Coordinated defender strategies for border patrols

Víctor Bucarey, Carlos Casorrán, Martine Labbé, Fernando Ordóñez, Oscar Figueroa

Submitted to *Operations Research*
manuscript (Please, provide the manuscript number!)

Coordinated defender strategies for border patrols

Víctor Bucarey
Departamento de Ingeniería Industrial, Universidad de Chile, Santiago, Chile, vbucarey@ing.uchile.cl

Carlos Casorrán
Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium. ccasorra@ulb.ac.be

Martine Labbé
Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium. mlabbe@ulb.ac.be

Fernando Ordoñez
Departamento de Ingeniería Industrial, Universidad de Chile, Santiago, Chile, fordon@dii.uchile.cl

Oscar Figueroa
Carabineros de Chile, Santiago, Chile., oscar.figueroa@carabineros.cl

An effective patrol of a large area can require the coordinated action of diverse security resources. In this work we formulate a Stackelberg Security game that coordinates such resources in a border patrol problem. In this security domain, resources from different precincts have to be paired to conduct patrols in the border due to logistic constraints. Given this structure the set of pure defender strategies is of exponential size. We describe the set of mixed strategies using a polynomial number of variables but exponentially many constraints that come from the matching polytope. We then include this description in a mixed integer formulation to compute the Strong Stackelberg Equilibrium efficiently with a branch and cut scheme. Since the optimal patrol solution is a probability distribution over the set of exponential size, we also introduce an efficient sampling method that can be used to deploy the security resources every shift. Our computational results evaluate the efficiency of the branch and cut scheme developed and the accuracy of the sampling method. We show the applicability of the methodology by solving a real world border patrol problem.

Key words: Stackelberg games, Matching polyhedra, Security application

1. Introduction

Security is a worldwide concern that can require the coordinated use of diverse types of resources. For instance, a police force can use patrol cars, motorcicles, police on horse, on foot, or by helicopter to patrol a city. These different types of resources have different capabilities (patrolling range, ability to move through different terrain, ability to interact with the population) and a coordinated use of these specialized resources is used in providing an overall security level. Pooling resources from different units also helps provide preventive patrols without burdening any single unit.

Securing national borders to defend from the illegal movement of contraband, drugs and people is a natural concern of nations. The European Union (EU) created the European Border and Coast Guard in October 2016 in response to increases in migrant flows into the EU Council of the EU (2016). In the United States the Department of Homeland Security states as a primary objective that of "protecting [the] borders from the illegal movement of weapons, drugs, contraband, and people, while promoting lawful entry and exit" claiming it is "essential to homeland security, economic prosperity, and national sovereignty" Department of Homeland Security (2017). The European Border and Coast Guard lists as one of its prime objectives "organizing joint operations and rapid border interventions to strengthen the capacity of the member states to control the external borders, and to tackle challenges at the external border resulting from illegal immigration or cross-border crime". Such operations involve different resources possibly from different locations in a joint effort to coordinate a global border plan. It is crucial to balance the effectiveness of a global plan with the cost and difficulty of locally coordinating resources in undermanned areas.

In this paper we consider the operational problem of patrolling the border in the presence of strategic adversaries that seek to trespass it taking into account the defender patrols. We model the situation as a Stackelberg game where the defender acts as the leader executing a preventive border patrol, which is observed prior to the optimal response by the strategic adversary, which acts as the follower. Due to the size of the border, the defender coordinates local resources to achieve a global defender strategy. Motivated by a real border patrol problem, we consider that resources from adjacent precincts are paired together to patrol a location in their area.

Optimization and mathematical models have been used in diverse homeland security applications to efficiently assign limited security resources to defend systems and infrastructure from attack Wright et al. (2006). Previous work that addresses operational decisions in homeland security applications include Dreiding and McLay (2013) and Yan and Nie (2016) that consider optimization models for container screening and port security respectively. Passenger screening decisions in airports is addressed in Nie et al. (2009) by an integer optimization model that uses risk level grouping and in McClay et al. (2010) where adaptive risk-based screening policies are determined via Dynamic Programming. Models that explicitly take into account the strategic nature of the adversary lead to game theory models, which include network interdiction problems, Dimitrov and Morton (2012), and patrolling games over networks, Alpern et al. (2011). In particular, in Papadaki et al. (2016), the authors determine the optimal patroling strategy on a border by considering a zero sum simultaneous game over a line network.

Stackelberg Games, introduced by von Stackelberg (2011), are an example of bilevel optimization programs, Bracken and McGill (1973), where top level decisions are made by a player – the leader – that takes into account an optimal response – of a follower – to a nested optimization problem. Recent research has used Stackelberg games to model and provide implementable defender strategies in real life security applications. In these applications, a defender aims to defend targets from a strategic adversary by deploying limited resources to protect them. The defender deploys resources to maximize the expected utility, anticipating that the adversary attacks a target that maximizes its own utility. Examples of Stackelberg security games applications include assigning Federal Air Marshals to transatlantic flights, Jain et al. (2010b), determining U.S. Coast Guard patrols of port infrastructure Shieh et al. (2012), police patrols to prevent fare evasion in public transport systems Yin et al. (2012), as well as protecting endangered wildlife Yang et al. (2014).

One of the challenges that has to be addressed in solving these Stackelberg games for real-world security applications is problem size. When the defender action is to allocate limited resources to various targets, the set of possible defender actions can be quite large, i.e the different combinations in which the resources can be assigned to the targets. The size of this set of actions is exponential in the number of resources. In Kiekintveld et al. (2009) a relaxation of the Stackelberg security game is formulated which determines the frequency with which each target is protected. This polynomial formulation (in the number of targets and security resources) is shown to be exact when there are no constraints on what constitutes a feasible defender action, but it is only an approximation in the general case. Decomposition approaches have been developed for the general case, when feasible defender actions satisfy additional constraints. For instance a column generation approach is introduced in Jain et al. (2010a) and a cutting plane approach in Yang et al. (2013). A branch and cut approach based on Benders' decomposition is introduced in Yin and Tambe (2012). A specialized decomposition method that exploits specific problem structure was introduced in Hochbaum et al. (2014).

To represent the coordination in our border patrol problem, we consider a graph composed of nodes for the police precincts and edges that determine if two precincts can be paired. Within each precinct there is a set of locations, or targets, that can be patrolled. In this case the set of pure strategies corresponds to the set of all matchings of size m together with the corresponding possible targets that are protected. This set increases exponentially as the instance grows. We address this computational challenge by introducing compact formulations and using cut a branch and cut approach to to efficiently solve these problems.

The contributions of this work include the introduction of a mixed integer optimization formulation for Stackelberg security games that allow for coordinated defender strategies. To address the exponential problem size we introduce equivalent compact formulations that have a polynomial number of variables but exponentially many matching constraints. Furthermore, to efficently solve these compact formulations, we adapt the Padberg and Rao (1982) branch and cut scheme. To execute the optimal defender strategies, it is necessary to sample from the exponential size defender strategy space. We present both an exact, column generation based, approach to sample from the defender strategy space and an approximate method. Our computational results explore the efficiency of solving the Stackelberg game for coordinated defender strategies and the accuracy of the approximate defender strategy sampling method. An early version of this work appeared in the conference Bucarey et al. (2017).

The structure of the paper is as follows: the notation used and problem formulations are presented in the next Section 2. In Section 3 we prove the correctness of the compact formulations showing they are equivalent to Stackelberg security games when the defender strategies require pairing of resources. Here we also outline the branch and cut algorithm used to solve these formulations. In Section 4 we discuss methods to sample from the exponential size defender strategy space. We present both an exact method and a computationally efficient approximate sampling method. Section 5, presents computational experiments to evaluate the difficulty of solving the different problem formulations and of the sampling methods. Furthermore we present in this section results on using this problem formulation on a case study constructed from a realistic border patrol instance. Finally, we present our conclusions and discuss future work in Section 6.

2. Problem Formulation and Notation

In this section we introduce the notation used and the general framework of Stackelberg security games (SSG). Then, we present an SSG that seeks to select coordinated defender strategies given heterogeneous resources.

2.1. Stackelberg Security games

We consider a general Bayesian Stackelberg game, where a leader is facing a set K of followers, as introduced in Paruchuri et al. (2008). In this setting, the leader commits to a payoff-maximizing strategy, anticipating that every follower knows the leader's choice and responds by selecting their own payoff-maximizing strategy. The solution concept we follow in this game is the *strong Stackelberg equilibrium*, introduced in Leitman (1978). In this equilibrium followers select a best response that favors the leader when they are indifferent between several optimal strategies.

In this model the leader knows the probability π_k of facing follower $k \in K$. We denote by I the finite set of pure strategies for the leader and by J be the finite set of pure strategies for each of the followers. A mixed strategy for the leader consists in a vector $\mathbf{x} = (x_i)_{i\in I}$, such that x_i is the probability that the leader plays pure strategy i. Analogously, a mixed strategy for follower $k \in K$ is a vector $\mathbf{q}^k = (q^k_j)_{j\in J}$ such that q^k_j is the probability that follower k plays pure strategy j. The payoffs are represented by the payoff matrices $(R^k, C^k)_{k\in K}$, with $R^k, C^k \in \mathbb{R}^{|I|\times|J|}$. The R^k_{ij} (C^k_{ij}) entry gives the reward for the leader (k-th follower) of taking the leader action i and the k-th follower action j. With these payoff matrices, given a mixed strategy $\mathbf{x}$ for the leader and strategy $\mathbf{q^k}$ for follower k, the expected utility for follower k is given by $\sum_{i\in I}\sum_{j\in J} C^k_{ij} x_i q^k_j$ while the expected utility for the leader is given by $\sum_{k\in K}\pi_k \sum_{i\in I}\sum_{j\in J} R^k_{ij} x_i q^k_j$.

The problem of finding the strong Stackelberg equilibrium can be formulated as the following bilevel bilinear problem, see Paruchuri et al. (2008):

$$\max_{x,q} \quad \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} \pi^k R^k_{ij} x_i q^k_j \tag{1}$$

$$\text{s.t.} \quad \mathbf{x}^\top \mathbf{1} = 1,\, \mathbf{x} \geq 0, \tag{2}$$

$$\mathbf{q}^k \in \arg\max \left\{ \sum_{i\in I}\sum_{j\in J} C^k_{ij} x_i r^k_j \;:\; {\mathbf{r}^k}^\top \mathbf{1} = 1,\, \mathbf{r}^k \geq 0 \right\} \qquad \forall k \in K. \tag{3}$$

The first level problem–or leader's problem– optimizes the leader's expected reward in (1) by selecting the mixed strategy (2) and taking into account the optimal follower response (3). The second level problems–or follower problems–in (3) requires each follower $k \in K$ to commit to a mixed strategy $\mathbf{q}^k$ which is a best response to the leader's strategy, $\mathbf{x}$, in that it maximizes follower k's payoff. Note that, since the problem in (3) is a linear optimization problem on the unit simplex, for any leader strategy x and any $k \in K$, there is a best response to follower k's problem that is a pure strategy, that is a vector $q^k \in \{0,1\}^{|J|}$ such that $\sum_{j\in J} q^k_j = 1$.

This observation has been used to propose mixed integer linear programming (MILP) formulations of Stackelberg games. For instance (D2) is formulated in Paruchuri et al. (2008) by linearizing the objective and the optimality constraint with big M constraints:

$$\text{(D2)} \quad \max_{x,q,s,f} \quad \sum_{k\in K} \pi^k f^k \tag{4}$$

$$\text{s.t.} \quad \mathbf{x}^\top \mathbf{1} = 1,\ \mathbf{x} \geq 0, \tag{5}$$

$$\mathbf{q}^{k\top} \mathbf{1} = 1,\ \mathbf{q}^k \in \{0,1\}^{|J|} \qquad \forall k \in K \tag{6}$$

$$f^k \leq \sum_{i \in I} R^k_{ij} x_i + M(1 - q^k_j) \qquad \forall j \in J,\ \forall k \in K, \tag{7}$$

$$0 \leq s^k - \sum_{i \in I} C^k_{ij} x_i \leq M(1 - q^k_j) \qquad \forall j \in J,\ \forall k \in K. \tag{8}$$

The (MIP-G) equivalent formulation, introduced in Casorrán et al. (2017), is constructed by using the linearizing variable $h^k_{ij} = x_i q^k_j$ and projecting constraints (7) and (8):

$$\text{(MIP-G)} \quad \max_{h,q} \quad \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \pi^k R^k_{ij} h^k_{ij} \tag{9}$$

$$\text{s.t.} \quad \sum_{j \in J} h^k_{ij} = \sum_{j \in J} h^1_{ij} \qquad \forall i \in I, \forall k \in K \tag{10}$$

$$\sum_{i \in I} h^k_{ij} = q^k_j,\ \mathbf{h} \geq 0 \qquad \forall j \in J, \forall k \in K \tag{11}$$

$$\mathbf{q}^{k\top} \mathbf{1} = 1,\ \mathbf{q}^k \in \{0,1\}^{|J|} \qquad \forall k \in K \tag{12}$$

$$\sum_{i \in I} \left(C^k_{ij} - C^k_{il}\right) h^k_{ij} \geq 0 \qquad \forall j, l \in J,\ \forall k \in K, \tag{13}$$

While (MIP-G) has more variables and constraints, Casorrán et al. (2017) shows that it provides a tighter linear optimization relaxation. We point the reader to that work for a detailed study of the strength and efficiency of these formulations.

In addition, an SSG assumes two things: 1) that the leader (defender) has to protect n targets by deploying resources capable of protecting up to $m < n$ of them. The follower (attacker) selects one target to attack. 2) the rewards depend only on whether the target attacked is protected or not. In this case the set of follower strategies is $J = \{1, \ldots, n\}$ and the strategy set for the defender consists of all feasible deployments that protect up to m targets simultaneously. That is $I \subset \{K \mid K \subset J,\ |K| \leq m\}$ since there might be logistical constraints that prohibit all combinations of m targets to be protected simultaneously. Let $j \in i$ denote the condition that defender strategy i patrols target j. We express the rewards in SSG as follows:

$$R^k_{ij} = \begin{cases} D^k(j|p) \text{ if } j \in i \\ \\ D^k(j|u) \text{ if } j \notin i \end{cases} \quad (14) \qquad\qquad C^k_{ij} = \begin{cases} A^k(j|p) \text{ if } j \in i \\ \\ A^k(j|u) \text{ if } j \notin i. \end{cases} \quad (15)$$

Here $D^k(j|p)$ ($D^k(j|u)$) denotes the utility of the defender when target $j \in J$ is protected (unprotected) and is attacked by attacker $k \in K$. Similarly, the utility of an attacker of type $k \in K$ when attacking a protected (an unprotected) target $j \in J$ is denoted by $A^k(j|p)$ ($A^k(j|u)$).

This reward matrix structure is exploited in Kiekintveld et al. (2009) to build a compact representation of the problem. The approach seeks to avoid using the entire leader strategy space I by introducing for each target $j \in J$ a coverage variable c_j that represents the frequency of coverage for that target, that is $c_j = \sum_{i \in I: j \in i} x_i$ the sum of probabilities over strategies that protect j.

2.2. Resource Combination in a SSG

We now formulate the SSG problem were the defender first teams up resources from different precincts to form m combined patrols and then decides where to deploy them. Let V be the set of police precincts. We let $E \subset V \times V$ be the set of edges representing the set of possible precinct pairings, forming an adjacency graph $G = (V, E)$. We denote by $\delta(v) \subset E$ the set of edges incident to precinct $v \in V$, similarly for any $U \subset V$, $\delta(U) \subset E$ denotes the edges between U and $V \setminus U$, and $E(U) \subset E$ denotes the edges between precincts in U. We can then represent the possible combinations of m precincts pairs as the set of matchings of size m, which is given by:

$$\mathcal{M}_m := \left\{ \mathbf{y} \in \{0,1\}^{|E|} : \sum_{e \in E} y_e = m, \sum_{e \in \delta(v)} y_e \leq 1 \quad \forall v \in V \right\}.$$

For every precinct $v \in V$, let J_v be the set of targets in that precinct. Note that $\{J_v\}_{v \in V}$ is a partition of the set of targets J, *i.e.*, $\cup_{v \in V} J_v = J$ and $J_u \cap J_v = \emptyset$ for all $u \neq v$. A defenders strategy selects m precinct pairings and also the target where each resource team is deployed. The combined patrol from the pairing of precincts u and v can only be deployed to a target in $J_u \cup J_v$. For each edge $e = (u, v) \in E$ we define $J_e = J_u \cup J_v$. It follows that the set I of defender strategies can be expressed as

$$I = \left\{ (\mathbf{y}, \mathbf{w}) \in \{0,1\}^{|E|+|J|} : \mathbf{y} \in \mathcal{M}_m, \sum_{j \in \cup_{v \in U} J_v} w_j \leq \sum_{e \in E(U) \cup \delta(U)} y_e \, \forall U \subseteq V, \sum_{j \in J} w_j = m \right\}. \quad (16)$$

For $(y,w) \in I$, variable y_e indicates whether edge e is selected for a precinct pairing and w_j indicates whether target j is patrolled. The first condition enforces that y identifies a matching of size m in V. The second bounds the coverage to a set of targets by the coverage on all incident edges to the precincts containing these targets. The last condition enforces that m targets are protected.

In this SSG, the defender selects $(y,w) \in I$ and an attacker of type $k \in K$, appearing with probability π^k, responds by selecting the target $j \in J$ to attack. This Stackelberg game can be represented by explicitly enumerating all strategies in I and using the MILP (D2) (or (MIP-G)) to obtain the optimal strategies. Unfortunately, this approach is computationally challenging as the resulting optimization problem considers one variable $x_{(y,w)}$ for each $(y,w) \in I$, which is an exponential number of variables in terms of edges. Below we introduce compact MILP formulations with a polynomial number of variables and exponentially many constraints in terms of the number of edges.

The compact formulation is derived similarly to Kiekintveld et al. (2009) and is based on the observation that if rewards are given by (14) and (15) then the utility of each player only depends on c_j, the coverage at a target $j \in J$. To introduce the compact formulation, given a probability distribution $x_{(y,w)}$ over the defender strategies $(y,w) \in I$, in addition to c_j we define z_e the coverage probability on edges $e \in E$ and $g_{e,j}$ the combined coverage over edge e and target $j \in J_e$ as follows:

$$c_j = \sum_{(y,w)\in I\,:\,w_j=1} x_{(y,w)} \qquad j \in J \tag{17}$$

$$z_e = \sum_{(y,w)\in I\,:\,y_e=1} x_{(y,w)} \qquad e \in E \tag{18}$$

$$g_{e,j} = \begin{cases} \displaystyle\sum_{(y,w)\in I:y_e=1,w_j=1} x_{(y,w)} & \text{if } j \in J_e \\ 0 & \text{o/w} \end{cases} \qquad e \in E, j \in J\ . \tag{19}$$

Given a graph $G = (V,E)$ with V the set of precincts and E the feasible pairings between precincts, Bucarey et al. (2017) introduce the following optimization problem as the compact version of (D2) when the set of defender strategies I is given by (16):

(BP)

$$\max_{c,z,q,s,f,g} \quad \sum_{k\in K} \pi^k f^k \tag{20}$$

$$\text{s.t.} \quad {\mathbf{q}^k}^\top \mathbf{1} = 1,\ \mathbf{q}^k \in \{0,1\}^{|J|} \qquad \forall k \in K, \tag{21}$$

$$\mathbf{c}^\top \mathbf{1} = m,\ \mathbf{c} \in [0,1]^{|J|} \tag{22}$$

$$\mathbf{z}^\top \mathbf{1} = m,\ \mathbf{z} \in [0,1]^{|E|} \tag{23}$$

$$\sum_{e\in\delta(v)} z_e \le 1 \qquad \forall v \in V, \tag{24}$$

$$\sum_{e\in E(U)} z_e \le \frac{|U|-1}{2} \qquad \forall U \subseteq V,\ |U| \ge 3, |U| \text{ odd} \tag{25}$$

$$\sum_{e\in E: j \in J_e} g_{e,j} = c_j \qquad \forall j \in J \tag{26}$$

$$\sum_{j\in J_e} g_{e,j} = z_e \qquad \forall e \in E \tag{27}$$

$$f^k \le D^k(j|p)c_j + D^k(j|u)(1-c_j) + (1-q_j^k)\cdot M \qquad \forall j \in J, \forall k \in K, \tag{28}$$

$$0 \le s^k - A^k(j|p)c_j - A^k(j|u)(1-c_j) \le (1-q_j^k)\cdot M \qquad \forall j \in J, \forall k \in K \tag{29}$$

$$\mathbf{s}, \mathbf{f} \in \mathbb{R}^K,\ \ \mathbf{g} \in [0,1]^{|E||J|}\ . \tag{30}$$

Constraint (21) ensures that attacker $k \in K$ attacks a single target $j \in J$. Constraints (22) and (23) indicate that the defender uses all resources and precincts are paired without exceeding the number of resources deployed. Constraint (24) indicates that a precinct's contribution to a pairing cannot exceed 1. Constraints (25) correspond to the *blossom* or *odd set* inequalities, introduced in Edmonds (1965), and together with (23) and (24) enforce that the coverage probabilities on the edges belong to the convex hull of the polytope of matchings of size m. Constraints (26) and (27) enforce the conservation between marginal coverages in nodes and edges. Finally, Constraints (28) and (29) ensure that c and q are mutual best responses. The objective function in our formulation, maximizes the defender's expected utility.

Here we present an alternative compact formulation (COMB) obtained from (MIP-G) when the set I is given by (16) by applying the coverage probabilities (17)-(19). In this formulation we use the transformation $t^k_{jl} = c_j q^k_l$ and that this implies $\sum_{l\in J} t^1_{jl} = c_j$ for any vector q feasible.

(COMB)

$$\max_{t,z,q,g} \quad \sum_{k\in K} \pi^k \left(D^k(j|p)t^k_{jj} + D^k(j|u)(q^k_j - t^k_{jj})\right) \tag{31}$$

s.t. (21), (23), (24), (25), (27)

$$\sum_{l\in J} t^k_{lj} = m q^k_j \qquad \forall j\in J, \forall k\in K \tag{32}$$

$$0 \le t^k_{lj} \le q^k_j \qquad \forall l,j\in J, \forall k\in K \tag{33}$$

$$\sum_{j\in J} t^k_{lj} = \sum_{j\in J} t^1_{lj} \qquad \forall l\in J, \forall k\in K \tag{34}$$

$$\sum_{e\in E: j\in J_e} g_{e,j} = \sum_{l\in J} t^1_{jl} \qquad \forall j\in J \tag{35}$$

$$A^k(j|p)t^k_{jj} + A^k(j|u)(q^k_j - t^k_{jj}) - A(l|p)t^k_{lj} - A^k(l|u)(q^k_j - t^k_{lj}) \ge 0 \qquad \forall l,j\in J, \forall k\in K \tag{36}$$

$$\mathbf{t}\in[0,1]^{|J|^2\times|K|},\ \mathbf{z}\in[0,1]^{|E|},\ \mathbf{g}\in[0,1]^{|E||J|}\ , \tag{37}$$

3. Correctness of formulations

In this section we show the correctness of formulations (BP) and (COMB) by proving they are respectively equivalent to (D2) and (MIP-G) when the set I is given by (16). We finish the section describing a branch and cut strategy to efficiently address the exponentially many odd set constraints in formulations (BP) and (COMB).

3.1. Equivalence of compact SSG formulations

To set the notation, we consider a combined resources SSG problem with defender action set given by (16) and defender and attacker utilities given by (14) and (15), respectively. We show that (BP) is equivalent to (D2) for a SSG with resource combination by showing that a feasible solution from one leads to a feasible solution in the other with same objective value and viceversa. We first state the following property, proved in the appendix.

Property 1 *Given $(y,w)\in I$ then the following holds*

$$w_j = 1 \Rightarrow \sum_{e\in E: j\in J_e} y_e = 1 \tag{38}$$

$$y_e = 1 \Rightarrow \sum_{j\in J_e} w_j = 1. \tag{39}$$

THEOREM 1. *Let the set I be given by (16) and the reward matrices by (14) and (15). Then Problem (BP) given by (20)-(30) is equivalent to (D2) given by (4)-(8).*

Proof of Theorem 1. We begin with a solution (x,q,s,f) feasible for (D2). Given this x we define $\mathbf{c}\in[0,1]^{|J|}$, $\mathbf{z}\in[0,1]^{|E|}$ and $\mathbf{g}\in[0,1]^{|E||J|}$ as in (17), (18), and (19), respectively. We now show that this (c,z,q,s,f,g) is feasible for (BP) and has the same objective value. Indeed, the objective function (20) and constraints (21) in (BP) are the same as (4) and (6) in (D2). In addition constraints (30) are satisfied from the definition of $\mathbf{c}$, $\mathbf{z}$ and $\mathbf{g}$ and the fact that $\mathbf{x}$ is a probability distribution.

Since the utilities are given by (14) and (15), we have that

$$\begin{aligned}
\sum_{(y,w)\in I} R^k_{(y,w)j} x_{(y,w)} &= \sum_{(y,w)\in I: w_j=1} R^k_{(y,w)j} x_{(y,w)} + \sum_{(y,w)\in I: w_j=0} R^k_{(y,w)j} x_{(y,w)} \\
&= D^k(j|p) \sum_{(y,w)\in I: w_j=1} x_{(y,w)} + D^k(j|u)\left(1 - \sum_{(y,w)\in I: w_j=1} x_{(y,w)}\right) \\
&= D^k(j|p)c_j + D^k(j|u)\left(1-c_j\right) \; .
\end{aligned}$$

This equality is used to conclude that (7) implies (28). Analogously for the attackers utility we have that (8) implies (29).

From the definition of $g_{e,j}$ we can write

$$\begin{aligned}
\sum_{j\in J_e} g_{e,j} &= \sum_{j\in J_e} \sum_{\substack{(y,w)\in I \\ y_e=1, w_j=1}} x_{(y,w)} = \sum_{j\in J_e} \sum_{(y,w)\in I} x_{(y,w)} y_e w_j \\
&= \sum_{(y,w)\in I} x_{(y,w)} y_e \sum_{j\in J_e} w_j = \sum_{(y,w)\in I} x_{(y,w)} y_e = z_e \; .
\end{aligned}$$

Where the next to last equality follows from (39). This shows that (27) is satisfied. Similarly expanding $\sum_{e\in E: j\in J_e} g_{e,j}$ and now using (38) gives that (26) is satisfied. In the same logic it follows that

$$\sum_{e\in E} z_e = \sum_{j\in J} c_j = \sum_{e\in E}\sum_{j\in J_e} g_{e,j} = \sum_{e\in E}\sum_{(y,w)\in I} x_{(y,w)} y_e = \sum_{(y,w)\in I} x_{(y,w)} \sum_{e\in E} y_e = m\ .$$

This proves that (23) is satisfied. Finally we observe above that the vector $z \in [0,1]^{|E|}$ satisfies $\mathbf{z} = \sum_{(y,w)\in I} x_{(y,w)}\mathbf{y}$ where $\mathbf{x}$ is a probability distribution over I and the vectors $\mathbf{y}$ correspond to m-matchings in the set V. Therefore $\mathbf{z} \in$ convex hull$(\mathcal{M}_m)$, which implies that $\mathbf{z}$ satisfies constraints (24) and (25) that characterize the m-matching polytope, completing the first part of the proof.

Algorithm 1 Procedure to identify mixed strategy $\mathbf{x}$ given feasible vectors $\mathbf{z}$, $\mathbf{c}$, $\mathbf{g}$

Step 1. For every edge $e \in E$ consider a column of height 1. Divide this $1 \times |E|$ rectangle in horizontal segments, one for each integer matching $y \in M_z$, with each segment of width λ_y. For each segment, corresponding to matching y, block out the areas corresponding to the edges that are not used in y (i.e.'NO' in Figure 1).

Step 2. For every edge $e \in E$, further subdivide the total area in its column that is not blocked out into smaller rectangles using the values of g_{ej} for all $j \in J_e$. Since $\sum_{j\in J_e} g_{ej} = z_e = \sum_{y\in M_z} \lambda_y y_e$, from (27) and the decomposition of $\mathbf{z}$, this partition uses all the area in the column of e that was not blocked out.

Step 3. Define $\mathbf{x}$ by extending each rectangle line into a horizontal line crossing all columns. The area between two horizontal lines represents a defender strategy. This area is contained in a matching y, that identifies the paired precincts, and for each edge e in the matching ($y_e = 1$), the area includes part of some g_{ej_e}, identifying the protected target j_e. Let the width of this horizontal line be $x_{(y,w)}$, where $w \in \{0,1\}^{|J|}$ is the indicator vector of the protected targets $\{j_e\}_{\{e: y_e=1\}}$.

We now consider a solution (c, z, q, s, f, g) feasible for (BP) and construct a solution (x, q, s, f) that is feasible for (D2) with the same objective function. Variables q, s and f are the same, this shows that the objective (4) and constraint (6) in (D2) are satisfied as they correspond to

(20) and (21) respectively. To build $\mathbf{x}$ we first recognize that by constraints (24) and (25), the vector $\mathbf{z} \in$ convex hull$(\mathcal{M}_m)$. Therefore, there is a finite set of integer m-matchings $M_z \subseteq \mathcal{M}_m$ such that we can express $\mathbf{z} = \sum_{y \in M_z} \lambda_y \mathbf{y}$, where $\lambda_y \geq 0$ for all $y \in M_z$ and $\sum_{y \in M_z} \lambda_y = 1$. Given this decomposition of $\mathbf{z}$, we apply Algorithm 1, illustrated in Figure 1, to obtain our variable $\mathbf{x}$.

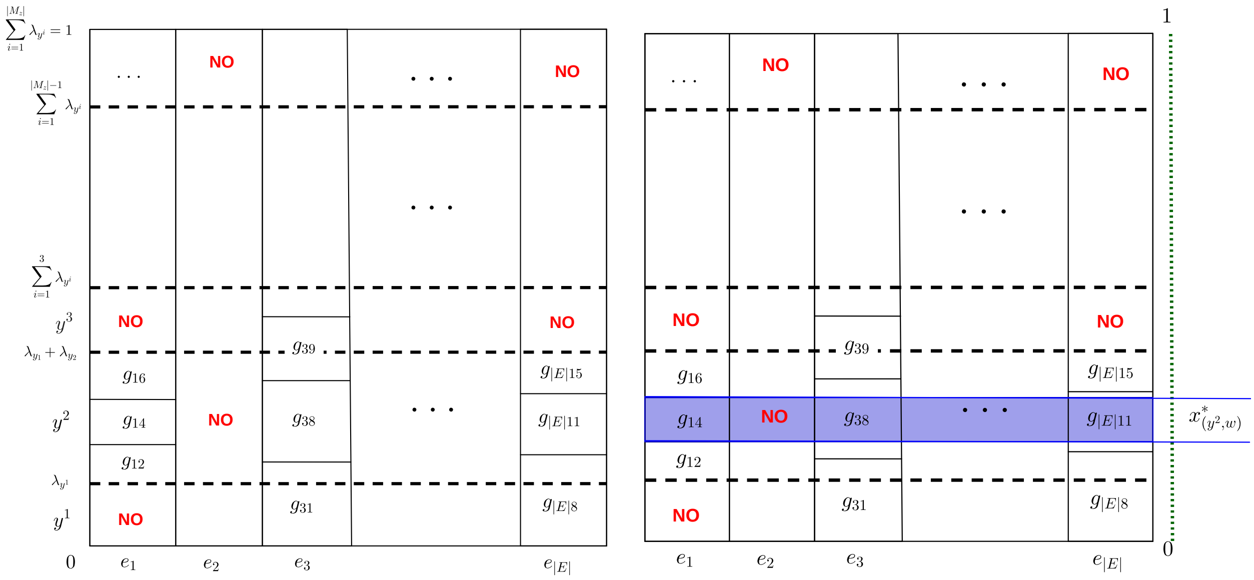

Figure 1 Construction to identify mixed strategy $\mathbf{x}$ given feasible vectors $\mathbf{z}, \mathbf{c}, \mathbf{g}$. Here matching y^2 includes edges $e_1, e_3, \ldots, e_{|E|}$. A constant width horizontal line (shaded) can be identified using portions of $g_{14}, g_{38}, \ldots, g_{|E|11}$. This shaded horizontal line corresponds to using precinct parings according to matching y^2 and protecting targets $w = \{4, 8 \ldots, 11\}$, whose width we label as $x_{(y^2,w)}$

We now show that all the (y, w) identified in Step 3 satisfy $(y, w) \in I$. First we have by construction that $y \in \mathcal{M}_m$. In addition for each edge $e = \{u_e, v_e\}$ used in y we identify a single target $j \in J_e = J_{u_e} \cup J_{v_e}$ so that $w_j = 1$. Therefore the vector w determines exactly m different targets. For any $U \subseteq V$, we have

$$\sum_{j \in \cup_{v \in U} J_v} w_j \leq \sum_{e:\{u_e,v_e\} \cap U \neq \emptyset} \; \sum_{j_e \in J_{u_e} \cup J_{v_e}} w_{j_e} = \sum_{e:\{u_e,v_e\} \cap U \neq \emptyset} y_e = \sum_{e \in E(U) \cup \delta(U)} y_e \; .$$

This proves that $(y, w) \in I$. Also, by construction, the horizontal lines corresponding to defender strategies cover the box completely, since the entire area of the $1 \times |E|$ that is not blocked out is covered by some $g_{ej} > 0$. Therefore Step 3 can be performed at any height of the box corresponding to a rectangle limit line. This implies that the values $x_{(y,w)}$ constructed in Step 3 are non-negative and sum to one, therefore they form a probability distribution over $(y,w) \in I$, i.e. the vector $\mathbf{x}$ constructed satisfies (5). To show that constraints (7) and (8) are satisfied we only need to show that the $\mathbf{x}$ constructed satisfies $c_j = \sum_{(y,w)\in I\,:\,w_j=1} x_{(y,w)}$.

Consider a target $j \in J$ and e such that $j \in J_e$ with $g_{ej} > 0$. We have from Step 2 in Algorithm 1 that the full amount of g_{ej} is assigned to some area in the column e intersected by matchings that use that column. That area can be partitioned into many strategies (y,w) but each of them has $y_e = 1$ and $w_j = 1$. Therefore we have that $g_{ej} = \sum_{(y,w)\in I} x_{(y,w)} w_j y_e$. We have then

$$c_j = \sum_{e:j\in J_e} g_{ej} = \sum_{e:j\in J_e} \sum_{(y,w)\in I} x_{(y,w)} w_j y_e = \sum_{(y,w)\in I} x_{(y,w)} w_j \sum_{e:j\in J_e} y_e = \sum_{(y,w)\in I} x_{(y,w)} w_j \ .$$

Where the last equality comes from (38). Q.E.D.

Analogously we can establish the equivalence between (MIP-G) and (COMB). The proof is almost identical and is given in the appendix.

THEOREM 2. *Let the set I be given by (16) and the reward matrices by (14) and (15). Then Problem (COMB) given by (31)-(37) is equivalent to (MIP-G) given by (9)-(13).*

3.2. A branch and cut solution method

Finding a solution for the compact formulations (BP) and (COMB) remains difficult since the blossom inequalities, present in both formulations in (25) are exponentially many. To face this large number of constraints we implement a branch and cut scheme where blossom inequalities are generated on the fly in a cut generation algorithm. The branch and cut scheme follows the approach introduced in Padberg and Rao (1982) and was implemented using the Lazy Constraint Callback in CPLEX 12.7.

We use the procedure introduced in Padberg and Rao (1982) that either detects whether a vector $\mathbf{z}$ satisfies the blossom inequality constraints or identifies the most violated one. In this procedure, the problem of detecting if a vector z satisfies all the set of blossom inequalities is reduced to a global min-cut set problem with odd cardinality in a related network. This problem can be solved in polynomial time with the Gomory-Hu algorithm. A simple implementation of the Gomory Hu algorithm, given by Gusfield (1990), is used in our implementation.

4. Recovering an implementable strategy

A strategy that can be implemented by security forces determines which pure strategy is done in each shift. In this section we describe how we use the optimal solutions to (BP) or (COMB), i.e. vectors $\mathbf{z}^*$, $\mathbf{c}^*$, to sample strategies (y, w) from I.

A first approach is to sample from I according to the optimal strategy $\mathbf{x}^*$. This can be achieved by constructing $\mathbf{x}^*$ from $\mathbf{z}^*$, $\mathbf{c}^*$, which is described in the second part of Theorem 1 (see Algorithm 1). This procedure requires that a fractional m-matching $\mathbf{z}^*$ is decomposed as a convex combination of pure matchings of size m. An algorithm to construct this decomposition is presented next. We finish this section with an alternative sampling method that uses solutions $(\mathbf{z}^*, \mathbf{c}^*)$ directly.

4.1. Decomposing $\mathbf{z}^*$ in pure matchings of size m

Given a vector $\mathbf{z}^*$ satisfying (23)-(25) we want to build a set $M_z \subseteq \mathcal{M}_m$ of matchings of size m and a set of weights $\{\lambda_y\}_{y\in\mathcal{M}_m}$ such that the vector $\mathbf{z}^*$ can be written as a convex combination of elements in M_z. This can be achieved by using a Dantzig-Wolfe approach, Dantzig and Wolfe (1960). The master problem can be stated as:

$$
\begin{aligned}
\text{(MP)} \quad \min \quad & \sum_{e\in E} Y_e + \Lambda && && (40)\\
& Y_e + \sum_{y\in\mathcal{M}_m} \theta_{ye}\lambda_y = z_e && \forall e \in E && (41)\\
& \sum_{y\in\mathcal{M}_m} \lambda_y + \Lambda = 1 && && (42)\\
& \lambda_y \geq 0 && \forall y \in \mathcal{M}_m && (43)\\
& Y_e, \Lambda \geq 0 && \forall e \in E, && (44)
\end{aligned}
$$

where λ_y is the weight of matching y. Variables Y_e and Λ represent auxiliary variables to be minimized. θ_{ye} is an indicator parameter taking value 1 when the edge e is included in the matching y. Let π_e and σ be the optimal dual variable associated to constraint (41) and (42) respectively. Then, the reduced costs of any column/matching is given by:

$$r_\theta = 0 - \sum_{e\in E} \pi_e \theta_e - \sigma.$$

The problem of adding a new column to a restricted master problem (i.e. only consider a subset of columns $M_z \subset \mathcal{M}_m$) can be stated as a maximum weight matching of size m with weights $\{\pi_e\}_{e\in E}$. If a matching θ has a weight greater than $-\sigma$ it is added to M_z and the restricted master is reoptimized. The algorithm stops when no new column is added (and the objective (40) is equal to zero), in which case z can be written as a convex combination of matchings in M_z.

4.2. An approximate sampling method

Finding the convex combination decomposition of $\mathbf{z}$ can be computationally expensive. We now propose a computationally efficient, but approximate, two stage sampling method that generates points $(y, w) \in I$ directly from $\mathbf{z}^*$ and $\mathbf{c}^*$. In Section 5, we investigate how accurate this approximate sampling method is.

Given z^*, we seek to select m edges that form a matching. We do this by iteratively sampling edges without replacement to build the set $\mathcal{E}$ and solving the maximum weight m-matching problem

$$\begin{aligned} \text{Max} \quad & \sum_{e\in\mathcal{E}} z_e^* x_e \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{M}_{\mathbf{m}}(\mathcal{E}) \ , \end{aligned}$$

where $\mathcal{M}_m(\mathcal{E})$ is the set of m-matchings using only edges in $\mathcal{E}$. To sample edges we consider a fixed index order for the set E. At a given iteration, let $\hat{m} = \sum_{e\in E\setminus\mathcal{E}} z_e^*$ be the sum of z^* over edges not yet sampled and generate a number between 0 and $\hat{m}$ uniformly. We select the edge in $E \setminus \mathcal{E}$ that causes the partial sum of z^*, according to the fixed order, meet or exceed the generated number for the first time. This edge is then added to $\mathcal{E}$.

Solving the maximum weight m-matching problem can have two possible outcomes: the problem is either infeasible and there are no matchings of size m with edges in $\mathcal{E}$ or the problem returns an optimal solution. If the problem is infeasible we continue iterating and sampling more edges for the set $\mathcal{E}$. If the problem returns an optimal matching we keep this as the sampled matching y. Note that this algorithm will produce a matching in at most $|E|$ iterations.

We generate the targets protected by sampling according to the normalized target coverage probability, defined below, for the precincts covered by the matching y found in the previous step. That is, we use the discrete probability

$$\bar{c}_j^* = \frac{c_j^*}{\sum_{k \in J_e} c_k^*} \qquad \forall j \in J_e \text{ s.t. } y_e = 1$$

to sample a single target for every $e \in E$ such that $y_e = 1$. This generates the protected targets that agree with the selected matching, giving the point $(y, w) \in I$. We refer to this method of generating solutions in I as the Approximate Sampling (AS) method.

5. Computational Results

In this section we evaluate the computational efficiency of the compact formulations (BP) and (COMB). Furthermore, we study the effectiveness of the alternative sampling procedure presented in Section 4.2. Finally we present a case study based on planning border patrols in the northernmost region of Chile. The experiments were programmed in Python 3.5 using CPLEX 12.7.

5.1. Computational Performance

We compare the performance of the general Stackelberg formulations (D2) and (MIP-G) to the Stackelberg security formulations developed specifically for the combined defender patrol problem (COMB) and (BP), and their branch and cut schemes described in 3.2. We refer to the branch and cut versions as (COMB_BC) and (BP_BC).

We consider randomly generated connected graphs with n nodes, a set of edges with an average node degree of 3 and m security resources. Further, we consider four targets within each node and three types of attackers. The payoff values are randomly generated such that penalties for the defender and each attacker are between - 5 and 0 and rewards for the defender and each attacker are between 5 and 10. We scale instances by increasing n and m.

We first focus on instances with two pairings (involving four nodes, $m = 2$) and $n \in \{5, 6, 7, \dots, 22\}$. We solve five random instances for each problem size considered and report expected results. In Figure 2, we show, on the left, the expected time (in seconds) it takes to solve the instances against the number of nodes in the graph, and on the right a performance profile, indicating the number of blossom inequalities added by the branch and cut procedure against the total percentage of instances solved. We note that the general Stackelberg formulations (MIP-G)

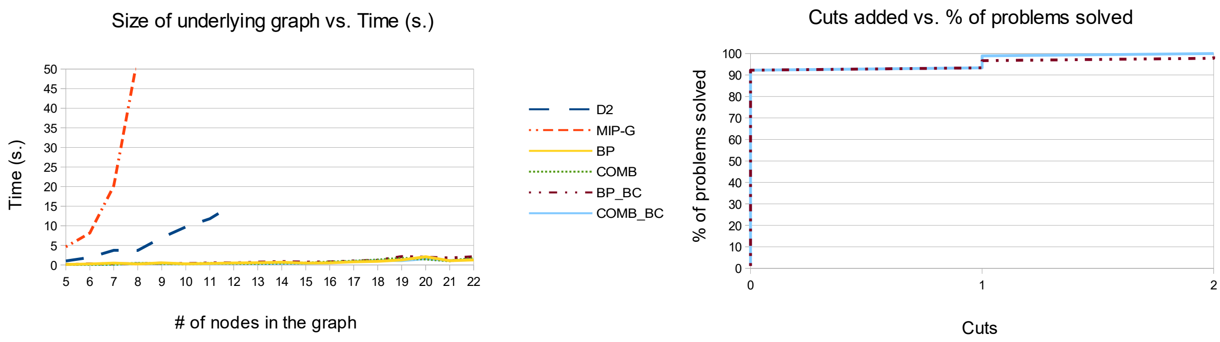

Figure 2 Instances with 2 security resources in graphs of up to 22 nodes

and (D2) can only solve small instances, at most with 8 and 11 nodes, respectively. We therefore do not consider formulations (MIP-G) and (D2) in the rest of our analysis. The compact formulations take less than three seconds to solve on average and the number of cuts generated by the branch and cut versions is small (at most two cuts are generated and 90% of instances require no cuts to find the optimal solution).

Our next set of experiments consider larger instances, with $n \in \{25, 30, 35, 40\}$ and $m = 4$ pairings. We present performance profile graphs for these instances in Figure 3. We observe that the branch and cut variants take less overall solution time and time to solve the LP relaxation, explore fewer nodes in the branch and bound tree and have a smaller MIP gap than (BP) and (COMB). This can be explained by the observation from Figure 2 that few blossom inequalities are required to solve these problems.

We further scale the instances to compare the efficiency of (BP_BC) and (COMB_BC). We consider graphs with $n \in \{30, 40, \ldots, 80\}$ and $m = 10$ security resources. In Figure 4, we show, on the left, the average running time of these formulations over the different size instances and, on the right, the number of cuts added by the branch and cut procedure against the percentage of

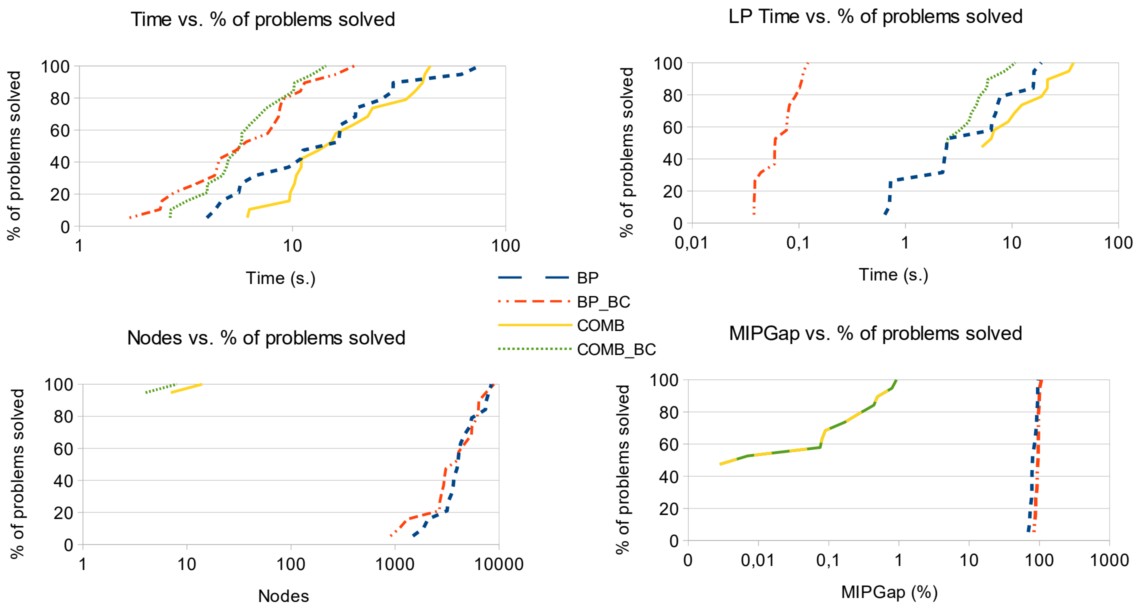

Figure 3 Performance profile graphs for instances with 4 security resources in graphs of up to 40 nodes

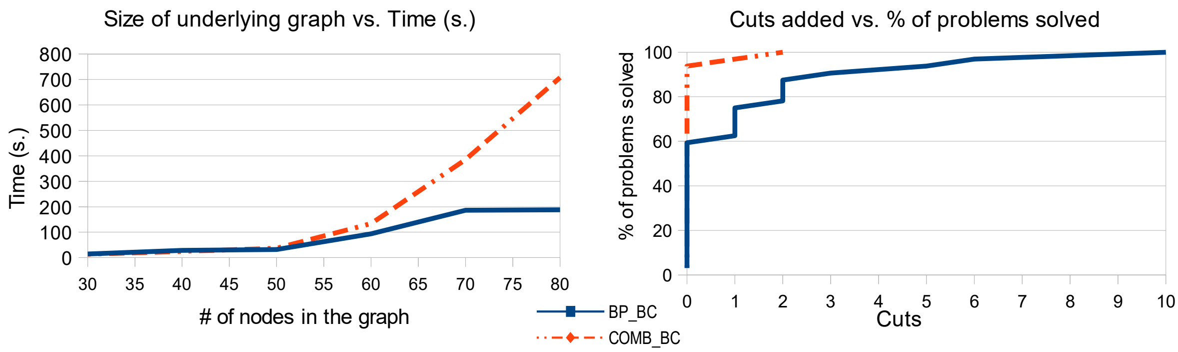

Figure 4 Instances with 10 security resources in graphs of up to 80 nodes

problems solved. In Figure 5 we present the performance profile graphs for these instances. These results show that solution times are comparable for both formulations, with a slight preference for (BP_BC) for larger instances. Here we observe the tradeoff between number of iterations and work per iteration. Formulation (COMB_BC) has a linear relaxation that takes longer to solve but that is compensated with achieving a smaller MIP gap, thus requiring less nodes in the branch and bound tree, and less blossom inequality cuts being generated.

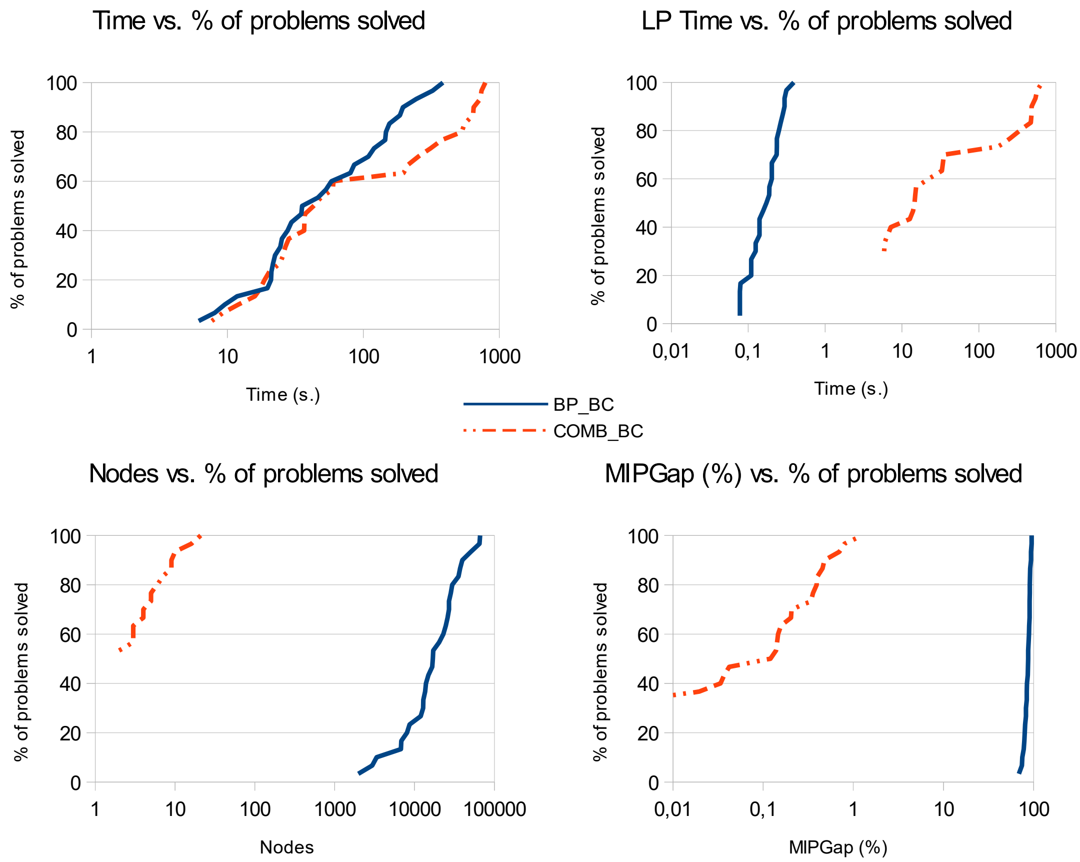

Figure 5 Performance profile graphs for instances with 10 security resources in graphs of up to 80 nodes

Further increasing problem size we observe in Figure 6 that the (BP_BC) formulation is able to solve instances on graphs with $n = 200$ and $m = 25$ security resources within a 1 hour time limit.

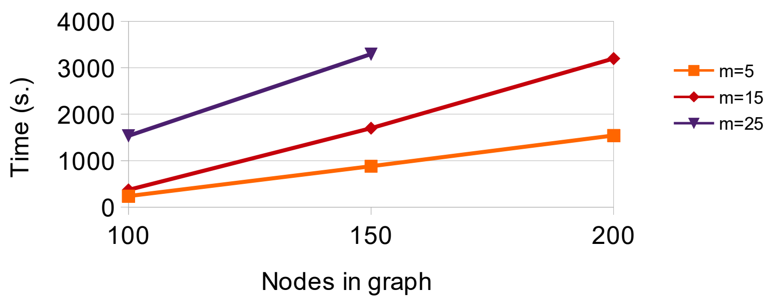

Figure 6 Solution times for (BP_BC) in instances with $n = 100, 150, 200$ and $m = 5, 15, 25$

5.2. Performance of sampling methods

Here we evaluate the computational time and accuracy of the Approximate Sampling (AS) method presented in Section 4.2. The sampling method obtains implementable patrolling strategies (solutions $(y,w) \in I$) from the solution (z,c) provided by the compact formulation of the combined defender security problem.

In Table 1 we show the computational times required by this heuristic method and the exact sampling method based on column generation (Section 4.1). The approximate sampling method can determine an implementable strategy in less than a second, independent of problem size. On the other hand, the computational time for the exact method increases dramatically with the size of the problem.

n/m	25/3	50/3	100/3	25/10	50/10	100/10
AS [s]	0.028	0.030	0.026	0.184	0.064	0.045
EM [s]	2.222	8.345	177.133	6.426	55.008	1176.458

Table 1 Computational time (seconds) of Approximate Sampling Method (AS) and Exact Method (EM).

To evaluate the accuracy of the approximate sampling method we compare the resulting empirical coverage $(\hat{z},\hat{c})$ obtained by repeatedly applying AS to the coverage solution (z,c) used to execute AS. This means that after conducting N approximate samples we let $\hat{c}_j = \frac{\text{Times target } j \text{ is selected}}{N}$ for each target $j \in J$. The value $\hat{z}_e$ is defined analogously. It is not possible to do this comparison with the corresponding strategies x over the set of combined actions I because there are multiple such strategies for every given coverage solution (z,c). To measure the error between c and our approximate $\hat{c}$ we use the infinity norm $\|c-\hat{c}\|_\infty = \max_{j\in J}|c_j - \hat{c}_j|$.

We consider a sampling size of $N = 1000$ over instances with $n \in \{25, 50, 100, 250\}$ nodes and $m \in \{3, 10\}$ security resources. For each instance size we generate 30 estimated $\hat{c}$ and plot the results as box diagrams as shown in Figure 7.

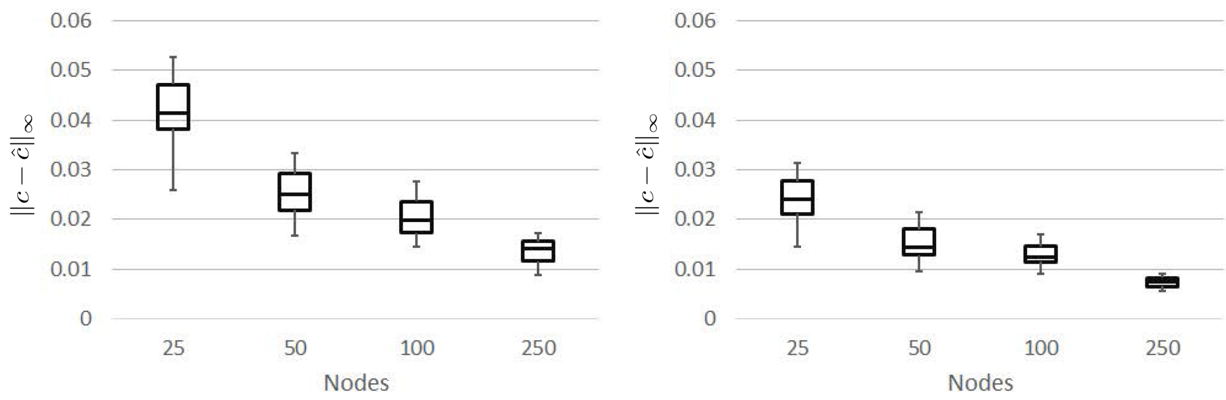

Figure 7 Total variation distance between c and $\hat{c}$ over instances of different size for $m = 3$ and $m = 10$ respectively.

First, we notice that for larger instances the distance between c and $\hat{c}$ decreases and so does the variance of this distance. Second, even for the smallest instance considered the maximum error it is not bigger than 0.06. So we achieve a fast and accurate way to obtain sampled strategies very close to the optimal ones. Analogous results are obtained for the Total Variation distance between z and the approximate $\hat{z}$.

5.3. Case study for border patrols

We now show the use of this methodology on a realistic border patrol problem proposed by Carabineros de Chile, which is the Chilean national police force. In this problem, Carabineros considers three different types of crime: drug trafficking, contraband of goods or animals and the illegal entry of people. Among other activities, Carabineros conducts patrols along the border with the aim of preventing the occurrence of these border crimes.

Here we describe a problem in planning specific night patrols. Carabineros has identified a set of locations along the border that can serve as vantage points from where to conduct surveillance with technical equipment such as night goggles and heat sensors. A night shift action consists of deploying a detail to these locations. Due to the large distances and harsh landscape, the detail is stationed at these locations overnight, forcing a commitment of these resources for a long period. The border region is divided into several police precincts. Due to the vast expanses and the lack of manpower, for the purpose of conducting night patrols, the border precincts are paired up, so each precinct that commits resources is not overburdened by this preventive patrolling action. A joint detail, combining personnel from paired precincts, conducts surveillance at one of the vantage points within the paired precincts territory. These border patrols are planned weekly.

We built an instance representing the border patrol problem for the Arica Parinacota Region of Chile. This region, which is Chile's northernmost region, is depicted in Figure 8. This region has international borders both with Peru and Bolivia. The border begins at the Pacific Ocean and goes to the north-east separating Chile from Peru while climbing to more than 4000 meters above the sea level. It then goes north-south in the Andean Plateau establishing the limit between Chile and Bolivia. The payoffs matrices were built using historic information of crime occurrences detailing location and a description of the crime, the patrol locations, and information on border foot paths and trails provided by police. We also used open source information on population estimates in Chile and across the border, police precinct boundaries, commercial valuations of goods and punishment for crimes in Chile (CEPAL (2000), Aduanas de Chile (2016)). A detailed description of how this instance is constructed appears in Bucarey et al. (2017). The payoff matrices constructed were vetted by Carabineros to ensure that the game reflected the observed behavior at the border.

In Figure 8 left, we illustrate a realistic instance where possible border precincts are enumerated from 1 to 8. In the right side, we give the graph representation that identifies possible precinct pairings. In this graph the possible outposts in each precinct are denoted by letters `[a]` to `[u]`.

For this instance, (COMB) with the branch and cut scheme computes the optimal coverage distribution in less than 3 seconds. Optimal coverage distribution with $m=3$ and the approximate coverage is showed in Figure 9. Dark bars represent the optimal coverage strategy and yellow bars represent the distribution of the approximate sampling method. Despite the greater deviation in the coverage in the edges, this difference is much smaller in locations coverage. The optimal solution, shown in Table 2 in the appendix, only uses 16 pure strategies $(y,w)\in I$.

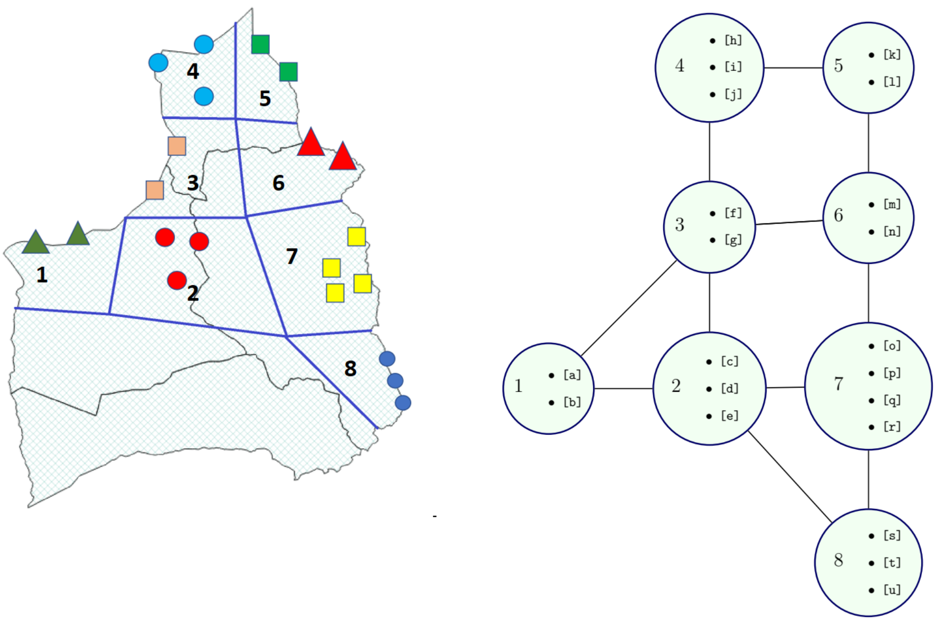

Figure 8 Realistic instance in the context of border patrolling and its graph representation.

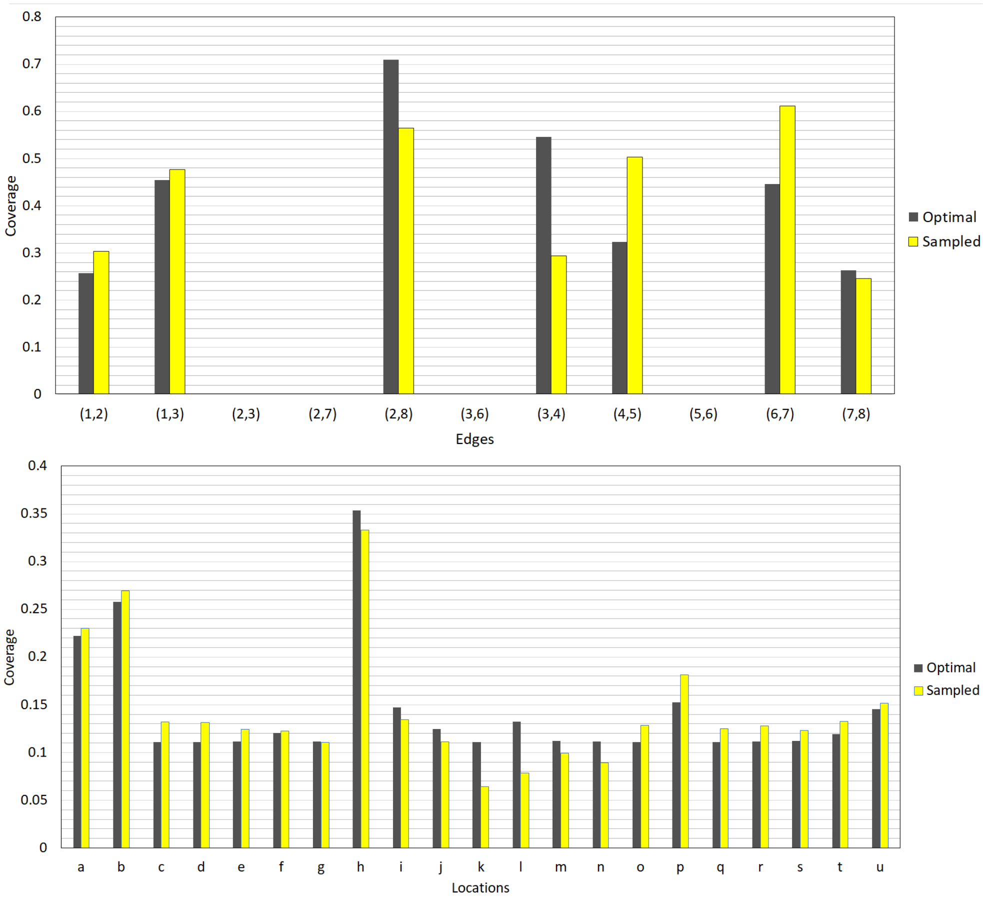

Figure 9 Coverage comparison in edges and targets in the realistic instance.

6. Conclusions and future work

In this article, we studied a special type of SSG played on a network. In this game, a defender has to combine resources to do patrol labors in a set of targets. We proposed a formulation, (COMB), that represent the set of mixed strategies in a compact form via marginal probabilities in the pairs of resources to team-up and the targets to cover. We proposed a method to retrieve an implementable strategy given the optimal marginals and we prove the validity of our formulation. To scale-up the instances that solvers are able to optimize, we used a cut generation algorithm.

We have further provided an alternative sampling method to recover an implementable defender strategy given the optimal coverage distributions. Computational tests have shown that the two-stage sampling method we describe, provides implementable strategies that do not deviate much from the optimal coverage distributions.

In future work we aim to extend our model and methodology to the case where each resource can be paired up with more than one resource, taking into account different capabilities, time schedules, and other considerations. The first case is easy because the b-matching polytope it also satisfies the Total Dual Integrality property.

References

Aduanas de Chile (2016) Qué tributos deben pagar las importaciones? https://www.aduana.cl/importaciones-de-productos/aduana/2007-02-28/161116.html.

Alpern S, Morton A, Papadaki K (2011) Patrolling games. *Operations Research* .

Bracken J, McGill JT (1973) Mathematical programs with optimization problems in the constraints. *Operations Research* 21(1):37–44.

Bucarey V, Casorrán C, Figueroa O, Rosas K, Navarrete H, Ordóñez F (2017) Building real stackelberg security games for border patrols. *Decision and Game Theory for Security: 8th International Conference, GameSec 2017.*

Casorrán C, Fortz B, Labbé M, Ordónez F (2017) A study of general and security stackelberg game formulations. Working Paper, Université Libre de Bruxelles.

CEPAL (2000) Costo económico de los delitos, niveles de vigilancia y políticas de seguridad ciudadana en las comunas del gran santiago. http://www.cepal.org/es/publicaciones/7258-costo-economico-de-los-delitos-niveles-de-vigilancia-y-politicas-de-seguridad.

Council of the EU (2016) European border and coast guard: final approval. http://www.consilium.europa.eu/en/press/press-releases/2016/09/14-european-border-coast-guard/, retrieved on 02/2017.

Dantzig GB, Wolfe P (1960) Decomposition principle for linear programs. *Oper. Res.* 8(1):101–111.

Department of Homeland Security (2017) Border security. https://www.dhs.gov/border-security, retreived on 02/2017.

Dimitrov NB, Morton DP (2012) Interdiction models and applications. Hermmann J, ed., *Handbook of Operations Research for Homeland Security*.

Dreiding R, McLay LA (2013) An integrated screening model for screening cargo containers for nuclear weapons. *European Journal of Operational Research* 230:181–189.

Edmonds J (1965) Maximum matching and a polyhedron with 0, l-vertices. *J. Res. Nat. Bur. Standards B* 69(1965):125–130.

Gusfield D (1990) Very simple methods for all pairs network flow analysis. *SIAM Journal on Computing* 19(1):143–155.

Hochbaum DS, Lyu C, Ordóñez F (2014) Security routing games with multivehicle chinese postman problem. *Networks* 64(3):181–191.

Jain M, Kardes E, Kiekintveld C, Ordoñez F, Tambe M (2010a) Security games with arbitrary schedules: A branch and price approach. *AAAI Conference on Artificial Intelligence*.

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

Kiekintveld C, Jain M, Tsai J, Pita J, Ordóñez F, Tambe M (2009) Computing optimal randomized resource allocations for massive security games. *AAMAS*, volume 1, 689–696.

Leitman G (1978) On generalized stackelberg strategies. *J. Optim. Theory Appl.* 26(4):637–643.

McClay LA, Lee AJ, Jacobson S (2010) Risk-based policies for airport security checkpoint screening. *Transportation Science* 44:333–349.

Nie X, Batta R, Drury CG, Lin L (2009) Passenger grouping with risk levels in an airport security system. *European Journal of Operational Research* 194(2):574–584.

Padberg MW, Rao MR (1982) Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research* 7(1):67–80.

Papadaki K, Alpern S, Lidbetter T, Morton A (2016) Patrolling a border. *Operations Research* .

Paruchuri P, Pearce JP, Marecki J, Tambe M, Ordóñez F, Kraus S (2008) Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. *AAMAS*, volume 2, 895–902.

Shieh E, An B, Yang R, Tambe M, Baldwin C, DiRenzo J, Maule B, Meyer G (2012) Protect: A deployed game theoretic system to protect the ports of the united states. *AAMAS*, volume 1, 13–20.

von Stackelberg H (2011) *Market Structure and Equilibrium* (Springer-Verlag Berlin Heidelberg), translated by Urch, L., Bazin, D., Hill, R.

Wright D, Liberatore M, Nydick RL (2006) A survey of operations research models and applications in homeland security. *Interfaces* 36:514–529.

Yan X, Nie X (2016) Optimal placement of multiple types of detectors under a small vessel attack threat to port security. *Transportation Research Part E: Logistics and Transportation Review* 93.

Yang R, Ford B, Tambe M, Lemieux A (2014) Adaptive resource allocation for wildlife protection against illegal poachers. *AAMAS*.

Yang R, Jiang AX, Tambe M, Ordoñez F (2013) Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. *IJCAI*.

Yin Z, Jiang AX, Tambe M, Kiekintveld C, Leyton-Brown K, Sandholm T, Sullivan JP (2012) Trusts: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine* 33(4):59–72.

Yin Z, Tambe M (2012) A unified method for handling discrete and continuous uncertainty in bayesian Stackelberg games. *AAMAS*.

Appendix

A1. Proof of Property 1

Property 1 *Given $(y,w) \in I$ then the following holds*

$$w_j = 1 \Rightarrow \sum_{e \in E : j \in J_e} y_e = 1 \tag{38}$$

$$y_e = 1 \Rightarrow \sum_{j \in J_e} w_j = 1 \ . \tag{39}$$

Proof of Property 1. Since $(y,w) \in I$ with $w_j = 1$, if we denote $v(j)$ the precinct that contains target j (i.e. $j \in J_{v(j)}$), we have

$$1 = w_j \le \sum_{k \in J_{v(j)}} w_k \le \sum_{e \in \delta(v(j))} y_e = \sum_{e \in E : j \in J_e} y_e \ ,$$

which proves (38). Here the first inequality comes from $w_k \in \{0,1\}$, the second from the definition of set I, and the last equality represents two equivalent ways of expressing the sum over the edges incident on $v(j)$. For the second implication, let $y_{e_1} = y_{e_2} = \cdots = y_{e_m} = 1$ be the set of arcs that are set to one in the matching y. Let $A(y)$ be the set of precincts v that are not paired by $y_{e_1} \ldots y_{e_m}$ (i.e. $A(y) = \{v \ : \ \delta(v) \cap \cup_{k=1}^m e_k = \emptyset\}$). Then, for any $v \in A(y)$ the definition of I implies $\sum_{k \in J_v} w_k \le \sum_{e \in \delta(v)} y_e = 0$. Similarly, for any arc $e_k = (u_k, v_k)$ we have, from the definition of I, that $\sum_{j \in J_{e_k}} w_j \le y_{e_k} + \sum_{e \in \delta(\{u_k, v_k\})} y_e = y_{e_k} = 1$. Here the sum in the second term is zero because $e \in \delta(\{u_k, v_k\})$ are not in matching y. With these inequalities we can separate

$$m = \sum_{j \in J} w_j = \sum_{v \in A(y)} \sum_{j \in J_v} w_j + \sum_{k=1}^m \sum_{j \in J_{e_k}} w_j \le \sum_{k=1}^m y_{e_k} = m \ ,$$

which completes the proof since the m terms $\sum_{j \in J_{e_k}} w_j \le 1$ and sum to m. Q.E.D.

A2. Proof of Theorem 2

Theorem 2 *Let the set I be given by (16) and the reward matrices by (14) and (15). Then Problem (COMB) given by (31)-(37) is equivalent to (MIP-G) given by (9)-(13).*

Proof of Theorem 2 . First we take a solution (h, q) feasible for (MIP-G). Note that from the definition of variables h we have that $h^k_{(y,w)l} = x_{(y,w)} q^k_l$, then $\sum_{l\in J} h^k_{(y,w)l} = x_{(y,w)}$ for each $k \in K$. We can then write the equivalent conditions to (17), (18), (19) and define variables (t, z, g) as:

$$t^k_{jl} = \sum_{(y,w)\in I: w_j=1} h^k_{(y,w),l} \qquad j,l \in J,\, k \in K \tag{45}$$

$$z_e = \sum_{(y,w)\in I: y_e=1} \sum_{l\in J} h^1_{(y,w),l} \qquad e \in E \tag{46}$$

$$g_{ej} = \begin{cases} \displaystyle\sum_{(y,w)\in I: y_e=1, w_j=1} \sum_{l\in J} h^1_{(y,w),l} & \text{if } j \in J_e \\ 0 & \text{o/w} \end{cases} \qquad e \in E, j \in J\ . \tag{47}$$

Now we show that the variables (t, z, q, g) are feasible for (COMB) with the same objective function. Since the utilities are given by (14) and (15). we have that:

$$\begin{aligned} \sum_{(y,w)\in I} R^k_{(y,w)j} h^k_{(y,w)j} &= \sum_{(y,w)\in I: w_j=1} R^k_{(y,w)j} h^k_{(y,w)j} + \sum_{(y,w)\in I: w_j=0} R^k_{(y,w)j} h^k_{(y,w)j} \\ &= D^k(j|p) \sum_{(y,w)\in I: w_j=1} h^k_{(y,w)j} + D^k(j|u) \sum_{(y,w)\in I: w_j=0} h^k_{(y,w)j} \\ &= D^k(j|p) t^k_{jj} + D^k(j|u) q^k_j \sum_{(y,w)\in I: w_j=0} x_{(y,w)} \\ &= D^k(j|p) t^k_{jj} + D^k(j|u)(q^k_j - t^k_{jj}). \end{aligned}$$

This expression is used to show that objective function is the same in (MIP-G) and (COMB). Using the analogous expression for the attacker utility function, we conclude that expression (13) implies (36) with this payoff structure.

Given the variables q are the same in both models, constrain (21) is also satisfied in (COMB). Given that conditions (45), (46) and (47) are equivalent to conditions (17), (18) and (19), the proof that z is in the matching polytope of fixed cardinality m is equivalent to the one showed in Theorem 1. In consequence, constrains (23), (24), (25) are satisfied. Note that $c_j = \sum_{l\in J} t^1_{jl}$ and the same argument in the proof of Theorem 1 lead us to the conclusion (27) and (35) are also satisfied. Also, constrains (10) directly imply (34). From the definition of t we have

$$\sum_{j\in J} t^k_{jl} = \sum_{j\in J} \sum_{(y,w)\in I: w_j=1} h^k_{(y,w),l} = \sum_{j\in J} \sum_{(y,w)\in I} h^k_{(y,w),l} w_j = \sum_{(y,w)\in I} h^k_{(y,w),l} \sum_{j\in J} w_j = q^k_j m.$$

This proves (32) is satisfied. Expression (33) is satisfied since by definition $t^k_{jl} \geq 0$ and

$$t^k_{jl} = \sum_{(y,w)\in I: w_j=1} h^k_{(y,w),l} \leq \sum_{(y,w)\in I} h^k_{(y,w),l} = q^k_j \ .$$

Now we take a solution (t, z, q, g) feasible for (COMB) and we build a feasible solution (h, q) for (MIP-G). To do so, we compute c from variables t as before. Then, we compute variables x using Algorithm 1. Finally we can compute variables $h^k_{jl} = x_{(y,w)} q^k_j$ which are feasible to (MIP-G) because it is an equivalent formulation to (D2) (for more details, see Casorrán et al. (2017)), and they have the same objective function. Q.E.D.

A3. Implementable strategies for the case study

Matching y	Locations w	Probabilty $x_{(y,w)}$	Matching y	Locations w	Probabilty $x_{(y,w)}$
{(1, 3), (2, 8), (4, 5)}	{a, c, h }	0.079	{(2, 8), (3, 4), (6, 7)}	{c, h, q}	0.051
{(1, 3), (2, 8), (4, 5)}	{a, c, k}	0.032	{(2, 8), (3, 4), (6, 7)}	{c, i, q}	0.014
{(1, 3), (2, 8), (4, 5)}	{a, d, k}	0.080	{(1, 2), (3, 4), (7, 8)}	{b, f, o}	0.111
{(1, 3), (2, 8), (4, 5)}	{a, d, l}	0.031	{(1, 2), (3, 4), (7, 8)}	{b,f, p}	0.146
{(1, 3), (2, 8), (4, 5)}	{a, e, l}	0.001	{(1, 2), (3, 4), (7, 8)}	{c, f, p}	0.001
{(1, 3), (2, 8), (4, 5)}	{f, e, l}	0.068	{(1, 3), (2, 8), (6, 7)}	{a, c, m}	0.132
{(2, 8), (3, 4), (6, 7)}	{c, h, m}	0.112	{(1, 3), (4, 5), (6, 7)}	{a, h, m}	0.026
{(2, 8), (3, 4), (6, 7)}	{c, h, n}	0.111	{(1, 3), (4, 5), (7, 8) }	{a, h, o}	0.006

Table 2 Implementable strategies in the realistic instance.