Two-phase Attacks in Security Games (Supplementary material)

Andrzej Nagórko[1,2] **Paweł Ciosmak**[2] **Tomasz Michalak**[2,3]

[1]Department of Mathematics, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland
[2]Ideas NCBR, ul. Chmielna 69, 00-801 Warsaw, Poland
[3]Department of Computer Science, University of Warsaw, ul. Banacha 2, 02-097 Warsaw, Poland

A NOTATIONAL CONVENTIONS

Throughout the paper, we use variable *superscripts* to denote parameters of probability distributions, e.g., $\{y^t\}_{1\leq t\leq n}$ denotes a family of first-move attacker's strategies, indexed by attacker type t.

Probability distribution dependency on other distributions is usually implicit, but if we want to state explicitly that y^t is picked with the knowledge of x (hence optimal y^t changes with x), we write $y^t(x)$ with *functional* notation.

We use variable *subscripts* to denote values of probability distributions (i. e. we use *matrix* notation), e. g. we write x_i to denote probability assigned to move $i \in I$ by probability distribution $x \in \mathrm{Prob}(I)$. Likewise, for player payoffs, e.g., $r_{i,t,j}$ is the defender's payoff after move i was played against the attacker of type t who played move j.

Parametrized set families are subscripted, as there is no other use for set subscript, e.g., $\mathscr{C}_{t,j}$ is the set of possible payoffs of attacker of type t after he played move j.

Often we use various combinations of variables i, t, j, k as subscripts, always keeping this order in accordance with the order of how values of these variables are picked (see Section 4).

B DOBSS

From discussion in Section 3 we can derive the following quadratic programming solution to one-phase Bayesian Stackelberg games.

$$
\begin{aligned}
&\underset{x,\,y^t}{\text{maximize}} && \sum_{i\in I}\sum_{t=1}^{n}\sum_{j\in J_t} p_t x_i y_j^t r_{i,t,j}, \\
&\text{subject to} && \\
&\sum_{i\in I} x_i = 1, && \\
&\sum_{j\in J_t} y_j^t = 1 && \text{for each } 1\leq t\leq n, \\
&\sum_{i\in I}\sum_{j\in J_t} x_i y_j^t c_{i,t,j} \geq \\
&\qquad \geq \sum_{i\in I} x_i c_{i,t,j} && \text{for each } 1\leq t\leq n, j\in J_t, \\
&x\geq 0, y^t \geq 0 && \text{for each } 1\leq t\leq n.
\end{aligned}
\tag{1}
$$

It is a quadratic program as it contains non-linear terms $x_i y_j^t$. There is no linear program (LP) formulation of polynomial size, as Bayesian Stackelberg Games are known to be NP-hard [Conitzer and Sandholm, 2006]. However, there are two

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

standard ways to deal with non-linear terms that we describe next as they are relevant to the solution of two-phase games studied in this paper.

B.1 HARSANYI TRANSFORMATION

If there is only one attacker type, then a linear relaxation of DOBSS (with constraints $y_j^t \in \{0,1\}$ dropped) computes an optimal strategy of the defender: Stackelberg games with one type of attacker are solvable in polynomial time [Conitzer and Sandholm, 2006]. A Bayesian Stackelberg game can always be transformed into a Stackelberg game (a normal form) using the Harsanyi transformation at the expense of the exponential explosion of the problem size.

In the normal form, the set of moves of the single attacker is a set J of sequences $(j_1, j_2, \ldots, j_n)$ with $j_t \in J_t$, $1 \leq t \leq n$. For move $j \in J$, the defender's payoff for move $i \in I$ is $r_{i,j} = \sum_{t=1}^{n} p_t r_{i,t,j_t}$ and attacker's payoff is $c_{i,j} = \sum_{t=1}^{n} p_t c_{i,t,j_t}$. In other words, the single attacker in a normal-form game selects in a single move attacks for all the attacker's types. The payoffs are the expected payoffs when the probability distribution over the types of the attacker is $\{p_t\}$.

It turns out that the two-phase Bayesian Stackelberg games studied in this paper can be transformed into Bayesian Stackelberg games using a similar transformation. Also, here, this would result in an exponential explosion of the problem size. We describe this in detail in Section 5.

The Harsanyi transformation is not an effective approach to Bayesian Stackelberg games. DOBSS solves the problem exponentially faster, even if the entire branch-and-bound tree is explored in the solution of the mixed integer linear program [Paruchuri et al., 2008]. As we discuss in Section 5, the situation is even worse in the case of two-phase games.

C LINEARIZATION OF PIECEWISE-LINEAR PROBLEMS

Since attackers have optimal pure strategies, without a loss of generality, we may put constraints $y_j^t \in \{0,1\}$ for each $1 \leq t \leq n$, $j \in J_t$ into problem (1). Then for non-linear terms $x_i y_j^t$, $j \in J_t$, we may introduce new variables $a_{i,j}^t$ and constraints

$$0 \leq a_{i,j}^t \leq y_j^t \text{ for each } i \in I, j \in J_t,$$
$$\sum_{j \in J_t} a_{i,j}^t = x_i \text{ for each } i \in I.$$

Since $y^t \in \mathrm{Prob}(J_t)$ and $y_j^t \in \{0,1\}$, in any feasible solution we have $a_{i,j}^t = x_i y_j^t$. We substitute $a_{i,j}^t$ for each occurrence of $x_i y_j^t$ in problem (1) to get mixed integer linear program (MILP) formulation of (1). This is the celebrated DOBSS algorithm [Pita et al., 2009].

In the paper, we exploit the observation that similar substitutions may be performed for any piecewise-linear problem. i.e., a problem in which a feasible set can be decomposed into a finite union of polyhedra with a property that the restriction of the objective function to each polyhedron is linear. Such problems can be characterized to be polynomial problems in which all higher-order terms are products of an arbitrary number of binary variables and, at most, one continuous variable.

D SOLVING TWO-PHASE GAMES

In the present section we derive quadratic and mixed integer optimization problems that compute optimal strategies in two-phase Bayesian Stackelberg games. We start with a MIQP version and then apply linearization trick described in Section C to get a MILP formulation.

D.1 A SOLUTION WITH QUADRATIC PROGRAMMING

Recall quadratic linear problem (4a). We will show that it finds the expected defender's payoff and the optimal attacker's and defender's strategies.

The objective function (4a) is the expected defender's payoff $\mathrm{E}(R + R')$ that he wishes to maximize, from equation (1). Conditions (4b), (4c) and (4d) together with (4h) assure that $x \in \mathrm{Prob}(I)$, $y^t \in \mathrm{Prob}(J_t)$ and $z^{t,j,c} \in \mathrm{Prob}(K_t)$ respectively.

We introduce variables $\gamma_{t,j,c}$ and constraints that enforce that

$$\gamma_{t,j,c} = \max_{k \in K_t} \sum_{i \in I_{t,j,c}} x_i c'_{i,t,j,k}.$$

From (4e), we have $\gamma \geq \max$. Therefore, for each $1 \leq t \leq n$ and each $j \in J_t$, we have

$$\sum_{c \in C_{t,j}} \gamma_{t,j,c} \geq \sum_{c \in \mathscr{C}_{t,j}} \max_{k \in K_t} \sum_{i \in I_{t,j,c}} x_i c'_{i,t,j,k} \geq$$
$$\sum_{c \in \mathscr{C}_{t,j}} \sum_{k \in K_t} \sum_{i \in I_{t,j,c}} x_i z_k^{t,j,c} c'_{i,t,j,k} =$$
$$\sum_{k \in K_t} \sum_{i \in I} x_i z_k^{t,j,c_{i,t,j}} c'_{i,t,j,k}.$$

Hence condition (4f) guarantees that each inequality in the above chain is equality, in particular the first inequality guarantees that for each γ we have $\gamma \leq \max$. It follows from Proposition 4.1 that strategy $z^{t,j,c}$ is optimal if and only if

$$\sum_{k \in K} \sum_{i \in I_{t,j,c}} z_k^{t,j,c} x_i c'_{i,t,j,k} = \gamma_{t,j,c}.$$

The second inequality in the above chain guarantees that it is indeed the case.

From Proposition 4.2, strategy y^t is optimal if and only if

$$\sum_{i \in I} x_i c_{i,t,j} + \sum_{c \in \mathscr{C}_{t,j}} \max_{k \in K_t} \sum_{i \in I_{t,j,c}} x_i c'_{i,t,j,k} \geq$$
$$\sum_{i \in I} x_i c_{i,t,j} + \sum_{c \in \mathscr{C}_{t,j}} \max_{k \in K_t} \sum_{i \in I_{t,j,c}} x_i c'_{i,t,j,k} \text{ for each } j \in J_t.$$

This inequality is encoded as (4g).

D.2 LINEARIZATION

We used substitutions

$$x_i y_j^t z_k^{t,j,c_{i,t,j}} \leftarrow w_{i,t,j,k},$$
$$x_i y_j^t \leftarrow \sum_{k \in K_t} w_{i,t,j,k},$$
$$x_i z_k^{t,j,c_{i,t,j}},$$
$$y_j^t \gamma_{t,j,c} \leftarrow u_{t,j,c}$$

for $1 \leq t \leq n, i \in I, j \in J_t, k \in K_t, c \in \mathscr{C}_{t,j}$.

Constraints (3i), (3d), (3h) and (3o) imply that

$$s_{i,t,j,k} = \begin{cases} x_i & \text{if } z_k^{t,j,c_{i,j}} = 1 \\ 0 & \text{if } z_k^{t,j,c_{i,j}} = 0, \end{cases}$$

hence indeed $s_{i,t,j,k} = x_i z_k^{t,j,c_{i,t,j}}$ in any feasible solution.

Constraints (3j), (3k), (3o) for big enough M imply that

$$u_{t,j,c} = \begin{cases} \gamma_{t,j,c} & \text{if } y_j^t = 1 \\ 0 & \text{if } y_j^t = 0, \end{cases}$$

hence indeed $u_{t,j,c} = y_j^t \gamma_{t,j,c}$.

	T_1		T_2		T_3		T_4		$\emptyset$	
T_1T_2	13,	-13	24,	-21	-42,	41	-85,	81	0,	0
T_1T_3	13,	-12	-20,	23	44,	-45	-80,	81	0,	0
T_1T_4	15,	-15	-22,	20	-45,	42	85,	-85	0,	0
T_2T_3	-14,	13	24,	-25	41,	-42	-82,	84	0,	0
T_2T_4	-13,	14	23,	-24	-40,	43	81,	-85	0,	0
T_3T_4	-13,	13	-25,	21	42,	-44	85,	-85	0,	0

	T_1		T_2		T_3		T_4		$\emptyset$	
T_1T_2	54,	-68	125,	-124	-202,	208	-403,	415	0,	0
T_1T_3	74,	-64	-115,	120	212,	-225	-406,	403	0,	0
T_1T_4	65,	-50	-112,	113	-219,	224	424,	-400	0,	0
T_2T_3	-72,	64	108,	-123	225,	-207	-418,	403	0,	0
T_2T_4	-60,	50	100,	-100	-220,	217	400,	-412	0,	0
T_3T_4	-71,	56	-113,	123	200,	-216	407,	-424	0,	0

Table 1: Payoff matrices discussed in Example E.1

Finally, constraints (3l), (3m), (3n) and (3o) imply that

$$w_{i,t,j,k} = \begin{cases} x_i & \text{if } y_j^t = 1 \text{ and } z_k^{t,j,c_{i,j}} = 1 \\ 0 & \text{if } y_j^t = 0 \text{ or } z_k^{t,j,c_{i,j}} = 0, \end{cases}$$

hence indeed $w_{i,t,j,k} = x_i y_j^t z_k^{t,j,c_{i,t,j}}$.

This shows equivalence of the MILP formulation (3a) and the MIQP formulation (4a).

E TRANSFORMATION TO SINGLE-PHASE GAME

Example E.1. For a Los Angeles airport security game with 4 terminals and 2 patrols with payoff matrices given in Table 1 (notice varying attacker payoffs) a two-phase MILP formulation has 465 variables (with 115 binary variables). In the reduction to a single-phase game discussed above, the attacker has 34505 moves. For this reduction, the MIQP formulation of DOBSS (which is much smaller than the MILP formulation) has 34513 variables (with 34505 binary variables).

References

Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, page 82–90, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595932364. doi: 10.1145/1134707.1134717. URL https://doi.org/10.1145/1134707.1134717.

Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 895–902, 2008.

James Pita, Manish Jain, Fernando Ordónez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Using game theory for los angeles airport security. *AI magazine*, 30(1):43–43, 2009.