the leader's strategy by taking into account the (human) follower's biases as postulated by AT. Consequently, the AT-adjusted leader's strategy exploits this non-perfectly-rational behavior of the follower, which results in potentially better outcomes when the opponent is, in fact, a human. Hence, in order to make a comprehensive assessment of the proposed AT formulation, it is crucial to perform experiments not only against other methods but also against humans. To this end, the authors have developed a game portal [28] and tested the two proposed AT formulations: ATSG (17) and ATSGL (18) by playing SSGs against humans who took the role of the follower.

5.1. Experimental setup

A group of volunteers (Math. and CS students at the Warsaw University of Technology in Poland) were asked to play the role of the follower against the system-generated leader's strategies. The leader's strategy in a given gameplay either accounted for or ignored the potential AT bias of the follower. This information was not revealed to the human player.

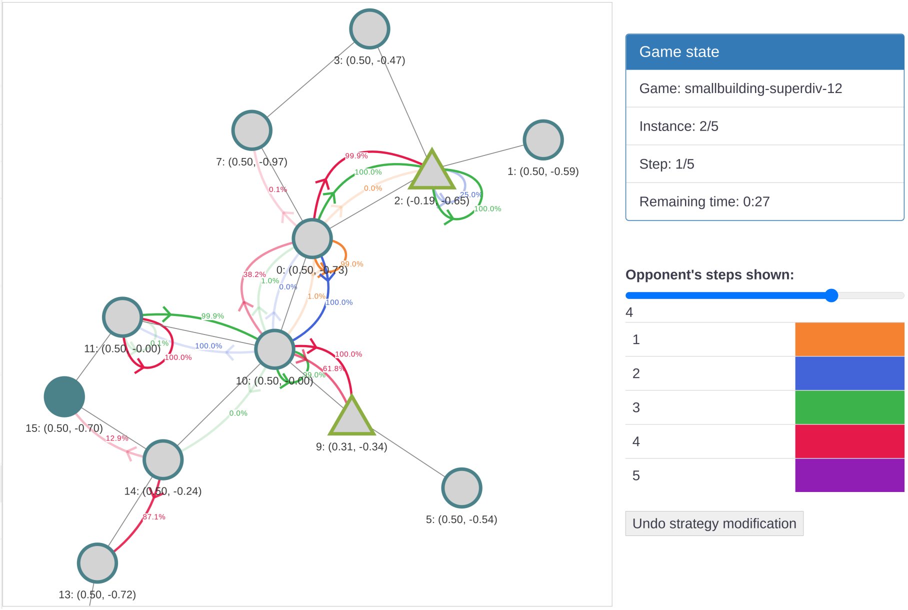
Game instances were a subset of 5-steps WNZ games played according to the rules defined in Section 4.1. Before each game, the user had 15 minutes to familiarize themselves with the opponent's (leader's) strategy. As per Stackelberg Game rules, the human participant did not know the exact sequence of the leader's moves planned to be played. He/she could only estimate the position of the leader in subsequent time steps based on the presented probabilities (the leader's mixed strategy). This strategy (with or without considering the AT bias) was precomputed by the algorithms described in Section 2.3. The game was presented in the form of a directed multigraph with the weights of the edges representing the probabilities of given moves in subsequent time steps.

The strategy committed to by the leader was visualized using colored arrows (see Figure 9) with the probabilities of the leader's actions/moves in subsequent time steps. Colors indicate possible moves in subsequent time steps according to the legend presented in the right part of the figure. The horizon of the currently visible strategy (in time steps) could be adjusted by the participant using a slider. Selecting ("clicking on") any of the arrows representing a single move in a given leader's mixed strategy hid all the pure strategies except those that included the indicated ("clicked") move.

In each game visualization presented to the experiment participant the position of his/her unit was denoted by a dark shaded circle and the targets were marked as green triangles (cf. Figure 9). The game was played for a fixed number of steps. If the player reached any of the targets within the time step limit and was not caught there, his/her result was equal to the first value in the brackets associated with this target. If the player was caught by the opponent, he/she received a penalty score equal to the second value in the brackets corresponding to the vertex in which the interception took place. Otherwise, the player received a payoff of 0.

Each participant played a given game instance 5 times in a row and then switched to the next one. In these 5 consecutive games, the distribution of payoffs was fixed. The human player could either differentiate his/her strategy or stick to a particular one, although *the realization of the leader's mixed strategy could also vary*, i.e. the leader could behave differently (another realization of his/her mixed strategy could have materialized) even though the human player repeated the same strategy.

The time for each action selection (moving to one of the adjacent vertices or staying in the occupied one) was set to 90 seconds. If no move was committed within the allotted time, the user's unit remained in the same vertex. There were three possible game endings: the user's unit was caught by the leader, the unit reached the target without being caught, or the step limit was reached. After the completion of the game, the payoff was presented to the user and he/she was offered to play again.



Game state

Game: smallbuilding-superdiv-12

Instance: 2/5

Step: 1/5

Remaining time: 0:27

Opponent's steps shown:

4

1	🟧
2	🟦
3	🟩
4	🟥
5	🟪

Figure 9: Example of the leader's strategy presentation in the web-based game portal [28]. Green triangles denote targets. The numbers below the nodes are the player's payoffs in the case of catching the follower in a given node. Arrows in different colors indicate the possible moves in subsequent time steps, as denoted by the legend presented in the right part of the figure, with probabilities of the leader selecting a given move in subsequent time steps. The horizon of the currently visible strategy (time steps) could be adjusted by the participant using a slider.

Each participant decided how many rounds of gameplay he/she wished to take. Some of them chose to play only one game, others played more than 80 games. Each time a user entered a new game, one of the three following leader's strategies was randomly chosen and presented to him/her:

- *no-AT* - the Stackelberg Equilibrium strategy for the leader was computed with no bounded rationality perturbations;
- *ATSG* - the leader's strategy was computed with a straightforward Anchoring Theory formulation (17). This form of an AT implementation is feasible only for the two metaheuristic approaches and is not feasible for

MILP methods;

- *ATSGL* - the leader's strategy was computed according to a modified ATSG formulation (18), feasible for both MILP and metaheuristic methods.

Players were neither provided with game theory knowledge related to the SE definition nor were they aware of the above-mentioned three different types of leader's strategy. Moreover, they were not informed about the experiment's purpose and were only provided with a brief tutorial describing the game rules and game platform navigation. Users had access to a brief technical tutorial describing the rules of the games they played, the way the games were visualized, and the way the leader's strategy was presented (cf. Figure 9). In order to increase the users' engagement, the platform displayed leaderboards presenting the players' results and encouraging them to take more attempts to move up in the rankings.

5.2. Results

Out of the 25 game instances, the ones with at least 5 playthroughs recorded for each of the three leader strategy derivation methods were selected. This way 16 games and 1056 playthroughs were evaluated.

Table 3 presents the average leader's payoffs of the 3 methods of leader strategy calculation for each of these 16 games. The last row presents the average ranking positions of these methods. For each game the possible payoffs came from the interval $[-1; 0.2]$ and a neutral value (relevant when the game ended with no interception and without reaching the target) was set to 0. Best results were obtained by a pure *ATSG* strategy with an average payoff of -0.106 and an average ranking position equal to 1.75. A close runner-up was the *ATSGL* strategy with the respective scores equal to -0.110 and 1.81. The *no-AT* strategy (without AT modifications) performed visibly worse in the majority of the games, with an average payoff over 20% lower. The differences are not statistically significant with p-values for the 1-tailed paired t-test equal to 0.187 (*no-AT* vs. *ATSG*), 0.176 (*no-AT* vs. *ATSGL*), and 0.387 (*ATSG* vs. *ATSGL*).

Table 3: Average leader's payoffs with average ranks of the three methods of leader strategy calculation. Best results are presented in bold.

game id	*no-AT*	*ATSG*	*ATSGL*
superdiv-02	-0.051	**0.000**	**0.000**
superdiv-09	-0.176	**−0.100**	-0.121
superdiv-16	-0.093	**−0.074**	-0.123
superdiv-17	-0.222	-0.159	**−0.132**
superdiv-28	**0.000**	-0.003	-0.010
superdiv-30	0.000	**0.028**	-0.151
superdiv-42	0.137	0.092	**0.190**
superdiv-43	0.086	**0.088**	0.021
superdiv-46	**0.000**	**0.000**	**0.000**
superdiv-51	-0.229	-0.136	**−0.114**
superdiv-56	-0.305	-0.113	**−0.009**
superdiv-59	-0.220	-0.311	**−0.108**
superdiv-68	**0.090**	0.059	-0.233
superdiv-70	0.000	**0.010**	0.000
superdiv-86	-0.282	-0.106	**0.000**
superdiv-95	**−0.924**	-0.968	-0.963
Avg payoff	-0.137	**−0.106**	-0.110
Avg rank position	2.13	**1.75**	1.81

Generally, the results indicate that both pure ATSG and its approximation ATSGL proposed in this paper present a viable alternative to the *fully-rational* Stackelberg Equilibrium approach in real-life scenarios in which humans play the role of the follower. In such cases an assumption about the bounded rationality of the follower underpins the two AT implementations, leading to higher results for the leader, whose strategy exploits the human-specific bias.

Figure 10 presents the number of playthroughs vs. categorized average payoffs for the 3 methods of leader strategy derivation. The main difference between the ATSG/ATSGL strategies and *no-AT* can be observed within the range of positive payoffs. The leader's payoff for catching the follower in the target / in any non-target vertex was equal to 0.2 / 0.1, respectively. Apparently the ATSG/ATSGL strategies distinctly more frequently led to catching the follower (directed by a human) in the targets than in the case of employing the *no-AT* strategy, which tended to catch the follower in non-target vertices. A possible

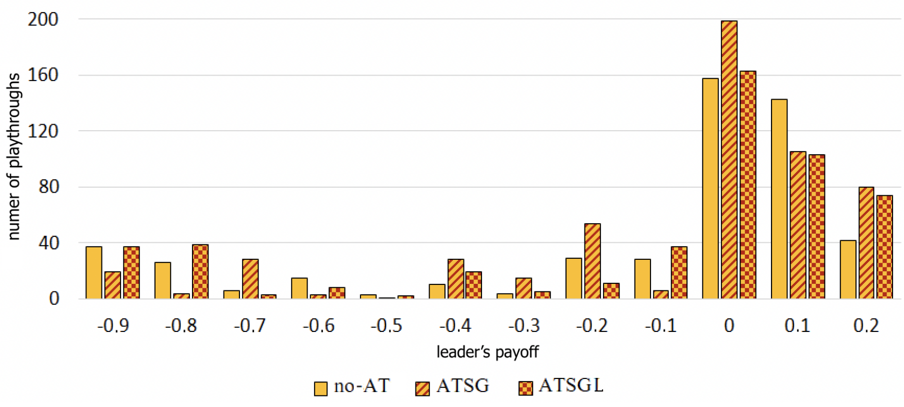


Figure 10: Number of playthroughs vs. categorized average payoffs for 3 methods of leader strategy calculation.

explanation of this phenomenon is that the follower, due to his/her bounded rationality, perceives the probability of being caught in a target as smaller than it really is, so he/she has a tendency to take higher risk and more often attempts to reach one of the targets than in the case of an undistorted probability perception. ATSG/ATSGL strategies exploit this biased perception of probabilities and adjust the leader's strategy accordingly, catching the opponent in the targets more frequently and yielding better payoffs.

6. Conclusions

This work considers an SG formulation in which the follower is not perfectly rational. Such a setting is motivated by real SG scenarios, in which humans performing the role of the follower are prone to certain inefficiencies in their perception and/or assessment of the leader's strategy. A particular implementation of the follower's bounded rationality considered in the paper refers to Anchoring Theory [18]. AT assumes the existence of a certain distortion (towards the uniform distribution of probabilities of possible actions) of the follower's perception of the leader's mixed strategy. The leader, being aware of this distortion, can exploit this weakness in their strategy formulation.

This paper proposes two efficient formulations of AT in the context of sequential extensive-form SGs. The first one (ATSG) is a straightforward extension of SG following directly from the AT definition. The other one (ATSGL) is a simplified version of ATSG with linear constraints, suitable for MILP methods. In the paper, ATSG is implemented for two metaheuristic approaches: *O2UCT* [24, 25] and *EASG* [26]. ATSGL, in turn, is implemented for three state-of-the-art MILP methods – two exact ones: *BC2015* [23] and *C2016* [21], and one approximate: *CBK2018* [22].

Experimental results on three sets of games show that non-MILP methods capable of using the non-simplified ATSG formulation outperform MILP methods in terms of the leader's payoff when playing against an AT-biased follower. Furthermore, they scale better for larger games as far as time is concerned.

The flexibility of non-MILP solutions is an additional advantage in the context of BR. This flexibility stems from virtually no restrictions being imposed on the form of the BR representation. Unlike MILP methods, which require a linear form of BR-related constraints, non-MILP solutions are suitable for the implementation of other, more complex BR models.

The efficacy of the ATSG implementation was additionally verified in experiments involving humans in the role of the follower. The results show that both AT implementations outperform the *fully-rational* Stackelberg Equilibrium approach. Although this advantage is not statistically significant (due to the limited number of human-involving experiments), the behavioral differences between ATSG/ATSGL and no-AT cases, which are presented in Figure 10 and stem from the AT-biased follower's perception, support the claim about the usefulness of considering the AT bias in real-life contexts, in which SSGs are employed to predict the behavior of human followers (terrorists, poachers, thieves, etc.).

The advantage of using the *ATSG* model in Sequential Stackelberg Games is the higher leader's average payoff compared with the baseline case which assumes that the follower is perfectly rational. A direct reference to the psychologically-grounded concept of Anchoring Theory (related to the human decision-making process) is an additional asset of the *ATSG* application. At the same time, the non-linear nature of the *ATSG* distortion hinders its direct implementation for the MILP/LP methods of solving Stackelberg Games, and for this reason, an approximate linear *ATSGL* version of *ATSG* is also proposed in this paper.

It should be noted, however, that *AT*, while very popular, is not the only BR model suggested in psychological research and, to the best of our knowledge, there is no consensus on a unique *BR* model that best approximates the human decision-making process.

Acknowledgment

This project was funded by the POB Research Centre Cybersecurity and Data Science of Warsaw University of Technology within the Excellence Initiative Program - Research University (ID-UB). The authors would like to thank Maciej Żelaszczyk for his careful reading and correction of the final version of the manuscript.

References

[1] G. Leitmann, On generalized Stackelberg strategies, Journal of Optimization Theory and Applications 26 (4) (1978) 637–643.

[2] H. von Stackelberg, Marktform und Gleichgewicht, Springer, Vienna, 1934.

[3] A. Sinha, F. Fang, B. An, C. Kiekintveld, M. Tambe, Stackelberg security games: Looking beyond a decade of success, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 5494–5501.

[4] A. Mahmoodi, Stackelberg–Nash equilibrium of pricing and inventory decisions in duopoly supply chains using a nested evolutionary algorithm, Applied Soft Computing 86 (2020) 105922.

[5] M. Pakseresht, I. Mahdavi, B. Shirazi, N. Mahdavi-Amiri, Co-reconfiguration of product family and supply chain using leader–follower

Stackelberg game theory: Bi-level multi-objective optimization, Applied Soft Computing 91 (2020) 106203.

[6] F. Ye, Y. Li, A Stackelberg single-period supply chain inventory model with weighted possibilistic mean values under fuzzy environment, Applied Soft Computing 11 (8) (2011) 5519–5527.

[7] F. Fang, P. Stone, M. Tambe, When Security Games go green: Designing defender strategies to prevent poaching and illegal fishing, in: Proceedings of the 24th IJCAI Conference, 2015, pp. 2589–2595.

[8] M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, F. Ordóñez, Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service, Interfaces 40 (4) (2010) 267–290.

[9] Z. Yin, A. X. Jiang, M. P. Johnson, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, M. Tambe, J. P. Sullivan, TRUSTS: Scheduling randomized patrols for fare inspection in transit systems, in: Twenty-Fourth IAAI Conference, 2012, pp. 2348–2355.

[10] E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, G. Meyer, PROTECT: A deployed game theoretic system to protect the ports of the United States, in: Proceedings of the 11th AAMAS Conference, 2012, pp. 13–20.

[11] J. B. Clempner, A. S. Poznyak, Conforming coalitions in Markov Stackelberg security games: Setting max cooperative defenders vs. non-cooperative attackers, Applied Soft Computing 47 (2016) 1–11.

[12] R. Yang, C. Kiekintveld, F. Ordóñez, M. Tambe, R. John, Improving resource allocation strategies against human adversaries in security games: An extended study, Artificial Intelligence 195 (195) (2013) 440–469.

[13] J. Pita, M. Jain, F. Ordóñez, M. Tambe, S. Kraus, R. Magori-Cohen, M. Tambe, Effective solutions for real-world Stackelberg games: When

agents must deal with human uncertainties, Security and Game Theory (2011) 193–212.

[14] H. A. Simon, Models of man: social and rational, Wiley, 1957.

[15] R. J. Aumann, Rationality and bounded rationality, Games and Economic behavior 21 (1-2) (1997) 2–14.

[16] A. Rubinstein, Modeling bounded rationality, MIT press, 1998.

[17] D. Kahneman, A. Tversky, Prospect theory: An analysis of decision under risk, in: Handbook of the fundamentals of financial decision making: Part I, World Scientific, 2013, pp. 99–127.

[18] A. Tversky, D. Kahneman, Judgment under uncertainty: Heuristics and biases, Science 185 (4157) (1974) 1124–1131.

[19] R. D. McKelvey, T. R. Palfrey, Quantal response equilibria for normal form games, Games and economic behavior 10 (1) (1995) 6–38.

[20] A. Tversky, D. Kahneman, The framing of decisions and the psychology of choice, Science 211 (4481) (1981) 453–458.

[21] J. Cermak, B. Bosansky, K. Durkota, V. Lisy, C. Kiekintveld, Using correlated strategies for computing Stackelberg equilibria in extensive-form games, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 439–445.

[22] J. Cerny, B. Bosansky, C. Kiekintveld, Incremental strategy generation for Stackelberg equilibria in extensive-form games, in: Proceedings of the 2018 ACM Conference on Economics and Computation, 2018, pp. 151–168.

[23] B. Bosansky, J. Cermak, Sequence-form algorithm for computing Stackelberg equilibria in extensive-form games, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI Press, 2015, pp. 805–811.

[24] J. Karwowski, J. Mańdziuk, Stackelberg equilibrium approximation in general-sum extensive-form games with double-oracle sampling method, in: Proceedings of the 18th AAMAS Conference, 2019, pp. 2045–2047.

[25] J. Karwowski, J. Mańdziuk, Double-oracle sampling method for stackelberg equilibrium approximation in general-sum extensive-form games, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 2054–2061.

[26] A. Żychowski, J. Mańdziuk, Evolution of Strategies in Sequential Security Games, in: Proceedings of the 20th AAMAS Conference, 2021, pp. 1434–1442.

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

[28] J. Mańdziuk, J. Karwowski, A. Żychowski, Simulation-based methods in multi-step Stackelberg Security Games in the context of homeland security (2022).
URL `https://sg.mini.pw.edu.pl`

[29] B. Bošanský, C. Kiekintveld, V. Lisý, M. Pěchouček, An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information, Journal of Artificial Intelligence Research 51 (2014) 829–866.

[30] J. Karwowski, J. Mańdziuk, A. Żychowski, Anchoring theory in sequential Stackelberg games, in: Proceedings of the 19th AAMAS Conference, 2020, pp. 1881–1883.

[31] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, S. Kraus, Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games, in: Proceedings of the 7th AAMAS Conference, 2008, pp. 895–902.

[32] D. Kar, F. Fang, F. D. Fave, N. Sintov, M. Tambe, A Game of Thrones : When Human Behavior Models Compete in Repeated Stackelberg Security Games, Proceedings of the 14th AAMAS Conference (2015) 1381–1390.

[33] J. Pita, R. John, R. Maheswaran, M. Tambe, R. Yang, S. Kraus, A robust approach to addressing human adversaries in security games, in: Proceedings of the 11th AAMAS Conference, 2012, pp. 1297–1298.

[34] R. Yang, C. Kiekintveld, F. Ordonez, M. Tambe, R. John, Improving resource allocation strategy against human adversaries in security games, in: Proceedings of the 22nd IJCAI Conference, 2011, pp. 458–464.

[35] T. H. Nguyen, R. Yang, A. Azaria, S. Kraus, M. Tambe, Analyzing the effectiveness of adversary modeling in security games, in: Proceedings of the 27th AAAI Conference on Artificial Intelligence, 2013, pp. 718–724.

[36] C.-W. Shao, Y.-F. Li, Optimal defense resources allocation for power system based on bounded rationality game theory analysis, IEEE Transactions on Power Systems 36 (5) (2021) 4223–4234.

[37] S. Cooney, P. Vayanos, T. H. Nguyen, C. Gonzalez, C. Lebiere, E. A. Cranford, M. Tambe, Warning time: optimizing strategic signaling for security against boundedly rational adversaries, in: Proceedings of the 18th AAMAS Conference, 2019, pp. 1892–1894.

[38] S. Gholami, S. Mc Carthy, B. Dilkina, A. J. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, M. Nsubaga, J. Mabonga, et al., Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers., in: Proceedings of the 17th AAMAS Conference, 2018, pp. 823–831.

[39] O. Thakoor, S. Jabbari, P. Aggarwal, C. Gonzalez, M. Tambe, P. Vayanos, Exploiting bounded rationality in risk-based cyber camouflage games, in: International Conference on Decision and Game Theory for Security, Springer, 2020, pp. 103–124.

[40] O. Thakoor, M. Tambe, P. Vayanos, H. Xu, C. Kiekintveld, F. Fang, Cyber camouflage games for strategic deception, in: International Conference on Decision and Game Theory for Security, Springer, 2019, pp. 525–541.

[41] A. Venugopal, E. Bondi, H. Kamarthi, K. Dholakia, B. Ravindran, M. Tambe, Reinforcement learning for unified allocation and patrolling in signaling games with uncertainty, in: Proceedings of the 20th AAMAS Conference, 2021, pp. 1353–1361.

[42] K. Wang, A. Perrault, A. Mate, M. Tambe, Scalable game-focused learning of adversary models: Data-to-decisions in network security games., in: Proceedings of the 19th AAMAS Conference, 2020, pp. 1449–1457.

[43] A. Perrault, B. Wilder, E. Ewing, A. Mate, B. Dilkina, M. Tambe, End-to-end game-focused learning of adversary behavior in security games., in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 1378–1386.

[44] H. Xu, B. Ford, F. Fang, B. Dilkina, A. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, M. Nsubaga, J. Mabonga, Optimal patrol planning for green security games with black-box attackers, Decision and Game Theory for Security (2017) 458–477.

[45] L. Kocsis, C. Szepesvári, Bandit based Monte-Carlo planning, in: Machine Learning: ECML 2006, Springer, Cham, Switzerland, 2006, pp. 282–293.

[46] J. Karwowski, J. Mańdziuk, A new approach to Security Games, in: Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICAISC'2015), Vol. 9120 of Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 402–411.

[47] J. Karwowski, J. Mańdziuk, Mixed strategy extraction from UCT tree in security games, in: Proceedings of the European Conference on Artificial Intelligence (ECAI'2016), 2016, pp. 1746–1747.

[48] J. Karwowski, J. Mańdziuk, A. Żychowski, F. Grajek, B. An, A Memetic Approach for Sequential Security Games on a Plane with Moving Targets, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 970–977.

[49] A. Żychowski, J. Mańdziuk, E. Bondi, A. Venugopal, M. Tambe, B. Ravindran, Evolutionary approach to Security Games with signaling, Proceedings of the 31st IJCAI Conference (2022) 620–627.

[50] J. Cerny, B. Bosansky, B. An, Finite state machines play extensive-form games, in: The 21st ACM Conference on Economics and Computation, 2020, pp. 509–533.

[51] C. R. Fox, Y. Rottenstreich, Partition priming in judgment under uncertainty, Psychological Science 14 (3) (2003) 195–200.

[52] D. Lawrence, T.-L. Davies, R. Bagshaw, P. Hewlett, P. Taylor, A. Watt, External validity and anchoring heuristics: application of DUNDRUM-1 to secure service gatekeeping in South Wales, BJPsych bulletin 42 (1) (2018) 10–18.

[53] M. Morrin, J. J. Inman, S. M. Broniarczyk, G. Y. Nenkov, J. Reuter, Investing for retirement: The moderating effect of fund assortment size on the 1/n heuristic, Journal of Marketing Research 49 (4) (2012) 537–550.

[54] M. Breton, A. Alj, A. Haurie, Sequential Stackelberg equilibria in two-person games, Journal of Optimization Theory and Applications 59 (1) (1988) 71–97.

[55] B. von Stengel, S. Zamir, Leadership games with convex strategy sets, Games and Economic Behavior 69 (2) (2010) 446–457.

Appendix A. Nomenclature and symbols

The following abbreviations are used in the paper:

AT – Anchoring Theory — the Bounded Rationality model considered in this paper

ATSG – A generalization of the AT model suitable for Sequential Games proposed in this paper, defined in equation (17)

ATSGL – A linear approximation of ATSG proposed in this paper, defined in equation (18)

BC2015 – A MILP-based method for solving Sequential Stackelberg Games, published in [23]

BR – Bounded Rationality

C2016 – A MILP-based method for solving Sequential Stackelberg Games, published in [21]

CBK2018 – A heuristic method employing MILP and game simplification for solving Sequential Stackelberg Games, published in [22]

EASG – An Evolutionary-Algorithm-based method for solving Sequential Stackelberg Games, published in [26]

O2UCT – A Monte-Carlo-sampling-based method for solving Sequential Stackelberg Games, published in [25]

SEG – A set of benchmark games used in [23]

SG – Stackelberg Games

SSG – Stackelberg Security Games

WHG – A set of benchmark games defined in [27]

WNZ – A set of benchmark games defined in [25]

The following notation is used throughout the paper:

- δ_i – mixed strategy of player i, $i = l$ (leader) or $i = f$ (follower)

- π_i – pure strategy of player i, $i = l$ (leader) or $i = f$ (follower)

- σ_i – sequence of moves of player i, $i = l$ (leader) or $i = f$ (follower)

- p_i, q_i – commonly used to denote a probability of some event i

- a – commonly denotes an action (move) to be played in the game; usually used with a subscript

- $u(\cdot, \cdot)$ – expected utility of a game when players play their respective strategies/move sequences

- I – denotes an information set in the game