A study of general and security Stackelberg game formulations

Carlos Casorrán\textsuperscript{a,b,c,*}, Bernard Fortz\textsuperscript{a,b}, Martine Labbé\textsuperscript{a,b}, and Fernando Ordóñez\textsuperscript{c}

\textsuperscript{a}Département d’Informatique, Université Libre de Bruxelles, Brussels, Belgium
\textsuperscript{b}INOCs, INRIA Lille Nord-Europe, Lille, France
\textsuperscript{c}Departamento de Ingeniería Industrial, Universidad de Chile, Santiago, Chile
*Corresponding author. E-mail: ccasorra@ulb.ac.be

Abstract

In this paper we analyze general Stackelberg games (GSGs) and Stackelberg security games (SSGs). Stackelberg games are hierarchical adversarial games where players select strategies to optimize their payoffs in a sequential manner. SSGs are a type of GSGs that arise in security applications where the strategies of the player that acts first consist in protecting subsets of targets and the strategies of the followers consist in attacking a target. We present a comparative study of existing mixed integer linear programming (MILP) formulations for GSGs, where we rank them according to the tightness of their linear programming (LP) relaxations. We establish a theoretical link between GSG and SSG formulations through projections of variables and exploit this link to derive a new tight SSG MILP formulation. We extend our comparison of GSG formulations to the security setting, showing that the new SSG formulation we derive i) has the tightest LP relaxation known among SSG MILP formulations and ii) its LP relaxation coincides with the convex hull of feasible solutions in the case of a single follower. We run computational experiments in both the general and the security setting to measure the performance of the corresponding formulations. Our new SSG formulation outperforms competing formulations and is better suited to tackle scaling up of the instances.

Keywords: Integer programming, Discrete optimization, Game theory, Bilevel optimization.

1 Introduction

Stackelberg games (SG) model situations where players with different objectives strive to optimize their payoff in a sequential, one-off encounter. If a player can commit to a given action first, it is referred to as the leader, whereas the players responding to the leader’s
action are referred to as the followers. If there is one leader and one follower, the payoffs in such a game are encoded in a matrix whose elements are couples. In this matrix, each row represents an action of the leader and each column represents an action of the follower. The payoffs that both players obtain, as a result of selecting a row and column, are determined by the values in the couple at the intersection between the chosen row and column: the first element of the pair is the reward for the leader and the second element of the pair is the reward for the follower. If the leader (resp. follower) commits to a single row (resp. column), he plays a pure strategy whereas if the leader (resp. follower) assigns a probability to a row (resp. column), he plays a mixed strategy. The leader’s objective is to take an action—a mixed strategy—that maximizes his payoff, knowing that the follower is aware of such an action and in turn optimizes his reward by selecting a strategy of his own.

When the leader faces a single follower, the problem is polynomially solvable and can be solved using the multiple LP method in [Conitzer and Sandholm, 2006]. If there are several followers, each with a distinct payoff matrix, then a model, due to [Harsanyi and Selten, 1972], combines the leader’s rewards over the different followers and assumes a known probability of facing each follower to compute the leader’s expected reward. We refer to this as the ‘p-follower’ problem. This problem is NP-hard, as shown in [Conitzer and Sandholm, 2006]. Solution methods for this problem are due to [Paruchuri et al., 2008], [Jain et al., 2011], [Yang et al., 2013], among others. If the number of follower types is fixed, the problem is polynomially solvable; the distinct payoff matrices can be transformed into a single payoff matrix, reducing the multiple follower type case to a single follower type case, by applying the Harsanyi transformation [Harsanyi and Selten, 1972]. The number of follower strategies in this single payoff matrix is exponential in the number of followers.

Stackelberg game theory has been recently used to solve real-world security problems. In this domain, the leader is referred to as the defender whereas followers are referred to as attackers. Stackelberg security game-theoretic applications have included effectively assigning Federal Air Marshals to transatlantic flights [Jain et al., 2010], determining randomized patrols for the U.S. Coast Guard to efficiently protect port infrastructure [Shieh et al., 2012], preventing fare evasion in public transport systems [Yin et al., 2012] as well as protecting endangered wildlife [Yang et al., 2014].

In this paper we present the following four key contributions, i) we provide an exhaustive comparative study of existing mixed integer linear programming (MILP) SG formulations. Starting from a natural bilevel representation of a general SG, we use well-known integer programming techniques such as Fourier-Motzkin elimination [Dantzig and Eaves, 1973] and Reformulation Linearization Technique [Sherali and Adams, 1994] to derive MILP for-
mulations from the literature. Our study leads to a ranking of known MILP formulations in terms of the strength of their linear programming (LP) relaxations; ii) we explicit a formal link between general SG (GSG) and security SG (SSG) formulations through projections of some variables. This allows to extend our study of GSG formulations to the security setting, leading to a comparison of SSG MILP formulations; iii) exploiting the projection link between formulations in the general and security setting, we derive (MIP-p-S_{q,y}), a new strong SSG MILP formulation. We show that (MIP-p-S_{q,y}) is the MILP formulation with the tightest linear relaxation among SSG formulations. We further show that its restriction to a single follower type is an ideal formulation, i.e., its linear relaxation provides a complete linear description of the convex hull of its feasible solutions; iv) we provide computational experiments that compare the performance of the MILP formulations in both settings. Our experiments show that the formulations with the tightest LP relaxations have faster solving times. In addition, (MIP-p-S_{q,y}) provides scaling-up capabilities beyond those of competing formulations, being able to tackle larger-sized instances.

This paper is organized as follows. In Section 2, we define general and security Stackelberg games. In Section 3, we deduce GSG formulations from the literature. We provide theoretical results comparing the formulations presented. In Section 4, we describe and analyze computational experiments for the formulations in Section 3. In Section 5, we present SSG formulations using projections, in the appropriate space of variables, of the formulations in Section 3, and derive (MIP-p-S_{q,y}), a new MILP formulation for SSGs. We then extend our theoretical comparisons of the general formulations to the security formulations. In Section 6, we describe and analyze the computational experiments for the security formulations. We conclude with some closing remarks in Section 7.

2 Notation and definition of the problem

In this section, we provide a formal definition of the two types of problems we study.

2.1 General Stackelberg games–GSGs

Let $K$ be the set of $p$ followers. We denote by $I$ the set of leader pure strategies and by $J$ the set of follower pure strategies. The leader has a known probability of facing follower $k \in K$, denoted by $\pi^k \in [0, 1]$. We denote the $n$-dimensional simplex by $\mathbb{S}^n = \{ x \in [0,1]^n : \sum_n x_i = 1 \}$. A mixed strategy for the leader consists in a vector $x \in \mathbb{S}^{|I|}$ such that for $i \in I$, $x_i$ is the probability with which the leader plays pure strategy $i$. Analogously, a mixed strategy for a follower $k \in K$ is a vector $q^k \in \mathbb{S}^{|J|}$ such that, $q^k_j$ is the probability with which follower $k$ replies with pure strategy $j \in J$. The rewards or payoffs for the leader and each follower,
resulting from their choice of strategy, are encoded in a different matrix for each follower. These payoff matrices are denoted by \((R^k, C^k)\), where \(R^k \in \mathbb{R}^{I \times J}\) is the leader’s reward matrix when facing follower \(k \in K\) and \(C^k \in \mathbb{R}^{I \times J}\) is the reward matrix for follower \(k\).

The expected reward of the leader and follower \(k\), respectively, can be expressed as follows:

\[
\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \pi^k R^k_{ij} x_i q^k_j, \quad (1)
\]

\[
\sum_{i \in I} \sum_{j \in J} C^k_{ij} x_i q^k_j, \quad \forall k \in K. \quad (2)
\]

For all \(k \in K\) we define the function \(B^k : \mathbb{S}^I \rightarrow \mathbb{S}^J\) as the function that, given the leader mixed strategy \(x\), returns a best response \(q^k\) for each follower \(k\). The solution concept used in these games is the Strong Stackelberg Equilibrium (SSE), introduced in [Leitman, 1978] and defined below.

**Definition 1.** A profile of mixed strategies \((x, \{B^k(x)\})_{k \in K}\) form an SSE if:

1. The leader always plays a payoff-maximizing strategy:
   \[
x^T R^k B^k(x) \geq x^T R^k B^k(x') \quad \forall x' \in \mathbb{S}^I, \forall k \in K.
   \]

2. Each follower always plays a best-response, \(B^k(x) \in F^k(x)\), where \(\forall k \in K\),
   \[
   F^k(x) = \arg \max_{q^k} \{x^T C^k q^k : q^k \in \mathbb{S}^J\}
   \]
   is the set of best responses for each follower.

3. Each follower breaks ties optimally in favor of the leader:
   \[
x^T R^k B^k(x) \geq x^T R^k q^k \quad \forall q^k \in F^k(x).
   \]

An SSE assumes that the follower breaks ties in favor of the leader by choosing, when indifferent between different follower strategies, the strategy that maximizes the payoff of the leader. An SSE is in practice always achievable as the leader can always induce one by selecting a sub-optimal mixed strategy arbitrarily close to the equilibrium, causing the follower to prefer the desired strategy [von Stackelberg, 1952].

**Remark 1.** For any leader strategy \(x\) and any \(k \in K\), there is a best response to the \(k\)-th follower’s problem that is given by a vector \(q^k \in \{0, 1\}^{|J|}\) such that \(\sum_{j \in J} q^k_j = 1\).

**Proof.** Assume that \(B^k(x) = q^k \notin \{0, 1\}^{|J|}\). We show that any canonical vector \(e^{jk}\) such that \(q^k_j > 0\), is also a best response vector, i.e., \(e^{jk} \in F^k(x)\) and \(x^T R^k e^{jk} \geq x^T R^k q^k\) for all \(q^k \in F^k(x)\). Since \(q^k = \sum_{j \in J} q^k_j e^{jk}\), with \(e^{jk} \in \mathbb{S}^{|J|}\), and \(x^T C^k e^{jk} \leq x^T C^k q^k\) for all \(j \in J\),
we have that \( x^T C^k \tilde{q}^k = \sum_{j \in J} \tilde{q}^k_j (x^T C^k e^k) \leq \sum_{j \in J} \tilde{q}^k_j (x^T C^k q^k) = x^T C^k \tilde{q}^k \). This implies that for any \( \tilde{q}^k_j > 0 \) we have \( x^T C^k e^k = x^T C^k \tilde{q}^k \), giving \( e^k \in F^k(x) \). A similar argument shows that for any \( j \) such that \( \tilde{q}^k_j > 0 \) we have \( x^T R^k e^k = x^T R^k \tilde{q}^k \); Hence, \( e^k \) is a best response vector.

In Mathematical Optimization, Stackelberg games are addressed by Bilevel Programming (BP). Introduced in [Bracken and McGill, 1973], BP targets hierarchical optimization problems in which part of the constraints translate the fact that some of the variables are an optimal solution to another nested optimization problem. Important BP surveys are those by [Kolstad, 1985, Savard, 1989, Anandalingam and Friesz, 1992, Labbé and Violin, 2016]. In our setting, the first level problem corresponds to the leader’s decision problem and the nested problem corresponds to the follower’s decision problem. The following model, \((\text{BIL}-p\text{-G}_{x,q})\), is a bilevel program for the general Stackelberg game problem:

\[
\begin{align*}
\text{Max}_{x,q} & \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \pi^k R^k_{ij} x_i q^k_j \\
\text{s.t.} & \quad \sum_{i \in I} x_i = 1, \\
& \quad x_i \in [0,1] \quad \forall i \in I, \\
& \quad q^k \in \arg \max_{q^k} \left\{ \sum_{i \in I} \sum_{j \in J} C^k_{ij} x_i r^k_j \right\} \quad \forall k \in K, \\
& \quad r^k_j \in \{0,1\} \quad \forall j \in J, \forall k \in K, \\
& \quad \sum_{j \in J} r^k_j = 1 \quad \forall k \in K.
\end{align*}
\]

The objective function maximizes the leader’s expected reward. Constraints (4)-(5) characterize the mixed strategies considered by the leader. The second level problem defined by (6)-(8) indicates that the follower maximizes his own payoff by best responding with a pure strategy to the leader’s commitment. If there are multiple optimal strategies for the follower, the main level problem selects the one that benefits the objective of the leader.

### 2.2 Stackelberg security games—SSGs

A Stackelberg security game (SSG) involves allocating the defender’s security resources to protect a subset of targets. Let \( J \) be the set of \( n \) targets that could be attacked and let \( \Omega \) be the set of \( m < n \) security resources available to protect these targets. Allocating resource \( \omega \in \Omega \) to target \( j \in J \) protects the target. The set \( I \) of defender pure strategies \( i \in I \) is composed by all \( \sum_{i=1}^{m} \binom{n}{i} \) subsets of at most \( m \) targets of \( J \) that the defender can protect simultaneously. The elements \( j \in J \) constitute the pure strategies of each attacker.
In SSGs, payoffs for the players only depend on whether a target is attacked and whether that target was covered or not. This means that many of the strategies have identical payoffs. We use this fact to construct a compact representation of the payoffs.

We denote by $D^k$ the utility of the defender when facing an attacker $k \in K$ and by $A^k$ the utility of attacker $k$. Associated with each target and each player are two payoffs depending on whether or not the target is covered, see Table 1. [Kiekintveld et al., 2009]

<table>
<thead>
<tr>
<th>Covered</th>
<th>Uncovered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender</td>
<td>$D^k(j</td>
</tr>
<tr>
<td>Attacker</td>
<td>$A^k(j</td>
</tr>
</tbody>
</table>

Table 1: Payoff structure in an SSG when target $j$ is attacked by an attacker $k$

We take advantage of the aforementioned compact representation to define a coverage vector $c$ whose components, $c_j$, represent the frequency of coverage of target $j$. The components of the vector $c$ satisfy $c_j = \sum_{i: j \in i} x_i$, $\forall j \in J$, i.e., the frequency of coverage is expressed as the sum of all probabilities of the strategies that assign coverage to that target. Variables $q^k_j$ indicate whether an attacker $k$ strikes a target $j$.

The defender’s and attacker $k$’s expected rewards, are, respectively:

$$\sum_{j \in J} \sum_{k \in K} \pi \cdot q^k_j \{c_j D^k(j|c) + (1 - c_j) D^k(j|u)\},$$  \hspace{1cm} (9)

$$\sum_{j \in J} q^k_j \{c_j A^k(j|c) + (1 - c_j) A^k(j|u)\}, \hspace{1cm} \forall k \in K.$$  \hspace{1cm} (10)

As with GSGs, such a game can be modeled by means of bilevel programming.

$$(\text{BIL-p-S}_{x,c,q})$$

$$\text{Max } \sum_{j \in J} \sum_{k \in K} \pi \cdot q^k_j \{c_j D^k(j|c) + (1 - c_j) D^k(j|u)\}$$

s.t. \hspace{1cm} \sum_{i \in I} x_i = 1,  \hspace{1cm} (11)

$$x_i \geq 0 \hspace{1cm} \forall i \in I, \hspace{1cm} (12)$$

$$\sum_{i: j \in i} x_i = c_j \hspace{1cm} \forall j \in J, \hspace{1cm} (13)$$

$$q^k \in \arg \max r^k \left\{ \sum_{j \in J} r^k_j \{c_j A^k(j|c) + (1 - c_j) A^k(j|u)\} \right\} \hspace{1cm} \forall k \in K,$$

$$r^k_j \in \{0, 1\} \hspace{1cm} \forall j \in J, \forall k \in K,$$

$$\sum_{j \in J} r^k_j = 1 \hspace{1cm} \forall k \in K.$$
The objective function maximizes the defender’s expected reward. Constraints (11)-(13) characterize the exponentially many mixed strategies considered by the defender and relate them to coverage frequencies over the targets. The remaining constraints constitute the second level optimization problem which ensures that the attacker maximizes his profit by attacking a single target, best responding to the defender’s selected strategy. Remark that a more compact formulation—one involving a polynomial number of variables and constraints—can be obtained if projecting out the exponentially many \( x \) variables does not lead to exponentially many constraints. This would give a polynomial size formulation involving only the \( c \) and the \( q \) variables. Given an optimal solution to this compact formulation—an optimal coverage vector \( c \) and an optimal attack vector \( q \)—a probability vector \( x \), solution to this game in extensive form, can be obtained by solving the system of linear inequalities defined by (11), (12) and (13). As this system involves \( n + 1 \) equalities, there exists a solution in which the number of variables \( x_i \) with a positive value is not larger than \( n + 1 \), i.e., the output size of an SSG, under extensive form, is polynomial in the input size. See Section 5 for more details.

3 General Stackelberg games—GSGs

In Section 3.1, we present equivalent MILP formulations for the \( p \) follower GSG. In Section 3.2 we compare the polyhedra of the LP relaxations for the different formulations.

3.1 General Stackelberg games: single level formulations

[Paruchuri et al., 2008] tackle the problem of solving the bilevel formulation presented earlier, (BIL-\( p \)-G\( x,q \)) by using a MILP reformulation. They replace the second level nested optimization problem, described by (6)-(8), by the following set of constraints:

\[
\sum_{j \in J} q^k_j = 1 \quad \forall k \in K, \quad (14)
\]

\[
q^k_j \in \{0, 1\} \quad \forall j \in J, \forall k \in K, \quad (15)
\]

\[
0 \leq (s^k - \sum_{i \in I} C^k_{ij} x_i) \leq (1 - q^k_j) \cdot M \quad \forall j \in J, \forall k \in K, \quad (16)
\]

where \( s^k \in \mathbb{R} \) for all \( k \in K \). The two inequalities in Constraint (16) ensure that \( q^k_j = 1 \) only for a pure strategy that maximizes the follower’s payoff. The problem defined by (3)-(5) and (14)-(15) is referred to as (QUAD\( x,q,s \)).

Formulation (D2\( x,q,a,f \)), below, avoids the quadratic term in the objective of (BIL-\( p \)-G\( x,q \)) by adding \( |K| \) new variables and introducing a second family of constraints involving a big
M constant.

\[
\text{(D2}_{x,q,s,f}) \quad \text{Max} \quad \sum_{k \in K} \pi^k f^k \tag{17}
\]

s.t. \quad \text{(14) \minus (16)},

\[
f^k \leq \sum_{i \in I} R^k_{ij} x_i + (1 - q^k_{ij}) \cdot M \quad \forall j \in J, \forall k \in K, \tag{18}
\]

\[
\sum_{i \in I} x_i = 1, \tag{19}
\]

\[
x_i \geq 0 \quad \forall i \in I, \tag{20}
\]

\[
s, f \in \mathbb{R}^{\vert K \vert} \quad \forall k \in K.
\]

Alternatively, one can eliminate the nonlinearity in the objective function [Paruchuri et al., 2008], by adding additional variables that represent the product between \( x \) and \( q \). To be more precise, we introduce \( z_{ij}^k = x_i q_{kj}^k \) for all \( i \in I, \ j \in J \) and \( k \in K \). This gives rise to the following formulation called (DOBSS\(_{q,z,s}\)):

\[
\text{(DOBSS}_{q,z,s}) \quad \text{Max} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \pi^k R^k_{ij} z_{ij}^k \tag{21}
\]

s.t. \quad \text{(14), (15)},

\[
\sum_{j \in J} z_{ij}^k = \sum_{j \in J} z_{ij}^1 \quad \forall i \in I, \forall k \in K, \tag{22}
\]

\[
\sum_{i \in I} z_{ij}^k = q_{ij}^k \quad \forall j \in J, \forall k \in K, \tag{23}
\]

\[
z_{ij}^k \geq 0 \quad \forall i \in I, \forall j \in J, \forall k \in K,
\]

\[
0 \leq s^k - \sum_{i \in I} \sum_{j' \in J} C_{ij}^k z_{ij}^{k'} \leq (1 - q_{ij}^k) \cdot M \quad \forall j \in J, \forall k \in K, \tag{24}
\]

\[
s \in \mathbb{R}^{\vert K \vert}.
\]

Additionally, the real variables \( s^k \) in Constraints (16) and (24) can be projected out by using Fourier-Motzkin elimination [Dantzig and Eaves, 1973]. This gives rise to constraints:

\[
\sum_{i \in I} (C_{ij}^k - C_{ij}^k) x_i \leq (1 - q_{ij}^k) \cdot M \quad \forall j, \ell \in J, \forall k \in K, \tag{25}
\]

\[
\sum_{i \in I} \sum_{j' \in J} (C_{ij}^k - C_{ij}^k) z_{ij}^{k'} \leq (1 - q_{ij}^k) \cdot M \quad \forall j, \ell \in J, \forall k \in K. \tag{26}
\]

Replacing (16) by (25) in (D2\(_{x,q,s,f}\)) and (24) by (26) in (DOBSS\(_{q,z,s}\)) yields (D2\(_{x,q,f}\)) and (DOBSS\(_{q,z}\)). We analyze the behavior of these last two formulations compared to that of (D2\(_{x,q,s,f}\)) and (DOBSS\(_{q,z,s}\)) to see if removing variables \( s \) at the expense of adding constraints is worthwhile.
Another equivalent MILP formulation for the p-follower GSG can be obtained by replacing Constraint (24) with the following constraint:

$$\sum_{i \in I} \left( C^k_{ij} - C^k_{i\ell} \right) z^k_{ij} \geq 0 \quad \forall j, \ell \in J, \forall k \in K. \quad (27)$$

This constraint is derived by multiplying Constraint (25) by $q^k_{ij}$, reorganizing and replacing the nonlinear terms $x_i q^k_{ij}$ by $z^k_{ij}$. This leads to (MIP-p-G$_{q,z}$):

$$(\text{MIP-p-G}_{q,z}) \quad \text{Max} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \pi^k R^k_{ij} z^k_{ij}
\text{s.t.} \quad (14), (15), (21) - (23),$$

$$\sum_{i \in I} \left( C^k_{ij} - C^k_{i\ell} \right) z^k_{ij} \geq 0 \quad \forall j, \ell \in J, \forall k \in K. \quad (28)$$

The linear relaxation of (MIP-p-G$_{q,z}$) appears in [Yin and Tambe, 2012]. The MILP formulation is a p-follower extension to the single follower formulation (MIP-1-G$_{q,z}$), due to [Conitzer and Korzhyk, 2011]. Formal proofs that the formulations seen thus far are equivalent MILP formulations, i.e., that they are valid for the p-follower GSG, appear in [Paruchuri et al., 2008], for (DOBSS$_{q,z,s}$) and [Paruchuri et al., 2008] and [Kiekintveld et al., 2009] for (D2$_{x,q,f}$). These proofs show that each of them is equivalent to (QUAD$_{x,q,s}$). The equivalence of (DOBSS$_{q,z}$) and (D2$_{x,q,f}$) is obtained from the Fourier-Motzkin elimination procedure [Dantzig and Eaves, 1973]. The equivalence proof for (MIP-p-G$_{q,z}$) is analogous to the proof used to show the equivalence for (DOBSS$_{q,z,s}$) and is omitted here.

[Paruchuri et al., 2008] state that the big M constants used are arbitrarily large. To be as computationally competitive as possible, we provide the tightest value for each big M constant in the formulations discussed thus far.

**Proposition 1.** The tightest values for the positive constants $M$ are:

1. In (18), $M = \max_{i \in I} \left\{ \max_{\ell \in J} \{ R^k_{i\ell} - R^k_{ij} \} \right\} \forall j, \forall k \in K$.
2. In (16) and (24), $M = \max_{i \in I} \left\{ \max_{\ell \in J} \{ C^k_{i\ell} - C^k_{ij} \} \right\} \forall j, \forall k \in K$.
3. In (25) and (26), $M = \max_{i \in I} \{ C^k_{ij} - C^k_{i\ell} \}, \forall j, \ell \in J, \forall k \in K$.

**3.2 Comparison of the formulations**

Given a formulation $F$, we denote by $\overline{F}$ its linear (continuous) relaxation and by $\mathcal{P}(\overline{F})$ the polyhedral feasible region of $\overline{F}$. Further, let $Q = \{(x,z) \in \mathbb{R}^n \times \mathbb{R}^m : A x + B z \leq d\}$. Then the projection of $Q$ into the $x$-space, denoted $Proj_x Q$, is the polyhedron given by $Proj_x Q = \{ x \in \mathbb{R}^n : \exists z \in \mathbb{R}^m \text{ for which } (x,z) \in Q \}$, see [Pochet and Wolsey, 2006].
First, we introduce an additional formulation which we denote by \((\text{DOBSS}_{x,q,z,s,f})\). This formulation is equivalent to \((\text{DOBBS}_{q,z,s})\), in the sense that the values of their LP relaxations coincide. In this formulation we introduce variables \(f^k\) for all \(k \in K\) to rewrite the objective function so that it matches (17). We also add variables \(x_i\) for all \(i \in I\) by rewriting (21) as \(\sum_{j \in J} z_{ij}^k = x_i\) for all \(i \in I\) and all \(k \in K\). Using this last condition, we can simplify (24) to (16). The formulation \((\text{DOBSS}_{x,q,z,s,f})\) is as follows.

\[\text{(DOBSS}_{x,q,z,s,f}) \quad \text{Max} \quad \sum_{k \in K} a^k f^k\]

\[\text{s.t.} \quad (14) - (16),\]

\[\sum_{i \in I} z_{ij}^k = q_j^k \quad \forall j \in J, \forall k \in K, \quad (29)\]

\[z_{ij}^k \geq 0 \quad \forall i \in I, \forall j \in J, \forall k \in K, \quad (30)\]

\[f^k = \sum_{i \in I} \sum_{j \in J} R_{ij}^k z_{ij}^k \quad \forall k \in K, \quad (31)\]

\[\sum_{j \in J} z_{ij}^k = x_i \quad \forall i \in I, \forall k \in K, \quad (32)\]

\[s \in \mathbb{R}^{|K|}.\]

Further, note that from the Fourier Motzkin elimination procedure we have that

\[\mathcal{P}(\overline{\mathcal{D}^2_{x,q,f}}) = \text{Proj}_{x,q,f}\mathcal{P}(\overline{\mathcal{D}^2_{x,q,s,f}}) \text{ and,}\]

\[\mathcal{P}(\text{DOBSS}_{q,z}) = \text{Proj}_{q,z}\mathcal{P}(\text{DOBSS}_{q,z,s}).\]

**Proposition 2.** \(\text{Proj}_{x,q,s,f}\mathcal{P}(\text{DOBSS}_{x,q,z,s,f}) \subseteq \mathcal{P}(\overline{\mathcal{D}^2_{x,q,s,f}}).\) Further, there exist instances for which the inclusion is strict.

**Proof.** Note that all the constraints of \(\mathcal{P}(\overline{\mathcal{D}^2_{x,q,s,f}})\) can be found in the description of \(\mathcal{P}(\text{DOBSS}_{x,q,z,s,f})\) except for Constraints (18) and (19). Constraint (19) is implied by Constraints (14), (29) and (32).

Further, the projection of \(\mathcal{P}(\text{DOBSS}_{x,q,z,s,f})\) on the \((x,q,s,f)\)-space can be obtained by applying Farkas’ Lemma [Farkas, 1902]. Constraints (29), (30), (31) and (32) are the only ones involving variables \(z_{ij}^k\) and are separable by \(k \in K\). For a fixed \(k \in K\) the projection is given by:

\[A^k = \{(x,q,f) : \alpha f^k + \sum_{i \in I} \beta_i x_i + \sum_{j \in J} \gamma_j q_j^k \geq 0 \forall (\alpha, \gamma, \beta) : \alpha R_{ij}^k + \beta_i + \gamma_j \geq 0 \forall i \in I, \forall j \in J\}\]

(33)

For a fixed \(j \in J\), define \(\alpha = -1, \beta_i = R_{ij}^k\) for all \(i \in I\), \(\gamma_j = 0\) and \(\gamma_\ell = \max_{i \in I} (R_{ij}^k - R_{ij}^\ell)\) for all \(\ell \in J\) with \(\ell \neq j\). This definition of the parameters satisfies \(\alpha R_{ij}^k + \beta_i + \gamma_j \geq 0\) for
all \( i, j \in J \). Substituting these parameters in the generic constraint of \( A^k \) yields

\[
f^k \leq \sum_{i \in I} R^k_{ij} x_i + \sum_{\ell \in J, \ell \neq j} \max_{i \in I} (R^k_{\ell i} - R^k_{ij}) q^k_{\ell} \quad \forall j \in J, \forall k \in K. \tag{34}
\]

Constraint (34) implies Constraint (18) for the tight value of \( M \) provided in Proposition 1 since for all \( j \in J \) and \( k \in K \),

\[
\sum_{\ell \in J, \ell \neq j} \max_{i \in I} (R^k_{\ell i} - R^k_{ij}) q^k_{\ell} \leq \max_{i \in I} \left\{ \max_{\ell \in J, \ell \neq j} (R^k_{\ell i} - R^k_{ij}) \right\} \sum_{\ell \in J, \ell \neq j} q^k_{\ell} = \max_{i \in I} \left\{ \max_{\ell \in J, \ell \neq j} (R^k_{\ell i} - R^k_{ij}) \right\} (1 - q^k_j).
\]

This proves the inclusion. To show that the inclusion may be strict, consider the following example where \(|I| = |J| = 3\) and \(|K| = 1\). Let the payoff matrix for the game be

\[
(R, C) = \begin{pmatrix}
(1, 0) & (0, 0) & (0, 0) \\
(0, 0) & (1, 0) & (0, 0) \\
(0, 0) & (0, 0) & (0, 0)
\end{pmatrix}
\]

and consider the point defined by \( x = (1, 0, 0)^t, q = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t, s = 10 \) and \( f = 2/3 \). Such a point is feasible for \( (\overline{D2}_{x,q,s,f}) \) but violates Constraint (34) for \( j = 2 \) and is therefore infeasible for \( \text{Proj}_{x,q,s,f} \mathcal{P}(\text{DOBSS}_{x,q,s,f}) \).

Next, we compare the polyhedra \( \mathcal{P}(\text{MIP-}p\text{-G}_{q,z}) \) and \( \text{Proj}_{q,z} \mathcal{P}(\text{DOBSS}_{q,z}) \).

**Theorem 1.** \( \mathcal{P}(\text{MIP-}p\text{-G}_{q,z}) \subseteq \mathcal{P}(\text{DOBSS}_{q,z}) = \text{Proj}_{q,z} \mathcal{P}(\text{DOBSS}_{q,z,s}) \). Further, there exist instances for which the inclusion is strict.

**Proof.** The description of \( \mathcal{P}(\text{DOBSS}_{q,z}) \) differs from that of \( \mathcal{P}(\text{MIP-}p\text{-G}_{q,z}) \) by only one constraint: (26) must hold instead of (28). Hence, the remainder of the proof consists in showing that (26) is implied by (14), (21)-(23), (28) and the nonnegativity of the \( q \) variables.

The LHS of (26) can be rewritten as:

\[
\sum_{i \in I} (C^k_{ij} - C^k_{i\ell}) z^k_{ij} + \sum_{i \in I} \sum_{j' \in J, j' \neq \ell} (C^k_{ij} - C^k_{i\ell}) z^k_{ij'} \leq \sum_{i \in I} \sum_{j' \in J, j' \neq \ell} (C^k_{ij} - C^k_{i\ell}) z^k_{ij'}, \quad \text{using (28),}
\]

\[
\leq \max_{i \in I} \{C^k_{ij} - C^k_{i\ell}\} \sum_{j' \in J, j' \neq \ell} \sum_{i \in I} z^k_{ij'} \leq M \sum_{j' \in J, j' \neq \ell} q^k_{j'}, \quad \text{given Proposition 1 and (29)}
\]

\[
= M(1 - q^k_j), \quad \text{by (14).}
\]

To show that the inclusion may be strict consider the \( p \)-follower GSG between a leader and a fixed follower \( k \in K \) where the payoff bimatrix is:

\[
(R^k, C^k) = \begin{pmatrix}
(0, 1) & (1, 0) \\
(0, 0) & (0, 0)
\end{pmatrix}
\]

\[
\]
The point with coordinates \( x = (1/2, 1/2)^t \), \( q^k = (1/2, 1/2)^t \) and
\[
\gamma^k = \begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}
\]
has an objective value of 1/4 and is feasible in \( P(D_{DOBSS_q,z}) \). However it is not a feasible point in \( P(MIP-p-G_{q,z}) \) as it doesn’t verify Constraint (28) for values of \( j = 2 \) and \( \ell = 1 \). ■

From an interpretation point of view, \( (MIP-p-G_{q,z}) \) can be seen as the result of applying Reformulation Linearization Technique (RLT) [Sherali and Adams, 1994] to \( (DOBSS_{q,z}) \). Indeed, by multiplying both sides of Constraint (25) by variable \( q^k_\ell \) and noticing that \( q^k_\ell (1-q^k_\ell) = 0 \) since \( q \) is binary, one obtains
\[
\sum_{i \in I} (C^k_{ij} - C^k_{i\ell}) xi q^k_\ell \leq 0
\]
which, once linearized by introducing variables \( z^k_{i\ell} \), yields (28).

For a given formulation \( F \), we denote its optimal value by \( v(F) \) and the optimal value of its LP relaxation by \( v(F) \). Since \( (D2_{x,q,s,f}) \) and \( (DOBSS_{x,q,s,f}) \) and \( (DOBSS_{q,z}) \) and \( (MIP-p-G_{q,z}) \) have the same objective function, the following corollary holds.

**Corollary 1.** \( v(MIP-p-G_{q,z}) \leq v(DOBSS_{q,z}) = v(DOBSS_{x,q,s,f}) \leq v(D2_{x,q,s,f}). \)

Finally, when \( (MIP-p-G) \) is restricted to a single follower type, [Conitzer and Korzhyk, 2011] showed that the integrality costraints are redundant, i.e., the remaining constraints in \( (MIP-1-G) \) provide a complete linear description of the convex hull of feasible solutions.

## 4 Computational experiments for GSGs

We present computational experiments for the formulations in Section 3. The machine used for these experiments is an Intel Core i7-4930K CPU, 3.40GHz, equipped with 64 GByte RAM, 6 cores, 12 threads and operating system Ubuntu release 12.10 (kernel linux 3.5.0-41-generic). The experiments were coded in the programming language Python and GUROBI version 6.5.1 was the optimization solver used with a 3 hour solution time limit.

The instances solved in the computational experiments are randomly generated. We consider two different ways of randomly generating the payoff matrices for the leader and the different follower types. First, we consider matrices where all the elements are randomly generated between 0 and 10 and second, we consider matrices where 90% of the values are between 0 and 10 but we allow for 10% of the data to deviate between 0 and 100. In the first case we say that there is no variability in the payoff matrices, in the sense that all the data is uniformly distributed, whereas in the second case, we refer to the payoff matrices as matrices with variability.

A general Stackelberg game instance is defined by three parameters: \(|I|\), the number of
leader pure strategies, $|J|$, the number of follower pure strategies and $|K|$, the number of follower types. For the purpose of these experiments, we have considered instances where $|I| \in \{10, 20, 30\}$, $|J| \in \{10, 20, 30\}$ and $|K| \in \{2, 4, 6\}$. For each instance size, 5 instances are generated without variability in the payoff matrices and 5 are generated with variability. In total, we consider 135 instances without variability and 135 instances with variability. Performance profiles summarize our results, with respect to the following 4 measures: total running time employed to solve the integer problem, running time employed to solve the linear relaxation of the integer problem, total number of nodes explored in the branch and bound solving scheme and gap percentage at the root node. The gap percentage at the root node is calculated by comparing the optimal values of the formulation and of its LP relaxation:

$$\frac{v(F) - v(F^*)}{v(F^*)} \cdot 100.$$  

A performance profile graph plots the total percentage of problems solved for each value of these measures.

We study the behavior of $(D^2_{x,q,s,f})$, $(D^2_{x,q,f})$, $(DOBSS_{q,z,s})$, $(DOBSS_{q,z})$ and $(MIP-p-G_{q,z})$. Figures 1 and 2 compare the performance profiles when the payoff matrices are generated without variability and with variability, respectively.

We observe that the instances where variability is introduced in the payoff matrices solve faster than those where no variability is considered. When there is no variability, $(DOBSS_{q,z,s})$ and $(MIP-p-G_{q,z})$ are the two most competitive formulations. $(D^2_{x,q,s,f})$ can also be solved efficiently for the mid-range instances but slows down for the more difficult instances. Introducing variability in the payoff matrices, however, leads to a dominance of

![Figure 1: GSGs: $|I| \in \{10, 20, 30\}$, $|J| \in \{10, 20, 30\}$, $|K| \in \{2, 4, 6\}$—without variability.](image)
(MIP-\(p\)-G\(_{q,z}\)) with (DOBSS\(_{q,z,s}\)) coming in a close second and (D2\(_{x,q,s}\)) becoming noncompetitive for these instances. In what regards the time spent solving the linear relaxation of the problems, (MIP-\(p\)-G\(_{q,z}\)) is the formulation that is hardest to solve, this due to the fact that is has the most variables and constraints, \(O(|K||J|^2)\). On the other hand, (D2\(_{x,q,s,f}\)), that has the lightest LP relaxation, with \(O(|K||J|)\) variables and constraints, is the fastest.

With respect to the number of nodes and gap percentage our theoretical findings are corroborated: (MIP-\(p\)-G\(_{q,z}\)) is the tightest formulation and therefore uses the fewest nodes. The effect is further intensified when variability is introduced.

Table 2 summarizes the mean gap obtained across the instances solved. Finally, remark that the formulations obtained through Fourier-Motzkin, (D2\(_{x,q,f}\)) and (DOBSS\(_{q,z}\)), explore slightly less nodes in the branch and bound scheme than their counterparts, (D2\(_{x,q,s,f}\)) and (DOBSS\(_{q,z,s}\)), but because of the increase in the number of constraints the time to solve each linear relaxation increases. This increases the overall solution time of the Fourier-Motzkin formulations.

<table>
<thead>
<tr>
<th></th>
<th>(D2(_{x,q,s,f}))</th>
<th>(DOBSS(_{q,z,s}))</th>
<th>(MIP-(p)-G(_{q,z}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean gap % (no variability)</td>
<td>117.68</td>
<td>23.01</td>
<td>9.94</td>
</tr>
<tr>
<td>Mean gap % (with variability)</td>
<td>103.44</td>
<td>40.74</td>
<td>5.17</td>
</tr>
<tr>
<td>Total mean gap %</td>
<td>110.56</td>
<td>31.88</td>
<td>7.56</td>
</tr>
</tbody>
</table>

Table 2: Mean gap percentage recorded for GSG formulations.
5 Stackelberg security games-SSGs

In this section, we derive three SSG formulations: (ERASER\(_{c,q,s,f}\)), due to [Kiekintveld et al., 2009], and (SDOBSS\(_{q,y,s}\)) and (MIP-p-S\(_{q,y}\)). We derive these formulations by exploring the inherent link between the general setting, considered up to now and the security setting, defined in Section 2.2. In this setting, the defender pure strategies \(i \in I\) are the different ways in which up to \(m\) targets can be protected simultaneously. For this problem, we refer to \(i \in I\) as a set indicating which targets are covered by security resources. Recall that the payoff matrices of SSGs satisfy:

\[
R_{kj}^k = \begin{cases} 
D^k(j|c) & \text{if } j \in i \\
D^k(j|u) & \text{if } j \notin i
\end{cases} \quad (35)
\]

\[
C_{kj}^k = \begin{cases} 
A^k(j|c) & \text{if } j \in i \\
A^k(j|u) & \text{if } j \notin i
\end{cases} \quad (36)
\]

The payoff for the leader when he commits to a pure strategy \(i \in I\) and a follower of type \(k \in K\) responds by selecting strategy \(j \in J\) is either a reward if pure strategy \(i \in I\) allocates security coverage to attacked target \(j \in J\), or, a penalty if strategy \(i\) does not cover target \(j\). The same argument explains the link between payoffs for the attackers.

5.1 Stackelberg security games: single level formulations

The first formulation we derive is based on (D2\(_{x,q,s,f}\)). Consider (D2\(_{c,x,q,s,f}\)), an extended description of (D2\(_{x,q,s,f}\)) where we introduce the \(c\) variables through Constraint (13) (see Section 2.2). We further use relations (35) and (36) to adapt the payoff structure:

\[
(D2_{c,x,q,s,f})
\]

Max \( \sum_{k \in K} n_k f^k \) \hspace{1cm} (37)

s.t. \( \sum_{i \in I, j \in i} x_i = c_j \) \hspace{1cm} \( \forall j \in J \) \hspace{1cm} (38)

\( \sum_{j \in J} q_j^k = 1 \) \hspace{1cm} \( \forall k \in K \), \hspace{1cm} (39)

\( q_j^k \in \{0,1\} \) \hspace{1cm} \( \forall j \in J, \forall k \in K \), \hspace{1cm} (40)

\( \sum_{i \in I} x_i = 1 \), \hspace{1cm} (41)

\( x_i \geq 0 \) \hspace{1cm} \( \forall i \in I \) \hspace{1cm} (42)

\( 0 \leq s^k - A^k(j|c)c_j - A^k(j|u)(1-c_j) \leq (1-q_j^k) \cdot M \) \hspace{1cm} \( \forall j \in J, \forall k \in K \), \hspace{1cm} (43)

\( f^k \leq D^k(j|c)c_j + D^k(j|u)(1-c_j) + (1-q_j^k) \cdot M \) \hspace{1cm} \( \forall j \in J, \forall k \in K \), \hspace{1cm} (44)

\( s, f \in \mathbb{R}^K \).
This extended formulation is equivalent to \((D2_{x,q,s,f})\), because, even though they are defined in different spaces of variables, the value of their LP relaxations coincide.

The formulation above has a large number of non-negative variables since in the security setting, the set \(I\) of all defender pure strategies is exponential in the number of targets as it contains all subsets of at most \(m\) targets of \(J\) that the defender can protect simultaneously.

In order to avoid having exponentially many non-negative variables in our formulation, we project out variables \(x_i, \ i \in I\), from the formulation. Note that only Constraints (38), (41) and (42) involve said variables.

Proposition 3. Consider the following two sets:

\[
A = \text{Proj}_{c}\left\{ (x,c) \in \mathbb{R}^{|I|} \times \mathbb{R}^{|J|} : (38),(41),(42) \right\}
\]

\[
B = \left\{ c \in \mathbb{R}^{|J|} : \sum_{j \in J} c_j \leq m, \ c_j \in [0,1] \ \forall j \in J \right\}
\]

Then, \(A = B\).

Proof. Remark first that using Farkas’ Lemma [Farkas, 1902]:

\[
A = \left\{ c \in \mathbb{R}^{|J|} : \sum_{j \in J} \alpha_j c_j + \alpha_{|J|+1} \geq 0 \ \forall \alpha \in \mathbb{R}^{|J|+1} : \sum_{j \in J, j \in i} \alpha_j + \alpha_{|J|+1} \geq 0 \ \forall i \in I : |i| \leq m \text{ and } \alpha_{|J|+1} \geq 0 \right\},
\]

Thus \(A \subseteq B\). Indeed, the following 2\(|J|\) + 1 vectors in \(\mathbb{R}^{|J|+1}\):

\[
\forall j \in J, \ e^j \in \mathbb{R}^{|J|+1} : e^j_j = 1, \ e^j_k = 0 \ \forall k \in J : k \neq j \text{ and } e^j_{|J|+1} = 0,
\]

\[
\forall j \in J, \ f^j \in \mathbb{R}^{|J|+1} : f^j_j = -1, \ f^j_k = 0 \ \forall k \in J : k \neq j \text{ and } f^j_{|J|+1} = 1 \text{ and } g \in \mathbb{R}^{|J|+1} : g_j = -1 \ \forall j \in J \text{ and } g_{|J|+1} = m,
\]

satisfy \(\sum_{j \in J, j \in i} \alpha_j + \alpha_{|J|+1} \geq 0 \) and \(\alpha_{|J|+1} \geq 0\). Additionally, when we substitute the above vectors into the generic constraint defining \(A\), they yield all the constraints defining \(B\).

To show that \(A = B\), it remains to show that any other inequality

\[
\sum_{j \in J} \alpha_j c_j + \alpha_{|J|+1} \geq 0
\]

such that \(\alpha\) satisfies

\[
\sum_{j \in J, j \in i} \alpha_j + \alpha_{|J|+1} \geq 0 \ \forall i \in I : |i| \leq m \text{ and } \alpha_{|J|+1} \geq 0,
\]
is dominated by some nonnegative linear combination of the constraints defining $B$.

First, note that we can restrict our attention to constraints such that $\alpha_j \leq 0$ for all $j \in J$. If there exists $\hat{j} \in J$ such that $\alpha_{\hat{j}} > 0$, since $\alpha$ must satisfy (46) and $|i \setminus \{\hat{j}\}| \leq |i| \leq m$, it follows that $\bar{\alpha}$ with $\bar{\alpha}_{\hat{j}} = 0$ and $\bar{\alpha}_j = \alpha_j$ for all $j \in J \setminus \{\hat{j}\}$ also satisfies (46) and since $c \geq 0$, we have that

$$\sum_{j \in J} \bar{\alpha}_j c_j + \bar{\alpha}_{|J|+1} \leq \sum_{j \in J} \alpha_j c_j + \alpha_{|J|+1}.$$ 

Therefore, the constraint defined by $\alpha$ is dominated by the constraint defined by $\bar{\alpha}$. We thus distinguish two cases of $\alpha$ satisfying (46):

Case 1. $|\{j : \alpha_j < 0\}| = k \leq m$, and

Case 2. $|\{j : \alpha_j < 0\}| = k > m$.

In Case 1, by considering a linear combination of inequalities $c_j \leq 1$ for $1 \leq j \leq k$ with respective weights $-\alpha_j \geq 0$, we obtain that:

$$0 \leq \sum_{j=1}^{k} \alpha_j c_j - \sum_{j=1}^{k} \alpha_j \leq \sum_{j \in J} \alpha_j c_j + \alpha_{|J|+1},$$

since $\alpha_j = 0$ for all $j > k$ and $\alpha$ satisfies (46) for $i = \{1, \ldots, k\}$.

For Case 2, assume w.l.o.g that $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_k < 0$ and $\alpha_j = 0$ for all $j > k$. Then, build a linear combination of inequality $\sum_{j \in J} c_j \leq m$ with weight $-\alpha_m \geq 0$ and inequalities $c_j \leq 1$ for $1 \leq j \leq m$ with respective weights $\alpha_m - \alpha_j \geq 0$. The valid inequality thus obtained is:

$$0 \leq \sum_{j=1}^{m} \alpha_j c_j + \sum_{j=m+1}^{m} \alpha_m c_j - \sum_{j=1}^{m} \alpha_j \leq \sum_{j \in J} \alpha_j c_j + \alpha_{|J|+1},$$

since $\alpha$ satisfies (46) for $i = \{1, \ldots, m\}$. ■

Proposition 3 leads to the following formulation based on $(D_{c,x,q,s,f}^2)$:

$$\text{(ERASER}_{c,q,s,f})$$

Max $\sum_{k \in K} \pi^k f^k$ \hspace{1cm} (47)

s.t. $\sum_{j \in J} c_j \leq m,$

$0 \leq c_j \leq 1 \hspace{1cm} \forall j \in J,$

$\sum_{j \in J} q^k_j = 1 \hspace{1cm} \forall k \in K,$

$q^k_j \in \{0, 1\} \hspace{1cm} \forall j \in J, \forall k \in K,$

$0 \leq s^k - A^k(j|c)c_j - A^k(j|u)(1 - c_j) \leq (1 - q^k_j) \cdot M \hspace{1cm} \forall j \in J, \forall k \in K,$ \hspace{1cm} (48)
\[ f^k \leq D^k(j|c)c_j + D^k(j|u)(1 - c_j) + (1 - q^k_j) \cdot M \quad \forall j \in J, \forall k \in K, \quad (49) \]

\[ s, f \in \mathbb{R}^K. \]

The above formulation involves a polynomial number of variables and constraints and was presented in [Kiekintveld et al., 2009]. The next result is also an immediate consequence of Proposition 3.

**Corollary 2.** \( \text{Proj}_{c,q,s,f} \mathcal{P}(D^2_{c,x,q,s,f}) = \mathcal{P}(\text{ERASER}_{c,s,f}). \)

We now derive SSG formulations based on (DOBSS\(_{q,z,s}\)) and (MIP-\(p\)-G\(_{q,z}\)). We first present extended descriptions of both formulations by considering \( y^k_{ij} \) variables satisfying:

\[ y^k_{ij} = \sum_{i \in I : \ell \in i} z^k_{ij} \quad \forall j, \ell \in J, \forall k \in K. \quad (50) \]

We use (35) and (36) to adapt the payoffs to the security setting leading to:

**DOBSS\(_{q,z,y,s}\)**

\[
\begin{align*}
\text{Max} & \quad \sum_{j \in J} \sum_{k \in K} \{ \pi^k(D^k(j|c)y^k_{jj} + D^k(j|u)(q^k_j - y^k_{jj})) \} \\
\text{s.t.} & \quad \sum_{j \in J} z^k_{ij} = \sum_{j \in J} z^1_{ij} \quad \forall i \in I, \forall k \in K, \quad (52) \\
& \quad \sum_{i \in I : \ell \in i} z^k_{ij} = y^k_{ij} \quad \forall \ell, j \in J, \forall k \in K, \quad (53) \\
& \quad \sum_{i \in I} z^k_{ij} = q^k_j \quad \forall j \in J, \forall k \in K, \quad (54) \\
& \quad z^k_{ij} \geq 0 \quad \forall i \in I, \forall j \in J, \forall k \in K, \quad (55) \\
& \quad 0 \leq s^k - A^k(j|c) \sum_{j' \in J} y^k_{jj'} - A^k(j|u)(1 - \sum_{j' \in J} y^k_{jj'}) \leq (1 - q^k_j) \cdot M \quad \forall j \in J, \forall k \in K, \quad (56) \\
& \quad \sum_{j \in J} q^k_j = 1 \quad \forall k \in K, \quad (57) \\
& \quad q^k_j \in \{0, 1\} \quad \forall j \in J, \forall k \in K, \quad (58) \\
& \quad s \in \mathbb{R}^{|K|}. \quad (59)
\end{align*}
\]

**MIP-\(p\)-G\(_{q,z}\)**

\[
\begin{align*}
\text{Max} & \quad \sum_{j \in J} \sum_{k \in K} \pi^k(D^k(j|c)y^k_{jj} + D^k(j|u)(q^k_j - y^k_{jj})) \\
\text{s.t.} & \quad (52) - (55), (57) - (58) \\
& \quad A^k(j|c)y^k_{jj} + A^k(j|u)(q^k_j - y^k_{jj}) - A^k(\ell|c)y^k_{\ell j} - A^k(\ell|u)(q^k_j - y^k_{\ell j}) \geq 0 \quad \forall j, \ell \in J, \forall k \in K. \quad (60)
\end{align*}
\]
Further, consider the following constraint:

$$\sum_{j \in J} y_{\ell j}^k = \sum_{j \in J} y_{\ell j}^1 \quad \forall \ell \in J, \forall k \in K,$$

and let us define the following polyhedra $C$ and $D$:

$$C := \left\{ (q, z, y, s) \in [0, 1]^{K|J|} \times [0, 1]^{K|J|} \times [0, 1]^{K|J|^2} \times \mathbb{R}^{|K|} : (53) - (57), (59), (61) \right\}$$

$$D := \left\{ (q, z, y) \in [0, 1]^{K|J|} \times [0, 1]^{K|J|} \times [0, 1]^{K|J|^2} : (53) - (55), (57), (60), (61) \right\}$$

**Lemma 1.** $C \supseteq P(\text{DOBSS}_{q,z,y,s})$ and $D \supseteq P(\text{MIP-}p\text{-G}_{q,z,y})$

**Proof.** Consider Constraint (52) and sum over all $i \in I$ such that $\ell \in i$:

$$\sum_{i \in I} \sum_{j \in J} z_{ij}^k = \sum_{i \in I} \sum_{j \in J} z_{ij}^1 \quad \forall \ell \in J, \forall k \in K. \quad (62)$$

Applying (53) to (62) yields (61) and the result follows.

We shall now project the $z$ variables from the larger polyhedra $C$ and $D$. Said variables only appear in Constraints (53)-(55).

**Lemma 2.** Consider the following two sets:

$$\mathcal{X} = \text{Proj}_{q,y} \left\{ (q, z, y) \in \mathbb{R}^{K|J|^2 + |K||J| + |I||J||K|} : (53) - (55) \right\}$$

$$\mathcal{Y} = \left\{ (q, y) \in \mathbb{R}^{K|J|^2 + |K||J|} : \sum_{i \in I} \sum_{j \in J} y_{ij}^k \leq mq_j^k \forall j \in J, \forall k \in K,$$

$$0 \leq y_{\ell j}^k \leq q_j^k \forall j, \ell \in J, \forall k \in K \right\}$$

Then, $\mathcal{X} = \mathcal{Y}$.

**Proof.** Note that Constraints (53)-(55) can be treated independently for each $k \in K$ and each $j \in J$. First consider the case where $q_j^k = 0$ for $j \in J$ and $k \in K$. Constraint (54) then implies that for all $i \in I$, $z_{ij}^k = 0$ and Constraint (53) forces $y_{ij}^k = 0$ for all $\ell \in J$ and the result holds. For all $j \in J$, $k \in K$ such that $q_j^k \neq 0$, consider $x_i = z_{ij}^k / q_j^k$ and $c_\ell = y_{\ell j}^k / q_j^k$ and apply Proposition 3. The result follows.

Consider $\text{Proj}_{q,y,s} C$ and $\text{Proj}_{q,y} D$ as the feasible regions of the linear relaxations of two MILP formulations—$(\text{SDOBSS}_{q,y,s})$ and $(\text{MIP-}p\text{-}S_{q,y})$—where we maximize the objective.
function (51) under the additional requirement that the $q$ variables be binary. Hence, we present (SDOBSS$_{q,y,s}$), a security formulation based on (DOBSS$_{q,z,y,s}$),

$$(\text{SDOBSS}_{q,y,s})$$

$$\text{Max} \sum_{j \in J} \sum_{k \in K} \pi^k(D^k(j|c)y^k_{j,j} + D^k(j|u)(q^k_j - y^k_{j,j}))$$

s.t.

$$\sum_{j \in J} q^k_j = 1 \quad \forall k \in K \quad (63)$$

$$q^k_j \in \{0, 1\} \quad \forall j \in J, \forall k \in K, \quad (64)$$

$$\sum_{j \in J} y^k_{j,j} = \sum_{j \in J} y^1_{j,j} \quad \forall \ell \in J, \forall k \in K, \quad (65)$$

$$\sum_{\ell \in J} y^k_{\ell,j} \leq mq^k_j \quad \forall j \in J, \forall k \in K, \quad (66)$$

$$0 \leq y^k_{\ell,j} \leq q^k_j \quad \forall j, \ell \in J, \forall k \in K, \quad (67)$$

$$0 \leq s^k_j - A^k(j|c) \sum_{j' \in J} y^k_{j,j'} - A^k(j|u)(1 - \sum_{j' \in J} y^k_{j,j'}) \leq (1 - q^k_j) \cdot M \quad \forall j \in J, \forall k \in K. \quad (68)$$

$$s \in \mathbb{R}^{|K|}.$$

And we also present (MIP-p-S$_{q,y}$), a security formulation based on (MIP-p-G$_{q,z,y}$),

$$(\text{MIP-p-S}_{q,y})$$

$$\text{Max} \sum_{j \in J} \sum_{k \in K} \pi^k(D^k(j|c)y^k_{j,j} + D^k(j|u)(q^k_j - y^k_{j,j}))$$

s.t.

$$(63) - (67) \quad (69)$$

$$A^k(j|c)y^k_{j,j} + A^k(j|u)(q^k_j - y^k_{j,j}) - A^k(\ell|c)y^k_{\ell,j} - A^k(\ell|u)(q^k_j - y^k_{\ell,j}) \geq 0 \quad \forall j, \ell \in J, \forall k \in K. \quad (70)$$

The following corollaries are an immediate consequence of Lemmas 1 and 2.

**Corollary 3.** $\text{Proj}_{q,y,s}P(\text{DOBSS}_{q,z,y,s}) \subseteq P(\text{SDOBSS}_{q,y,s}).$

**Corollary 4.** $\text{Proj}_{q,y}P(\text{MIP-p-G}_{q,z,y}) \subseteq P(\text{MIP-p-S}_{q,y}).$

In addition, note that if we restrict (MIP-p-G$_{q,z,y}$) to a single type of follower, Constraint (52) disappears and one thus obtains the following corollary.

**Corollary 5.** $\text{Proj}_{q,y}P(\text{MIP-1-G}_{q,z,y}) = P(\text{MIP-1-S}_{q,y}).$

The above corollary immediately leads to the following theorem.
Theorem 2. \((\text{MIP-1-}S_{q,y})\) is a linear description of the convex hull of feasible solutions for the Stackelberg security game with a single type of attacker.

Proof. The result follows from Corollary 5 and from [Conitzer and Korzhyk, 2011] showing that \((\text{MIP-1-G}_{q,y})\) is a linear description for general games.

As in general games, we can use Fourier-Motzkin elimination on Constraints (48) and (68) to project out the \(s\) variables from formulations \((\text{ERASER}_{c,q,s,f})\) and \((\text{SDOBSS}_{q,y,s})\) respectively. This leads to the following two families of inequalities:

\[
(A^k(j|c) - A^k(j|u))c_j + (A^k(\ell|u) - A^k(\ell|c))c_\ell + A^k(j|u) - A^k(\ell|u) \leq (1 - q^k_\ell) \cdot M \quad \forall j, \ell \in J, \forall k \in K,
\]

\[
(A^k(j|c) - A^k(j|u) + (A^k(\ell|u) - A^k(\ell|c)) \sum_{h \in J} y^k_{jh} + A^k(j|u) - A^k(\ell|u) \leq (1 - q^k_\ell) \cdot M \quad \forall j, \ell \in J, \forall k \in K,
\]

Replacing Constraint (48) by (71) in \((\text{ERASER}_{c,q,s,f})\) and (68) by (72) in \((\text{SDOBSS}_{q,y,s})\) leads to \((\text{ERASER}_{c,q,f})\) and \((\text{SDOBSS}_{q,y})\).

In the same spirit as Proposition 1, we present the following proposition, establishing the tightest values for the big \(M\) constants in the formulations seen so far:

**Proposition 4.** The tightest values for the positive constants \(M\) are:

1. In (49), \(M = \max_{\ell \in J} \{D^k(\ell|c), D^k(\ell|u)\} - \min \{D^k(j|c), D^k(j|u)\}, \forall j, k \in K\).

2. In (48), (68), \(M = \max_{\ell \in J} \{A^k(\ell|c), A^k(\ell|u)\} - \min \{A^k(j|c), A^k(j|u)\}, \forall j, k \in K\).

3. In (71), (72), \(M = \max \{A^k(j|c), A^k(j|u)\} - \min \{A^k(\ell|c), A^k(\ell|u)\}, \forall j, \ell \in J, k \in K\).

### 5.2 Comparison of the formulations

First, we introduce an additional formulation which we denote by \((\text{SDOBSS}_{c,q,y,s,f})\). This formulation is equivalent to \((\text{SDOBSS}_{q,y,s})\), in the sense that the value of their LP relaxations coincide. In this formulation we introduce variables \(f^k\) for all \(k \in K\) to rewrite the objective function so that it matches (47). We also add variables \(c_\ell\) for all \(\ell \in J\) and rewrite Constraint (65) as \(\sum_{j \in J} y^k_{ij} = c_\ell\) for all \(\ell \in J\) and all \(k \in K\). Using this last condition we can simplify (68) to (48). The formulation \((\text{SDOBSS}_{c,q,y,s,f})\) is as follows.

\[
(\text{SDOBSS}_{c,q,y,s,f}) \quad \text{Max} \quad \sum_{k \in K} \pi^k f^k
\]

\[
\text{s.t.} \quad (63), (64), (66) - (68),
\]

\[
f^k = \sum_{j \in J} \{y^k_{ij}(D^k(j|c) - D^k(j|u)) + q^k_j D^k(j|u)\} \quad \forall k \in K
\]
\[
\sum_{j \in J} y_{\ell j}^k = c_\ell \quad \forall \ell \in J, \forall k \in K, \quad (74)
\]

Note that
\[
P(\text{ERASER}_{c,q,f}) = \text{Proj}_{c,q,f} P(\text{ERASER}_{c,q,s,f})\]
and
\[
P(\text{SDOBSS}_{q,y}) = \text{Proj}_{q,y} P(\text{SDOBSS}_{q,y,s}).
\]

**Proposition 5.** \(\text{Proj}_{c,q,s,f} P(\text{SDOBSS}_{c,q,y,s,f}) \subseteq P(\text{ERASER}_{c,q,s,f})\). Further, there exist instances for which the inclusion is strict.

**Proof.** The projection of \(P(\text{SDOBSS}_{c,q,y,s,f})\) onto the \((c,q,s,f)\)-space is obtained by applying Farkas’ Lemma. Constraints (66)-(67) and (73)-(74) are the only ones involving variables \(y_{\ell j}^k\) and are separable by \(k \in K\). For a fixed \(k \in K\), the projection is given by:

\[
A_k = \{(c,q,f) : \alpha(f^k - \sum_{j \in J} D^k(j|u)q_j^k) + \sum_{\ell \in J} \beta_\ell c_\ell + m \sum_{j \in J} \gamma_j q_j^k + \sum_{j \in J} \sum_{\ell \in J} \delta_{\ell j} q_j^k \geq 0
\]

\[
\forall (\alpha, \beta, \gamma, \delta) : \gamma, \delta \geq 0, \beta_\ell + \gamma_j + \delta_{\ell j} \geq 0 \forall \ell, j \in J : \ell \neq j, \text{ and}
\]

\[
\alpha(D^k(j|c) - D^k(j|u)) + \beta_j + \gamma_j + \delta_{\ell j} \geq 0 \forall j \in J\}
\]

(75)

Consider, for each \(k \in K\), the following set \(B_k\):

\[
B_k = \{(c,q,f) : c_\ell \leq \sum_{j \in J} q_j^k, \quad \forall \ell \in J, \quad (76)
\]

\[
c_\ell \geq 0, \quad \forall \ell \in J, \quad (77)
\]

\[
\sum_{\ell \in J} c_\ell \leq m \sum_{j \in J} q_j^k, \quad \forall \ell \in J, \quad (78)
\]

\[
f^k \leq c_j(D^k(j|c) - D^k(j|u)) + \sum_{\ell \in J} \ell \neq j \sum_{\ell \in J} \ell \neq j q_j^k D^k(\ell|c) + q_j^k D^k(j|u) \quad \forall j \in J, \quad (79)
\]

\[
q_j^k \geq 0 \quad \forall j \in J, \forall k \in K.\}
\]

Let us see that \(A_k \subseteq B_k\) for all \(k \in K\). First note that if we set \(\alpha = 0\), the following definitions of the parameters \(\beta, \gamma\) and \(\delta\) comply with the conditions in (75):

\[
\beta = e^h, \gamma = \{0\}_{j \in J}, \delta = \{0\}_{\ell,j \in J}, \forall h \in J,
\]

\[
\beta = -e^\ell, \gamma = \{0\}_{j \in J}, \delta = \{1\}_{\ell \in J}, \forall \ell \in J,
\]

\[
\beta = \{-1\}_{\ell \in J}, \gamma = \{1\}_{j \in J}, \delta = \{0\}_{\ell,j \in J},
\]

\[
\beta = \{0\}_{\ell \in J}, \gamma = \{0\}_{j \in J}, \delta_1 = \{e^j\}, \forall j \in J.
\]
Substituting these valid parameters in the generic constraint in $A^k$, produces all of the constraints in $B^k$ except (79). Further, for a fixed $j \in J$, consider $\alpha = -1$, $\beta_\ell = 0$ and $\gamma_\ell = \frac{1}{m}(D^k(\ell|c) - D^k(\ell|u))$ for all $\ell \in J$ such that $\ell \neq j$, $\beta_j = D^k(j|c) - D^k(j|u)$ and $\gamma_j = 0$. Finally, set $\delta_\ell = 0$ for all $\ell, j \in J$. This definition of parameters is valid as it satisfies the conditions in (75). Substituting in the generic constraint in $A^k$ yields (79).

It remains to show that for all $k \in K$, Constraint (79) implies (49) for the tight value of $M$ shown in Proposition 4. The implication holds because

$$
\sum_{\ell \in J: \ell \neq j} q^k_{\ell} D^k(\ell|c) \leq \max_{\ell \in J} \{D^k(\ell|c)\} \sum_{\ell \in J: \ell \neq j} q^k_{\ell} \left(1 - q^k_{\ell}\right) \max_{\ell \in J} \{D^k(\ell|c)\} \quad \forall j \in J, \forall k \in K.
$$

Hence, $\text{Proj}_{c,q,s,f} \mathcal{P}(\text{SDOBS}\text{s}_{c,q,y,s,f}) \subseteq \mathcal{P}(\text{ERASER}_{c,q,s,f})$. To show that the inclusion may be strict, consider the following example where $m = 1$, $|J| = 3$ and $|K| = 1$. Let the reward and penalty matrices for the defender and attacker be $D(\cdot|c) = [1, 0, 0]$, $D(\cdot|u) = [0, 0, 0]$, $A(\cdot|c) = [0, 0, 0]$ and $A(\cdot|u) = [0, 0, 0]$. Consider the point defined by $q = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$, $c = (1, 0, 0)^t$, $s = 10$ and $f = 2/3$. Such a point is feasible for $(\text{ERASER}_{c,q,s,f})$ but violates Constraint (79) for $j = 2$ and is therefore infeasible for $\text{Proj}_{c,q,f,s} \mathcal{P}(\text{SDOBS}\text{s}_{c,q,y,s,f})$.

Based on Theorem 1 we can present the following theorem comparing the polyhedra $\mathcal{P}(\text{MIP}-p\text{-S}_{q,y})$ and $\text{Proj}_{q,y} \mathcal{P}(\text{SDOBS}\text{s}_{q,y,s})$:

**Theorem 3.** $\mathcal{P}(\text{MIP}-p\text{-S}_{q,y}) \subseteq \mathcal{P}(\text{SDOBS}\text{s}_{q,y}) = \text{Proj}_{q,y} \mathcal{P}(\text{SDOBS}\text{s}_{q,y,s})$.

**Proof.** The inclusion is a consequence of Theorem 1, the relations between the payoffs described in (35) and (36) and the relation between the $z$ and $y$ variables described in (50). To show that the inclusion may be strict, consider the following game. We set $m = 2$, $|J| = 2$ and $|K| = 1$. The reward and penalty payoff matrices for both the defender and the attacker are given by $D(\cdot|c) = [1, 0]$, $D(\cdot|u) = [0, 0]$, $A(\cdot|c) = [0, 0]$ and $A(\cdot|u) = [0, 1]$. Additionally, the point with coordinates

$$
c^t = (1/2, 1/2), \quad q^t = (1/2, 1/2) \quad \text{and} \quad y^k = \begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}
$$

has an objective value of 1/4 and is a valid feasible solution of $\mathcal{P}(\text{SDOBS}\text{s}_{q,y})$. However, it is not feasible in $\mathcal{P}(\text{MIP}-p\text{-S}_{q,y})$ as it does not verify Constraint (70) for $j = 1$ and $\ell = 2$.

Remark that $(\text{MIP}-p\text{-S}_{q,y})$ can be obtained by applying RLT [Sherali and Adams, 1994] to $(\text{SDOBS}\text{s}_{q,y})$. Multiplying both sides of Constraint (71) by variable $q^k_{\ell}$ and noticing that $q^k_{\ell}(1 - q^k_{\ell}) = 0$, since $q^k_{\ell}$ is binary, one obtains a constraint that once linearized, introducing variables $y^k_{\ell j}$, yields (70). Since $(\text{ERASER}_{c,q,s,f})$ and $(f\text{-SDOBS}_{c,q,s,f})$ and $(\text{SDOBS}\text{s}_{q,y})$ and $(\text{MIP}-p\text{-S}_{q,y})$ have the same objective function, the following corollary holds.
Corollary 6. \( v(MIP-p-S_{q,y}) \leq v(SDOBSS_{q,y}) = v(SDOBSS_{c,q,s,f}) \leq v(ERASER_{c,q,s,f}) \).

6 Computational experiments for SSGs

Our security experiments are run on randomly generated instances. For each instance, four payoff matrices have to be generated that satisfy \( D^k(\cdot|c) \geq D^k(\cdot|u) \) and \( A^k(\cdot|u) \geq A^k(\cdot|c) \).

We consider two ways of generating these matrices. First, we generate matrices where the values for the penalty matrices \( (D^k(\cdot|u) \) and \( A^k(\cdot|c) \)) are randomly generated between 0 and 5 and all values for the reward matrices \( (D^k(\cdot|c) \) and \( A^k(\cdot|u) \)) are randomly generated between 5 and 10. We shall refer to these as matrices with no variability. Second, we consider an alternative where 90% of the values for the penalty matrices are randomly generated between 0 and 5 (between 5 and 10 for the reward matrices) and 10% of the values for the penalty matrices are randomly generated between 0 and 50 (between 50 and 100 for the reward matrices). We refer to these as matrices with variability. We use a solution limit of 3 hours.

A Stackelberg security game instance is defined by \(|J|\), the number of targets, \(|K|\) the number of attacker types and \(m\), the number of security resources available to the defender. Recall from the computational experiments for GSGs that using payoff matrices with variability amounts to endowing the game with more structure, thus making it somewhat easier to solve. We have encountered the same phenomenon in SSGs. For games whose payoff matrices have variability, we have considered \(J = \{30, 40, 50, 60, 70\}, \ K = \{6, 8, 10, 12\}\) and we have allowed \(m\) to be either 25%, 50% or 75% of the number of targets. For games whose payoff matrices don’t have variability we have had to be less ambitious in order to solve all instances to optimality within the stipulated time limit and have considered \(J = \{10, 20, 30, 40, 50\}, \ K = \{2, 4, 6, 8\}\) while still considering \(m\) to be either 25%, 50% or 75% of the number of targets. In either case, for each instance size we generate 5 random instances as described above. In total, we consider 300 randomly generated instances.

We study the behavior of \( (ERASER_{c,q,s,f}) \), \( (SDOBSS_{q,y,s}) \) and \( (MIP-p-S_{q,y}) \). For the sake of clarity we no longer consider the Fourier-Motzkin formulations \( (ERASER_{c,q,f}) \) and \( (SDOBSS_{q,y}) \). Performance-wise, \( (ERASER_{c,q,s,f}) \) and \( (SDOBSS_{q,y,s}) \) compare to their Fourier-Motzkin formulations in a similar way to how \( (D2_{x,q,s,f}) \) and \( (DOBSS_{q,z,s}) \) compared to theirs in Section 4. We plot performance profile graphs in Figures 3 and 4.

Remark that for the experiments with variability, \( (ERASER_{c,q,s,f}) \) is the fastest formulation for most of the instances. However, we see that for the more difficult instances, its solution time increases significantly, almost surpassing the solution time of \( (MIP-p-S_{q,y}) \). This indicates that for these instances \( (ERASER_{c,q,s,f}) \) ceases to be competitive and \( (MIP-\)
Figure 3: SSGs: $K = \{6, 8, 10, 12\}, J = \{30, 40, 50, 60, 70\}$–with variability

Figure 4: SSGs: $K = \{2, 4, 6, 8\}, J = \{10, 20, 30, 40, 50\}$–without variability
As for the instances whose payoff matrices have no variability, and are thus harder to solve, we observe that (ERASER_{c,q,s,f}) outperforms the running time of the other two formulations for 80% of the instances. However, for the most difficult instances, (MIP-p-S_{q,y}) is faster than the other two formulations. For the last 5% of the instances, (ERASER_{c,q,s,f}) is the worst formulation. In terms of size of the formulations, (ERASER_{c,q,s,f}) is the formulation with the least number of constraints and variables: \( O(|\mathcal{J}||\mathcal{K}|) \). Observe that (MIP-p-S_{q,y}) and (SDOBSS_{q,y,s}) have \( O(|\mathcal{J}|^2|\mathcal{K}|) \) constraints and variables. Thus, these formulations have significantly heavier LP relaxations and thus take longer time to solve than (ERASER_{c,q,s,f}) does. However, Figures 3 and 4 confirm our theoretical findings: (MIP-p-S_{q,y}) has the tightest LP relaxation and this translates into a clear dominance with respect to node usage in the B&B solving scheme.

In the above results we have observed a trend that indicates that for difficult instances, particularly in the case of payoff matrices with no variability, one could expect (ERASER_{c,q,s,f}) and (SDOBSS_{q,y,s}) to perform very poorly compared to (MIP-p-S_{q,y}). To analyze this, we consider instances where the payoff matrices have no variability and where \( \mathcal{K} = \{6, 8, 10, 12\} \), \( \mathcal{J} = \{30, 40, 50, 60, 70\} \) and \( m \) is 25%, 50% and 75% of the targets. We generate 5 random instances for each size. In addition, for practical reasons, we consider a time limit of 30 minutes. The computational results for these instances are shown in Figure 5.

Note that (MIP-p-S_{q,y}) is able to solve 95% of the 300 instances within the stipulated time limit, outperforming (SDOBSS_{q,y,s}) and (ERASER_{c,q,s,f}) which are only able to solve

Figure 5: SSGs: \( \mathcal{K} = \{6, 8, 10, 12\} \), \( \mathcal{J} = \{30, 40, 50, 60, 70\} \)--without variability
56% and 45% of the instances, respectively, within the same time frame. For the 45% of instances which can be solved by the three formulations, we observe that (MIP-\(p\)-\(S\)) offers a much tighter gap percentage than the other two formulations. Because of this, the node usage in the branch and bound scheme is significantly smaller in (MIP-\(p\)-\(S\)) compared to (ERASER_{c,q,s,f}) and (SDOBSS_{q,y,s}).

Table 3 records the mean gap percentage across all the instances for the three formulations under study. Observe that (MIP-\(p\)-\(S\)) is significantly tighter than the LP relaxations of the other formulations. We may thus conclude that for the payoff matrices without variability, (MIP-\(p\)-\(S\)) is the fastest formulation for the most difficult instances. (ERASER_{c,q,s,f}) is the fastest formulation when we endow the security game with further structure by allowing matrices to experience variability. Even then, (ERASER_{c,q,s,f}) looses ground to (MIP-\(p\)-\(S\)). This is due to the fact that (MIP-\(p\)-\(S\)) has the tightest LP relaxation. The quality of the upper bound obtained from (MIP-\(p\)-\(S\)) translates into a smaller B&B tree and this translates into reaching optimality of the integer problem faster in many cases.

### Table 3: Mean gap percentage recorded for SSG formulations.

<table>
<thead>
<tr>
<th></th>
<th>(ERASER_{c,q,s,f})</th>
<th>(SDOBSS_{q,y,s})</th>
<th>(MIP-(p)-(S))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean gap % (no variability)</td>
<td>241.26</td>
<td>38.87</td>
<td>3.09</td>
</tr>
<tr>
<td>Mean gap % (with variability)</td>
<td>168.37</td>
<td>18.66</td>
<td>0.35</td>
</tr>
<tr>
<td>Total mean gap %</td>
<td>204.82</td>
<td>28.76</td>
<td>1.72</td>
</tr>
</tbody>
</table>

7 Conclusions and future work

In this paper we consider Stackelberg games in two different settings. We first analyze the general Stackelberg setting, which models a hierarchical competitive game between different agents, and the specific Stackelberg security setting, where an agent must secure subsets of targets from attackers.

In the general setting, we have studied known MILP formulations and have ordered them with respect to the strength of their linear relaxations. We have presented a formal theoretical link between GSG formulations and SSG formulations involving the projection of variables. Exploiting this link has allowed to i) derive a new SSG MILP formulation (MIP-\(p\)-\(S\)); and ii) extend our study of GSG formulations to SSG formulations, leading to a ranking of the security formulations with respect to the strength of their linear relaxations, where (MIP-\(p\)-\(S\)) has been shown to be the strongest SSG formulation. Further, we have shown its single type of attacker restriction, (MIP-1-\(S\)), to be an ideal formulation.

Our computational studies have shown that (MIP-\(p\)-\(G\)) and (MIP-\(p\)-\(S\)), the tightest formulations in each setting, are highly competitive with respect to solving time. Further, in the case of (MIP-\(p\)-\(S\)), we have seen it scales significantly better than competing for-
mulations when tackling instances with no variability in their payoff structure. (MIP-p-S) represents a significant theoretical and practical improvement over previously existing SSG formulations.

However, the obvious bottleneck, at this time, is solving the tighter but significantly heavier LP relaxations provided by (MIP-p-Gq,z) and (MIP-p-Sq,y). The main challenge is to provide an efficient way of solving these tight formulations. It is our contention that this can be done by exploiting the inherent problem structure in the Stackelberg paradigm to develop either decomposition or cutting plane approaches.

Finally, to the best of our knowledge, literature concerning heuristics and meta-heuristics specific to the security domain is rather scarce. We believe that contributing heuristics to this domain could be a fruitful avenue of research to follow.

Acknowledgements

The first author wishes to acknowledge the FNRS for funding his PhD research through a FRIA grant. This work is also partially supported by the Interuniversity Attraction Poles Programme P7/36 “COMEX” initiated by the Belgian Science Policy Office.

References


