

Building Real Stackelberg Security Games for Border Patrols

Victor Bucarey¹(✉), Carlos Casorrán^{1,2}(✉), Óscar Figueroa³, Karla Rosas¹, Hugo Navarrete¹, and Fernando Ordóñez¹(✉)

¹ Departamento de Ingeniería Industrial, Universidad de Chile, Santiago, Chile
vbucarey@ing.uchile.cl, fordon@dii.uchile.cl

² Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium
ccasorra@ulb.ac.be

³ Carabineros de Chile, Santiago, Chile

Abstract. We present a decision support system to help plan preventive border patrols. The system represents the interaction between defenders and intruders as a Stackelberg security game (SSG) where the defender pools local resources to conduct joint preventive border patrols. We introduce a new SSG that constructs defender strategies that pair adjacent precincts to pool resources that are used to patrol a location within one of the two precincts. We introduce an efficient formulation of this problem and an efficient sampling method to construct an implementable defender strategy.

The system automatically constructs the Stackelberg game from geographically located past crime data, topology and cross border information. We use clustering of past crime data and logit probability distribution to assign risk to patrol areas. Our results on a simplified real-world inspired border patrol instance show the computational efficiency of the model proposed, its robustness with respect to parameters used in automatically constructing the instance, and the quality of the sampled solution obtained.

Keywords: Stackelberg games · Security application · Border patrol

1 Introduction

Securing national borders is a natural concern of a country to defend it from the illegal movement of contraband, drugs and people. The European Union (EU) created the European Border and Coast Guard in October 2016 in response to the recent increase in migrant flows into the EU [1]. 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” [2].

The task of patrolling the border requires monitoring vast stretches of land 24/7. Given the size of the problem and resource constraints, the global border

monitoring is constructed by pooling and coordinating resources from different locations. 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”.

In this paper we consider the problem of patrolling a border in the presence of strategic adversaries that aim to cross the border, taking into account the defender patrolling strategies. We consider 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 patrol problem the defender coordinates local resources to achieve a global defender strategy. Stackelberg Games, introduced by [3], are an example of bilevel optimization programs [4] where top level decisions are made by a player – the leader – that takes into account an optimal response of a second player – the follower – to a nested optimization problem.

Recent research has used Stackelberg games to model and provide implementable defender strategies in real life security applications. The Stackelberg games used in this context are referred to as Stackelberg security games (SSG). In these games, 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 his own utility. Examples of Stackelberg security games applications include assigning Federal Air Marshals to transatlantic flights [5], determining U.S. Coast Guard patrols of port infrastructure [6], police patrols to prevent fare evasion in public transport systems [7], as well as protecting endangered wildlife [8].

One of the challenges that has to be addressed in solving SSGs is problem size. When the defender action is to allocate limited resources to various targets, the set of possible defender actions is exponential in the number of resources and targets. In [9] a relaxation of the SSG 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.

In this work we tackle a border patrol problem, suggested by the Chilean National Police Force (known as Carabineros de Chile), where precincts pair up to jointly patrol border outposts in the presence of strategic attackers that observe these patrols before attacking. Combining resources from adjacent precincts provides overall coverage without excessively tasking each precinct. We use a mixed integer formulation for an SSG on a network, where the decision variables are two coverage distributions, one over the edges of the network and one over the targets that need to be protected. Further, we provide an approximate but computationally efficient sampling method that, given these optimal

coverage distributions, recovers an implementable strategy for the defender, i.e., a valid pairing of precincts, and a set of targets to protect. We conduct computational experiments to measure the performance of our formulation and further experiments are carried out to measure the quality of the implementable defender strategies—recovered through the approximate sampling method—with respect to the optimal coverage probabilities returned by our formulation. In addition, we describe a case study that tackles the real-life border patrol problem presented by Carabineros de Chile. We develop a parameter generation methodology to construct Stackelberg games that model the border patrol setting and we carry out a sensitivity analysis to study the effect that perturbations in our parameter generation methodology can have on the solutions provided by our software.

The rest of the paper is as follows. In Sect. 2, we present the problem formulation considered. In Sect. 3 we describe the sampling method proposed to retrieve an implementable patrolling strategy from the optimal solution obtained. In Sect. 4 we describe the border patrol case study. In Sect. 5, we provide computational experiments that evaluate the efficiency of our sampling strategy and measure the performance of our problem formulation. Finally, we present our conclusions and discuss future work in Sect. 6.

2 Problem Formulation and Notation

In this section we first introduce the general framework of Stackelberg games, the notation and a review of benchmark models. Then, we present a Stackelberg game that seeks to select coordinated defender strategies given heterogeneous resources when facing strategic attackers.

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 [10]. 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 with which the leader plays pure strategy i . Analogously, a mixed strategy for follower $k \in K$ is a vector $\mathbf{q}^k = (q_j^k)_{j \in J}$ such that q_j^k is the probability with which follower k plays pure strategy j . The payoffs for the agents are represented in the payoff matrices $(R^k, C^k)_{k \in K}$, where $R^k \in \mathbb{R}^{|I| \times |J|}$ gives 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 $k \in K$. The R_{ij}^k (C_{ij}^k) entry gives the reward for the leader (follower) when the leader takes action i and the k -th follower takes 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 $\sum_{i \in I} \sum_{j \in J} C_{ij}^k x_i q_j^k$ while the expected utility for the leader is given by $\sum_{k \in K} \pi_k \sum_{i \in I} \sum_{j \in J} R_{ij}^k x_i q_j^k$.

The objective of the game is for the leader to commit to a payoff-maximizing strategy, anticipating that every follower will best respond by selecting a payoff-maximizing strategy of their own. The solution concept used in these games is the Strong Stackelberg Equilibrium (SSE), introduced in [11]. In an SSE, the leader selects the strategy that maximizes his payoff given that every follower selects a best response breaking ties in favor of the leader when a follower is indifferent between several strategies. Therefore, without loss of generality the SSE concept can consider pure strategies as best response for each follower.

The problem of finding an SSE can be formulated as the following Mixed Integer Linear Program (MILP), referred to as (D2) [10]:

$$(D2) \quad \text{Max} \quad \sum_{k \in K} \pi^k f^k \tag{1}$$

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

$$\mathbf{q}^k \mathbf{1} = 1, \mathbf{q}^k \in \{0, 1\}^{|J|} \quad \forall k \in K \tag{3}$$

$$f^k \leq \sum_{i \in I} R_{ij}^k x_i + M(1 - q_j^k) \quad \forall k \in K, \forall j \in J \tag{4}$$

$$0 \leq s^k - \sum_{i \in I} C_{ij}^k x_i \leq M(1 - q_j^k) \quad \forall k \in K, \forall j \in J, \tag{5}$$

Constraints (2) and (3) indicate that the leader selects a mixed strategy and each follower responds with a pure strategy. The constant M in Constraints (4) and (5) is a large positive constant relative to the highest payoff value that renders the constraints redundant if $q_j^k = 0$. In Constraint (4), f^k is a bound on the leader’s reward when facing the follower of type $k \in K$. This bound is tight for the strategy $j \in J$ selected by that follower. In Constraint (5), s^k is a bound on follower k ’s expected payoff. This bound is tight for the best response strategy for that follower. Together, Constraints (4) and (5) ensure that the leader’s strategy and each follower’s strategies are mutual best responses. The objective function maximizes the leader’s expected reward.

The rewards in a Stackelberg game in a security setting only depend on whether the attack on a target is successful (if it is unprotected) or not (if the target is protected). Thus, we denote by $D^k(j|c)$ the utility of the defender when an attacker of type $k \in K$ attacks a covered target $j \in J$ and by $D^k(j|u)$ the utility of the defender when an attacker of type $k \in K$ attacks an unprotected target $j \in J$. Similarly, the utility of an attacker of type $k \in K$ when successfully attacking an unprotected target $j \in J$ is denoted by $A^k(j|u)$ and that attacker’s utility when attacking a covered target $j \in J$ is denoted by $A^k(j|c)$. We express as $j \in i$ the condition that defender strategy i patrols target j . The relationship between the payoffs in a security game and in a general game are as follows:

$$R_{ij}^k = \begin{cases} D^k(j|c) & \text{if } j \in i \\ D^k(j|u) & \text{if } j \notin i \end{cases} \tag{6}$$

$$C_{ij}^k = \begin{cases} A^k(j|c) & \text{if } j \in i \\ A^k(j|u) & \text{if } j \notin i. \end{cases} \tag{7}$$

2.2 SSG for Border Patrol

We now adapt the SG formulation above to a problem where the defender first pairs up the resources from different precincts to form m combined patrols and then decides where to deploy these combined patrols. Let V be the set of police precincts and 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 precinct 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 to patrol that are inside 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 $u \neq v$. The set of defender strategies selects the m precinct pairings and for each pairing, further selects a target within the precincts in the pairing where the resource team for that given pairing 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

$$I = \left\{ (\mathbf{y}, \mathbf{w}) \in \{0, 1\}^{|E|} \times \{0, 1\}^{|J|} : \begin{array}{l} \mathbf{y} \in \mathcal{M}_m, \\ \sum_{j \in \cup_{v \in U} J_v} w_j \leq \sum_{e \in E(U) \cup \delta(U)} y_e \quad \forall U \subseteq V, \\ \sum_{j \in J} w_j = m \end{array} \right\}. \quad (8)$$

For $(y, w) \in I$, the 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 indicates that the coverage provided to any subset of targets is bounded by the coverage on all incident edges to this subset of targets. The second condition enforces that total target coverage is equal to the required number of resources. The set of pure strategies for an attacker of type $k \in K$ consist in, for each $k \in K$, selecting a single $j \in J$ with probability 1. In our border context $q_j^k = 1$ indicates that a criminal of type $k \in K$ attempts to penetrate the border through target $j \in J$.

The Border Patrol problem can be formulated as (D2) given in (1)–(5) by explicitly considering the exponentially-many defender pure strategies in the set (8) with defender and attacker rewards given by (6) and (7), respectively. This formulation however is computationally challenging as the resulting MILP considers variables $x_{(y,w)}$ for each $(y, w) \in I$, of which there is an exponential number in terms of targets and edges. We derive a compact formulation following the formulation presented in [9]. Our formulation is based on the observation that

if rewards are given by (6) and (7) then the utility of each player only depends on c_j , the coverage at a target $j \in J$. Where this coverage can be expressed as

$$c_j = \sum_{(y,w) \in I : w_j=1} x_{(y,w)} \quad j \in J. \quad (9)$$

We further consider the variables

$$z_e = \sum_{(y,w) \in I : y_e=1} x_{(y,w)} \quad e \in E, \quad (10)$$

$$g_{ej} = \sum_{(y,w) \in I : y_e=1, w_j=1} x_{(y,w)} \quad e \in E, j \in J_e \quad (11)$$

where z_e is the coverage on edge $e \in E$ and can be obtained by summing over the pure strategies that assign coverage to that edge. Similarly, g_{ej} represents the combined coverage on edge $e \in E$ and target $j \in J$ and can be obtained by summing over the pure strategies where edge e and target j receive coverage.

Given a graph $G = (V, E)$ with V the set of precincts and E the feasible pairings between precincts, we propose the following SSG formulation for a border patrol (BP) problem:

(BP)

$$\text{Max} \quad \sum_{k \in K} \pi^k f^k \quad (12)$$

$$\text{s.t.} \quad \mathbf{q}^k \top \mathbf{1} = 1, \mathbf{q}^k \in \{0, 1\}^{|J|} \quad \forall k \in K, \quad (13)$$

$$\sum_{j \in J} c_j = \sum_{e \in E} z_e = m, \quad (14)$$

$$\sum_{e \in \delta(v)} z_e \leq 1 \quad \forall v \in V, \quad (15)$$

$$\sum_{e \in E(U)} z_e \leq \frac{|U| - 1}{2} \quad \forall U \subseteq V, |U| \geq 3, |U| \text{ odd} \quad (16)$$

$$\sum_{e \in E : j \in J_e} g_{e,j} = c_j \quad \forall j \in J \quad (17)$$

$$\sum_{j \in J_e} g_{e,j} = z_e \quad \forall e \in E \quad (18)$$

$$f^k \leq D^k(j|c)c_j + D^k(j|u)(1 - c_j) + (1 - q_j^k) \cdot M \quad \forall j \in J, \forall k \in K, \quad (19)$$

$$0 \leq s^k - A^k(j|c)c_j - A^k(j|u)(1 - c_j) \leq (1 - q_j^k) \cdot M \quad \forall j \in J, \forall k \in K \quad (20)$$

$$\mathbf{c} \in [0, 1]^{|J|}, \mathbf{z} \in [0, 1]^{|E|}, \quad (21)$$

$$\mathbf{s}, \mathbf{f} \in \mathbb{R}^K, \mathbf{g} \in [0, 1]^{|E||J|}. \quad (22)$$

Constraints (13) ensure that the each attacker $k \in K$ attacks a single target $j \in J$ with probability 1. Constraint (14) indicates that the defender uses all his resources in a feasible solution and that in order to form his resources he pairs up precincts without exceeding the number of resources he wants to deploy. Constraint (15) indicates that a precinct’s contribution to a pairing cannot exceed 1. Constraints (16) correspond to the *Odd Set Inequalities*, as introduced in [12], and together with (14) and (15) enforce that the coverage probabilities on the edges belong to the convex hull of the matching polytope of size m . Constraints (17) and (18) enforce the conservation between marginal coverages in nodes and edges. Finally, Constraints (19) and (20) are the same as in the formulation introduced in [9] and ensure that c and q are mutual best responses. The objective function in (BP), maximizes the defender’s expected utility.

2.3 Discussion

To ensure the correctness of the formulation (BP) we need to be able to recover the variables $x_{(y,w)}$ for $(y,w) \in I$ —that represent the probability distribution over the defender pure strategies—from an optimal solution $\mathbf{c}, \mathbf{z}, \mathbf{q}, \mathbf{s}, \mathbf{f}, \mathbf{g}$ to (BP). In particular, we need to find variables $\mathbf{x} \in [0, 1]^{|I|}$ that satisfy constraint (10). Note that the odd set inequalities (16) are necessary. We give an example of z variables for which there does not exist a probability distribution over I that would satisfy (10). Consider the example in Fig. 1, which shows a non-negative z that satisfies constraints (14) and (15) for $m = 2$ but violates the odd set inequalities (for the set $U = \{3, 4, 5\}$). We observe that this solution cannot be expressed as a convex combination of pure matchings of size 2, making it impossible to retrieve an implementable defender strategy \mathbf{x} .

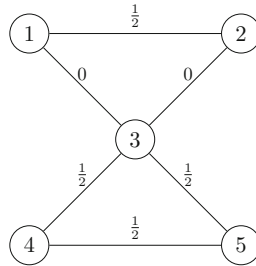


Fig. 1. Variables $\mathbf{z} \in [0, 1]^{|E|}$ that satisfy (14) and (15) but violate (16) for $m = 2$

Similarly, Constraints (17) and (18) also play a vital role in that they establish a link between the coverage variables on the edges and on the targets. This becomes much more apparent if one applies Farkas’ Lemma [13] on the linear system defined by (17), (18) and $g \geq 0$ to understand which conditions on c and

z guarantee feasibility of the system. Applying Farkas provides the following necessary conditions on c and z which offer a more direct interpretation:

$$\sum_{e \in E: e \in \bar{E}(U)} z_e \geq \sum_{j \in \cup_{u \in U} J_u} c_j \quad \forall U \subseteq V, \tag{23}$$

$$\sum_{j \in J: j \in \cup_{u \in V: e \in \delta(u) \cap E'} J_u} c_j \geq \sum_{e \in E'} z_e \quad \forall E' \subseteq E. \tag{24}$$

Constraint (23) states that given a subset of nodes, the coverage provided on all targets inside these nodes cannot exceed the weight of the edges incident to these nodes. Constraint (24) indicates that given a fixed set of edges E' , the weights on those edges is a lower bound on the coverage of the targets in nodes to which those edges are incident. Figure 2 shows an example of variables $\mathbf{z} \in [0, 1]^{|E|}$ and $\mathbf{c} \in [0, 1]^{|J|}$ that satisfy all constraints in (BP) except (17) and (18). The numbers on the top of the nodes represent total coverage on targets in that node, $\sum_{j \in J_u} c_j$, and the numbers on the edges represent the coverage probabilities on the edges, z_e . The solution in Fig. 2 violates (23) for $U = \{1, 2\}$. It is also not possible to find in this example an implementable defender strategy $\mathbf{x} \in [0, 1]^{|J|}$ related to these variables \mathbf{z} and \mathbf{c} .

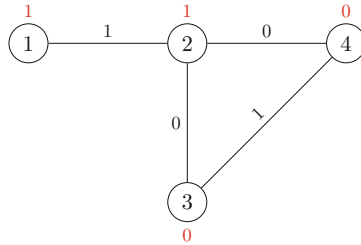


Fig. 2. Variables $\mathbf{z} \in [0, 1]^{|E|}$ and $\mathbf{c} \in [0, 1]^{|J|}$ that do not satisfy (17) and (18) with $m = 2$

It can in fact be proven that (BP) is a valid formulation for the SSG by showing that it is equivalent to (D2) in the sense that a feasible solution from one leads to a feasible solution in the other with same objective value and viceversa. Given a feasible solution to (D2), one can construct a feasible solution to (BP), with same objective value, through conditions (9)–(11). Conversely, given a feasible solution to (BP), a feasible solution to (D2), with same objective value, can be obtained relying on the fact that the cardinality constrained matching polytope is integral, [14]. The formal proof is omitted here.

3 Sampling Method

In this section we consider a two-stage approximate sampling method to recover an implementable mixed strategy $\mathbf{x} \in [0, 1]^{|J|}$ which complies with the optimal

probabilities on targets and edges, (c^*, z^*) as given by (BP). The quality of the proposed sampling method is later tested in Sect. 5.

Given z^* , the coverage vector over $|E|$ of size m , we discard all the edges in E such that $z_e = 0$. We then select m of the remaining edges according to a uniform random variable $U(0, m)$. This, in itself, could provide edges that do not form a matching. Therefore, let M be the set of m edges we have sampled. Now, solve the following optimization problem:

$$\begin{aligned} \text{Max} \quad & \sum_{e \in M} z_e^* y_e \\ \text{s.t.} \quad & \mathbf{y} \in \mathcal{M}_m \end{aligned}$$

Out of all matchings of size m , the objective function guarantees that we pick a maximum weight matching. The optimization problem either returns an optimal solution, in which case the edges in M admit a matching of size m , or, the problem is infeasible and such a matching cannot be constructed. If the problem is infeasible, we sample a new edge which we add to the set M and we re-optimize the optimization problem above. The sampled matching respects the optimal coverages if our algorithm returns the required matching after one iteration. Otherwise, the matching will deviate from the optimal coverages. We proceed in this iterative fashion until we construct a matching of size m . Note that this algorithm will produce a matching in at most $|E| - m$ iterations, as we know that such a matching exists in the original graph.

Having obtained M^* , the sampled matching of size m , the second stage of our sampling consists in sampling an allocation of resources to targets that satisfies the optimal target coverage probability returned by our formulation. To do so, we discard targets j that belong to precincts which are not paired. For each target j that belongs to a paired pair of precincts, say u and v , we normalize their coverage probability by the weight of the total coverage provided by the optimal coverage vector c^* in the two areas u and v that are paired and denote it by \bar{c}_j^* :

$$\bar{c}_j^* = \frac{c_j^*}{\sum_{j \in J_u \cup J_v} c_j^*} \quad \forall (u, v) = e \in M^*.$$

This way, we ensure that one resource is available per paired pairs of precincts. The defender's coverage on targets is composed by sampling over the newly constructed \bar{c}^* .

4 Case Study: Carabineros de Chile

In this section, we describe a realistic border patrol problem proposed by Carabineros de Chile. In this problem, Carabineros considers three different types of crime, namely, drug trafficking, contraband and illegal entry. In order to minimize the free flow of these types of crime across their borders, Carabineros organizes both day shift patrols and night shift patrols along the border, following different patterns and satisfying different requirements.

We are concerned with the specific actions that Carabineros can take during night shift patrols. The region is divided into several police precincts. Due to the vast expanses and harsh landscape at the border to patrol and the lack of manpower, for the purpose of the defender actions under consideration, a number of these precincts are paired up when planning the patrol. Furthermore, Carabineros have identified a set of locations along the border of the region that can serve as vantage points from where to conduct surveillance with technical equipment such as night goggles and heat sensors (Figs. 3 and 4). A night shift action consists in deploying a joint detail with personnel from two paired precincts to conduct vigilance from 22h00 to 04h00 at the vantage point located within the territory of the paired precincts. Due to logistical constraints, for a given precinct pair, Carabineros deploys a joint detail from every precinct pair to a surveillance location once a week.



Fig. 3. A Carabinero conducts surveillance



Fig. 4. Harsh border landscape

Carabineros requires a schedule indicating the optimal deployment of details to vantage points for a given week. Figure 5 depicts a defender strategy in a game with $m = 3$ pairings, $|V| = 7$ precincts and $|J| = 10$ locations. Table 1 shows a tabular representation of the implemented strategy for that week.

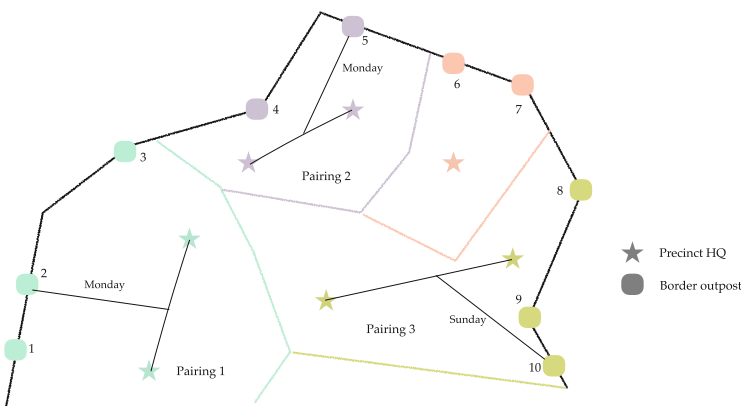


Fig. 5. Feasible schedule for a week

Table 1. Tabular representation for the feasible schedule in Fig. 5

Pairing/outpost	1	2	3	4	5	6	7	8	9	10
Pairing 1		M								
Pairing 2					M					
Pairing 3										Su

Therefore, we have an adjacency graph $G(V, E)$ where V is the set of police precincts and E are the edges that represent valid pairing of precincts. Further, the set of vantage points that need to be protected, corresponds to the set of targets J . Furthermore, the vantage points are partitioned among the different vertices of G , such that for a given $u \in V$, J_u contains all the vantage points inside precinct u . The set of attacker types is given by $K = \{\text{Drugs, Contraband, Illegal entry}\}$. In this setting, the pairings among precincts is fixed at the beginning of each month. Therefore, the game is separable into different standard SSG within every pair of paired precincts and one can use a standard SSG formulation such as the one presented in [9] to solve the different subproblems. Within each subproblem, the defender has a single resource to allocate to one of the different vantage points on a given day of the week. Given a coverage strategy over the targets, an adversary of type $k \in K$ plays the game with probability π^k and tries to cross the border through the vantage point $j \in J$ and on the day of the week that maximizes his payoff. It remains to construct the payoffs of the game for the problem described. To that end, Carabineros supplied us with arrest data in the region between 1999 and 2013 as well as other relevant data discussed next. In the following section, we discuss a payoff generation methodology.

4.1 Payoff Estimation

An accurate estimation of the payoffs for the players is one of the most crucial factors in building a Stackelberg model to solve a real-life problem. For each target in the game, we need to estimate 12 different values corresponding to a reward and penalty for Carabineros and the attacker for each type of crime $k \in K$.

We tackle this problem in several steps. First, we use QGIS [15], an open source geographic information system, to determine what we call action areas around each vantage point provided by Carabineros, based on the visibility range from each outpost. Such an action area represents the range of a detail stationed at a vantage point, *i.e.*, the area within which the detail will be able to observe and intercept a criminal.

Further, consider, for each type of crime $k \in K$, a network $\mathcal{G}^k(\mathcal{V}^k, \mathcal{E}^k)$ that models that type of crime’s flow from some nodes outside the border to some nodes inside the border, crossing the border precisely through the action areas previously defined. As nodes of origin for the different types of crime, we consider

cross border cities. As destination nodes we consider the locations inside Chile where Carabineros has performed an arrest of that type of crime. In order to have a more manageable sized network, we consider a clustering of these destination nodes. We later show that our methodology is robust versus changes in the number of cluster nodes.

Specifically, for a crime of type $k \in K$, let us define $S^k \subset \mathcal{V}^k$ as the nodes of origin situated outside the borders, $F^k \subset \mathcal{V}^k$ as the nodes of destination and J as the set of action areas along the border. Each destination node, $f \in F^k$, resulting from a clustering procedure is then assigned a demand $b(f)$ which corresponds to the number of destination nodes which are contracted into f . We use the k-means model to cluster crime data. For each $k \in K$, the edge set \mathcal{E}^k is constructed as follows. All nodes of origin are linked to all action areas. These areas are then linked to all of the destination nodes for crime $k \in K$. Figure 6 is a representation of such a network. The nodes to the left represent the points of origin of crime and the three nodes to the right are clusters of destination nodes for those crime flows. Note that crime enters the country through the four action areas marked as squares along the border.

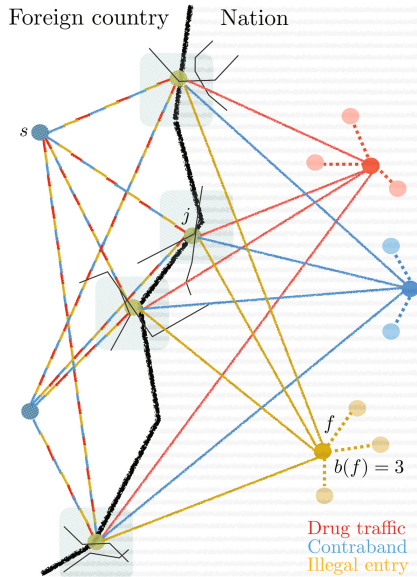


Fig. 6. Three crime flow networks, one per type of crime

We propose the following attractiveness parameter for a given action area $j \in J$ for a criminal of type $k \in K$ attempting to move from node $s \in S^k$ to node $f \in F^k$ through action area j :

$$U_{sf}^j = \frac{\text{Kilometers of roads inside action area } j}{d_{sj} + d_{jf}},$$

where d_{uv} is the distance in kilometers between nodes $u \in \mathcal{V}^k$ and $v \in \mathcal{V}^k$. This attractiveness parameter is proportional to the total length of roads inside a given action area and it is inversely proportional to how much an attacker moving from s^k to f^k has to travel in order to cross the border through area j .

We model the flow of crime $k \in K$ through a single route from $s \in S^k$ to $f \in F^k$ passing through $j \in J$ as follows:

$$x(s, j, f, k) = \frac{e^{\lambda U_{sf}^j}}{\sum_{s' \in S^k} \sum_{j' \in J} e^{\lambda U_{s'f}^{j'}}} \cdot b(f).$$

The flow of crime $k \in K$ through a route (s, j, f) is expressed as a proportion with respect to the flow of crime $k \in K$ through all routes leading into the same destination point $f \in F^k$. The parameter $\lambda \in \mathbb{R}_+$ provides a proxy of how the defender expects crime to behave. A value of $\lambda = 0$ means that crime $k \in K$ between any node of origin and destination distributes itself evenly among the different action areas. A high value of λ , however, is consistent with a flow of that type of crime through those action areas $j \in J$ with a higher attractiveness parameter U_{sf}^j . It follows that the total flow of crime of type $k \in K$ through $j \in J$ can be computed by summing over all origin nodes $s \in S^k$ and all destination nodes $f \in F^k$:

$$x(j, k) = \sum_{s \in S^k} \sum_{f \in F^k} \frac{e^{\lambda U_{sf}^j}}{\sum_{s' \in S^k} \sum_{j' \in J} e^{\lambda U_{s'f}^{j'}}} \cdot b(f) \quad \forall j \in J, \forall k \in K.$$

Based on this parameter, we propose the following values for the players' payoff values:

$$\begin{aligned} A^k(j|u) &= x(j, k) \cdot AG(k) && \forall j \in J, \forall k \in K, \\ A^k(j|c) &= -x(j, k) \cdot OC(k) && \forall j \in J, \forall k \in K, \\ D^k(j|c) &= 0 && \forall j \in J, \forall k \in K, \\ D^k(j|u) &= -x(j, k) \cdot AG(k) && \forall j \in J, \forall k \in K, \end{aligned}$$

where $AG(k)$ denotes the average gain of successfully committing crime $k \in K$, and $OC(k)$ the opportunity cost of being captured while attempting to perpetrate a crime $k \in K$. Note that the reward Carabineros perceives when capturing a criminal is 0, irrespective of the crime. Carabineros is only penalized when a crime is successfully perpetrated on their watch. These values were calculated following open source references [16–18] and were then vetted by Carabineros to ensure that our estimates were realistic.

4.2 Building Software for Carabineros

We provide Carabineros with a graphical user interface developed in PHP to determine optimal weekly schedules for the night shift actions for a set of border

precincts in the XV region of Chile. The software provided for Carabineros is divided into two parts: a first part devoted to the parameter generation of the game according to the indications of the previous section, and a second part, which solves for the optimal deployment of resources. We discuss the two parts separately.

Parameter Estimation Software. The objective of the parameter estimation software is to construct the payoff matrices for the SSG. This software allows for the matrices to be updated when new criminal arrests are recorded in Carabineros' database. The input for this software is a csv data file with arrest data which is uploaded to the software. The main screen of the software shows a map of the region to the left and the following options to the right:

1. *Crimes*: Shows all criminal arrests in the area, color-coded according to the type of crime.
2. *Nodes of origin*: Shows the nodes of origin used in the networks constructed to determine the crime flow through the action areas.
3. *Cluster*: Clusters the criminal arrest points and constructs the crime flow networks joining nodes of origin, action areas and the clustered arrest points, which are the destination nodes for the different types of crime. It displays the payoff matrices for the different action areas.
4. *Input file and update*: Allows to upload a csv data file with arrest data. One then re-clusters to obtain new destination points and to construct the new crime flow networks that lead to new payoff matrices.

Deployment Generation Software. The deployment generation software is the part of the software that optimizes the SSGs and returns an implementable patrols strategy for Carabineros. The user is faced with a screen that on the left shows a map of the region where the different action areas are color-coded along the border, and on the right shows different available user options. Clicking on an action area reveals the payoff values for that area. The values can be modified on-screen although this is discouraged. The user can additionally select the number of resources in a given paired pair of precincts. Increasing the number of resources can be used to model that a joint detail can perform a night-shift patrol as many times during a week as the number of resources he has. Further, the user can select the number of weekly schedules that are to be sampled from the optimal target coverage distribution, allowing him to change the weekly schedule to a monthly schedule. Once all parameters are set, clicking on solve returns the desired patrol schedule such as the one shown in Table 1.

Once a patrol strategy has been returned, the user can perform several actions. If the patrol is not to the planner's liking, he can re-sample based on the optimal coverage distribution returned by the optimization. This produces a different patrol strategy that still complies with the same coverage distribution over targets. The user can further impose different types of constraints on each paired pair of precincts to model different requirements such as forcing a

deployment on a given day of the week or to a particular target. Similarly, the user can forbid a deployment on a given day of the week, or forbid deployment to a given target. Further, the user can ensure that at least one of a subset of targets is protected or that deployment to a given target happens on at least one out of a subset of days. Solving the game under these constraints and sampling will produce a deployment strategy that complies with the user's requirements.

4.3 Robustness of Our Approach

We study the robustness of the solutions produced by our software to variations in the payoff matrices. Specifically, we study the robustness of our method against variations of two key parameters in the payoff generation methodology: λ , which models the defender's belief on how crime flows across the border and $b(f)$, which indicates the number of nodes clustered into a given destination node f . Equivalently, one can consider variations in a vector $h = (h_1, h_2, h_3)$ which determines the number of cluster nodes for the three types of crime considered. We study the effects of variations in the parameter λ and in the vector of cluster nodes h separately.

As a base case, we generate payoffs for the players by setting $\bar{\lambda} = 50$ and $\bar{h} = (6, 6, 6)$. This appears reasonable given the size of the problem and distribution and number of arrests per type of crime in the studied region. Let $\lambda \in \Lambda = \{0.5\bar{\lambda}, 0.75\bar{\lambda}, 1.25\bar{\lambda}, 1.5\bar{\lambda}\}$ and $h \in H = \{(h_1, h_2, h_3) \in \mathbb{N}^3 : h_t = \bar{h}_t \pm s, t \in \{1, 2, 3\}, s \in \{0, 1, 2, 3\}\}$. We denote by $c(\lambda, h)$, the optimal coverage probabilities on the targets when the payoffs have been defined according to λ and h . Given two vectors $p, q \in \mathbb{R}_+^{|J|}$, we consider the usual distance function between them:

$$d(p, q) = \sqrt{\sum_{j \in J} (p_j - q_j)^2}.$$

We identify $\lambda^* \in \arg \max\{d(c(\bar{\lambda}, \bar{h}), c(\lambda, \bar{h}))\}$ and $h^* \in \arg \max\{d(c(\bar{\lambda}, \bar{h}), c(\bar{\lambda}, h))\}$ and plot $c(\bar{\lambda}, \bar{h})$, $c(\lambda^*, \bar{h})$ and $c(\bar{\lambda}, h^*)$.

Figure 7 shows the optimal coverage probabilities $c(\bar{\lambda}, \bar{h})$, $c(\lambda^*, \bar{h})$ and $c(\bar{\lambda}, h^*)$ for a game with five paired police precincts and twenty targets. One can see that the optimal probabilities are very robust towards variations in the number of clusters. As one could expect, they are less robust to variations in the parameter λ . Recall that a low value of λ constructs the payoff matrices under the assumption that crime distributes itself uniformly among the different action areas $j \in J$. It is therefore understandable that the optimal coverage probabilities reflect this by trying to cover the targets uniformly. On the other hand the optimal coverage probabilities tend to be more robust for higher values of λ .

5 Computational Experiments

In this section we run computational experiments to explore the quality of the two-stage sampling method described in Sect. 3 that recovers an implementable

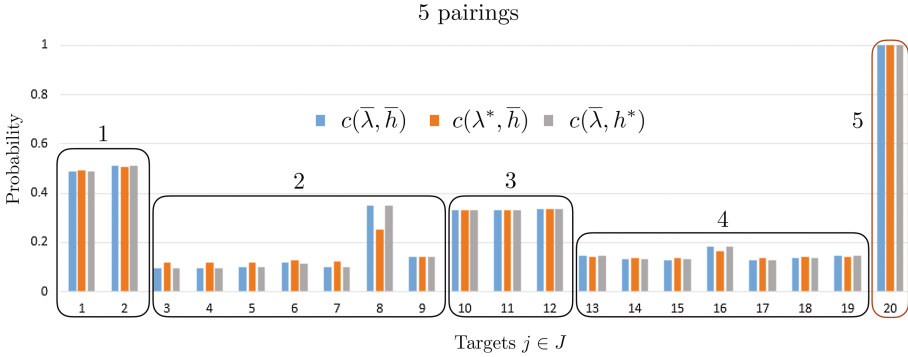


Fig. 7. Robustness of the solution method to variations in the parameters λ and h

defender strategy given an optimal solution to (BP). Further, we analyze the performance of the proposed formulation (BP) against solving the game, that results from explicitly enumerating all the defender pure strategies, with formulation (D2).

5.1 Performance of the Alternative Sampling Method

To evaluate the performance of the proposed alternative sampling method, consider an optimal solution to (BP). In particular, (c^*, z^*) are the optimal coverage distributions over targets and edges. Repeated executions of the sampling method will lead to estimates on said distributions (\hat{c}, \hat{z}) . In this section, we describe how to construct these estimates and study how close the estimated coverage distributions are to the optimal distributions.

Consider z^* and construct \hat{z} as follows. Sample $i = 1, \dots, N$ matchings of size m according to the first stage of the sampling method. In our experiments, $N = 1000$. For each edge $e \in E$, its estimated coverage is given by $\hat{z}_e = \frac{1}{N} \sum_{i=1}^N z_e^i$, where $z_e^i \in \{0, 1\}$ depending on whether or not edge $e \in E$ was sampled in sampling $i \in \{1, \dots, N\}$.

We use the Kullback-Leibler divergence [19] to measure the closeness of the two distributions z^* and \hat{z} over instances with n nodes where $n \in \{5, 25, 50, 100\}$. For each instance size, we generate 30 estimations \hat{z} and plot the results as box diagrams as shown in Fig. 8.

Observe that the Kullback-Leibler distance between z^* and \hat{z} is very small, below 0.2 over all instances, which is a good indicator that \hat{z} is a good estimator for z^* . in particular we observe that the larger the set of nodes in an instance the better an estimator \hat{z} appears to be. Further, for instances with 100 nodes, most of the \hat{z} have a Kullback-Leibler distance to z^* which is below 0.02. We performed the same analysis to measure the closeness of the optimal coverage distribution over targets c , to an estimated distribution \hat{c} , obtained from N samplings in the second stage of our sampling method, but omit it here due to space limitations. Our analysis reveals that \hat{c} is a good estimator for c^* .

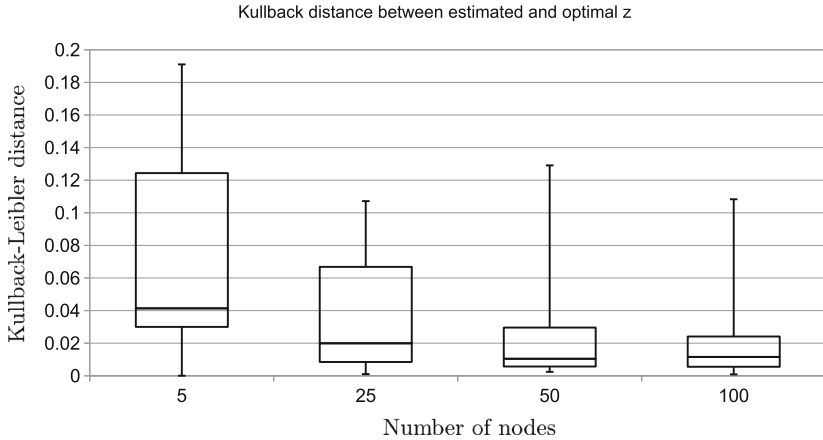


Fig. 8. Kullback-Leibler distance between z^* and \hat{z} over instances of different size

5.2 Performance of (BP)

We study the performance of the proposed formulation (BP) on randomly generated instances against using the formulation (D2) to solve the Stackelberg game that results from explicitly enumerating all the defender pure strategies. The instances we consider are generated as follows. We consider random graphs with n nodes, where $n \in \{5, 6, \dots, 22\}$ and edges such that the graphs are connected and that, in average, each node has degree three. Further, we consider four targets inside each node. The set of targets, J , is thus of size $|J| = 4n$. We consider $|K| = 3$. We then uniformly generate payoff values for the defender and each attacker type by considering for each player, rewards $D^k(j|c)$ and $A^k(j|u)$ for all $k \in K$ and $j \in J$ in the range $[0, 100]$ and penalties $D^k(j|u)$ and $A^k(j|c)$ for all $k \in K$ and $j \in J$ in the range $[-100, 0]$.

In Fig. 9, we show the running time of the different solution methods over the generated instances. On the left hand side, we consider instances where the number of pairings is 2. On the right hand side, we consider instances where the number of pairings is 3. For these last instances, we only consider graphs with up to 20 nodes. In both plots, for each instance size, we record the average solving time of 30 randomly generated instances. Our compact formulation (BP) outperforms (D2). The set of leader strategies grows exponentially and (D2) can only explicitly enumerate these strategies for very small graphs of less than 12 nodes. For graphs that both methods can handle, (BP) solves instances much faster than (D2). Our compact formulation (BP) scales much better than (D2) being able to comfortably handle instances on graphs with up to 20–21 nodes.

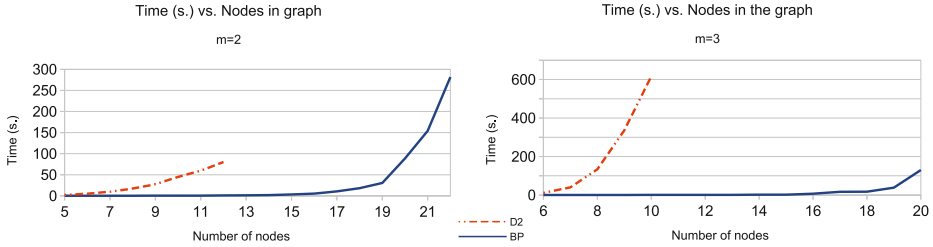


Fig. 9. Solving time (s) vs. number of nodes. Comparing (D2) and (BP)

6 Conclusions

In this paper, we have studied a special type of SSG played on a network. In this game, the defender has to commit to a mixed strategy which consists of two distribution strategies, one over the edges of the network, representing pairings between nodes, and one over the targets of the game which are inside the nodes. The defender can pair m nodes and protect m targets. Further, coverage on a target can only occur if the node in which the target is contained, is incident to a covered edge.

We have provided a compact formulation for the SSG network problem presented and also provided a sampling method to recover an implementable defender strategy given the optimal coverage distributions. In addition, we have described a real-life border patrol problem and have presented a parameter generation methodology that takes into account past crime data and geographical and social factors to construct payoffs for the Stackelberg game. Robustness tests have shown that the solutions our software provides are fairly robust to the networks we generate as well as to minor changes in the flow of crime along the border. 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. Further computational tests have shown (BP) to have smaller solution times and better scaling capabilities than the extensive formulation (D2) on randomly generated security instances.

There are many promising lines of future work. First, from a Mathematical Programming perspective, we intend to develop decomposition approaches for (BP)—which has an exponential number of the so-called odd set inequalities—to allow it to efficiently solve instances on larger graphs. Second, from a modeling perspective several enhancements could be addressed. In the model presented, a single security resource is available to patrol a target in a pairing of precincts, as it happens in the border patrol problem studied. A natural extension is to consider that different pairings of precincts have different numbers of security resources available to patrol. Further, our payoff estimation methodology could be enhanced in different ways. Temporal weighing of crime data would increase the relative importance of the more recent crimes. Our estimation methodology

currently builds the attractiveness of the action areas for a certain type of criminal based on road density around the action area and distances to be traveled by the criminals from source to destination. Other environmental factors such as maximum altitude or availability of shelter along a route or distance of settlements from a route could be taken into account to compute a more realistic attractiveness of an outpost. The research question that remains is verifying whether these modeling enhancements can lead to a better payoff estimation and, thus, to a better representation of the game. Finally, we plan to evaluate the proposed patrol planner following deployment. This evaluation should include both a comparison of crime rate data before and after the deployment of this system and the expert validation that Carabineros de Chile will undertake.

Acknowledgements. Casorrán 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. Ordóñez acknowledges the support by CONICYT through Fondecyt grant 1140807 and the Complex Engineering Systems Institute, ISCI (CONICYT: FB0816).

References

1. Council of the EU: European border and coast guard: final approval (2016). <http://www.consilium.europa.eu/en/press/press-releases/2016/09/14-european-border-coast-guard/>. Accessed Feb 2017
2. Department of Homeland Security: Border security (2017). <https://www.dhs.gov/border-security>. Accessed Feb 2017
3. von Stackelberg, H.: Principios de teoria economica. Oxford University Press, New York (1952)
4. Bracken, J., McGill, J.T.: Mathematical programs with optimization problems in the constraints. *Oper. Res.* **21**(1), 37–44 (1973)
5. Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., Ordóñez, F.: Software assistants for randomized patrol planning for the lax airport police and the federal air marshal service. *Interfaces* **40**(4), 267–290 (2010)
6. Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G.: Protect: a deployed game theoretic system to protect the ports of the United States. In: Richland, S. (ed.) Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Richland, SC, vol. 1, pp. 13–20. International Foundation for Autonomous Agents and Multiagent Systems (2012)
7. Yin, Z., Jiang, A.X., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., Sullivan, J.P.: Trusts: scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Mag.* **33**(4), 59–72 (2012)
8. Yang, R., Ford, B., Tambe, M., Lemieux, A.: Adaptive resource allocation for wildlife protection against illegal poachers. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2014)
9. Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., Tambe, M.: Computing optimal randomized resource allocations for massive security games. In: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2009, Richland, SC, vol. 1, pp. 689–696. International Foundation for Autonomous Agents and Multiagent Systems (2009)

10. Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordóñez, F., Kraus, S.: Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, Richland, SC, vol. 2, pp. 895–902. International Foundation for Autonomous Agents and Multiagent Systems (2008)
11. Leitman, G.: On generalized Stackelberg strategies. *J. Optim. Theor. Appl.* **26**(4), 637–643 (1978)
12. Edmonds, J.: Maximum matching and a polyhedron with 0, 1-vertices. *J. Res. Nat. Bur. Stand. B* **69**(1965), 125–130 (1965)
13. Farkas, J.: Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik* **124**, 1–27 (1902)
14. Schrijver, A.: *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, Heidelberg (2003)
15. QGIS: QGIS Geographic Information System. Open Source Geospatial Foundation (2009)
16. Comisión Económica para América Latina y el Caribe: Costo económico de los delitos, niveles de vigilancia y políticas de seguridad ciudadana en las comunas del gran santiago (2000). <http://www.cepal.org/es/publicaciones/7258-costo-economico-de-los-delitos-niveles-de-vigilancia-y-politicas-de-seguridad>
17. Aduanas de Chile: Qué tributos deben pagar las importaciones? (2016). <https://www.aduana.cl/importaciones-de-productos/aduana/2007-02-28/161116.html>
18. Ministerio del Trabajo y Previsión Social: Reajusta monto del ingreso mínimo mensual, de la asignación familiar y maternal del subsidio familiar, para los periodos que indica (2016). <http://www.leychile.cl/Navegar?idLey=20763>
19. Kullback, S., Leibler, R.A.: On information and sufficiency. *Ann. Math. Stat.* **22**(1), 79–86 (1951)