Discussion of Fairness and Implementability in Stackelberg Security Games

Víctor Bucarey, Martine Labbé

To cite this version:

Discussion of Fairness and Implementability in Stackelberg Security Games*

Victor Bucarey[1,2] and Martine Labbé[1,2]

[1] Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium.
{vbucarey, mlabbe}@ulb.ac.be
[2] INOCS, Inria Lille-Nord Europe, France.

Abstract. In this article we discuss the impact of fairness constraints in Stackelberg Security Games. Fairness constraints can be used to avoid discrimination at the moment of implementing police patrolling. We present two ways of modelling fairness constraints, one with a detailed description of the population and the other with labels. We discuss the implementability of these constraints. In the case that the constraints are not implementable we present models to retrieve pure strategies in a way that they are the closest in average to the set of fairness constraints.

Keywords: Fairness · Implementability · Stackelberg Security Games.

1 Introduction

In the last years, Stackelberg Security Games have been applied in several real domains such as airport security [11], IRIS for security of flights [7], ports [13] and border [1] patrolling and fare evasion [5], among others. In these games, the leader, also called *defender* must protect a set of *targets* with limited *resources* available from a possible attack performed by the follower or *attacker*. The payoff structure depends only on whether the target attacked is being protected or not [8]. Each defender resource has a set of possible *schedules*, that is, the possible subset of targets they can protect in one strategy. If all the resources have the same possible set of schedules we name this game with *homogeneous resources*. Otherwise, it is called with *heterogeneous resources*. Several mixed integer formulations, for both, general Bayesian Stackelberg games and Bayesian Stackelberg Security games are presented in [9], [4].

One of the key points of the scalability of SSG is the representation of the set of strategies of the defender. Instead of taking into account every single pure strategy, this set is represented through the frequency in which each target is protected. These frequencies are called *coverage distribution*.

In some applications, payoffs matrices are generated in a black-box. In others, they are built by real data. In both cases these payoffs can lead to discriminative outcomes. In the first case, this is due to the fact that there is not a known

* Authors have been partially supported by the Fonds de la Recherche Scientifique -FNRS under Grant(s) no PDR T0098.18.

methodology. In the second case, this is due to the manner in which data is gathered. For instance, according to [10], officers generally stop black drivers at higher rates than white drivers, and stop Hispanic drivers at similar or lower rates than white drivers. Another example came from the United Kingdom. According to the Governamental institution Ethnicity in the UK[3], in 2016/17, there were 4 stop and searches for every 1,000 White people, compared with the 29 stop and searches for every 1,000 Black people. Among specific ethnic groups, the Chinese and Mixed White/Asian groups consistently had the lowest rates of stop and search since 2009/10 [6].

The relationship between algorithms and discrimination is a major concern for the European Union Agency for Fundamental Rights - FRA [12]. They state:

> When algorithms are used for decision making, there is potential for discrimination against individuals. The principle of non-discrimination, as enshrined in Article 21 of the Charter of Fundamental Rights of the European Union (EU), needs to be taken into account when applying algorithms to everyday life.

And later,

> Big data claims to be neutral. It isn't ... machine learning depends upon data that has been collected from society, and to the extent that society contains inequality, exclusion or other traces of discrimination, so too will the data. Consequently, unthinking reliance on data mining can deny members of vulnerable groups full participation in society.

While there are several real world problems where fairness is important, there is no a specific way to measure fairness. For instance, for classifications problems, it is used as fairness constraints the following considerations: *disparate treatment* which implies that the probability of any classifier output does not change after observing a sensitive feature; *disparate impact* which implies that the probability of classifying with a positive value; and *disparate treatment* which implies that the probability of misclassification does not change with some sensitive feature [15].

In the SSG context, bias in the data can be translated into allocate more/less surveillance focused in race, wealth/poverty or any other type of discrimination. In this work we study different ways to include constraints in order to avoid discrimination SSGs from a tactical point of view. We are interested in studying how these considerations could be implemented and how much we lose by adding these considerations in terms of expected utilities.

SSG can be seen as the problem of allocating resources to targets in a random fashion. A relevant work that came from the problem of designing randomized allocations is presented in [3]. In this context, authors define the concept *implementability* of a set of constraints when any random allocation under this set of constraints can be decomposed in a convex combination of pure

[3] http://www.ethnicity-facts-figures.service.gov.uk/

allocations satisfying this set of constraints. Sufficient and necessary conditions for the implementability of a set of constraints are given, based on the *bihierarchy* structure of the constraints.

Sometimes, there are constraints that are necessary in real applications but they are not implementable in general. In this article we show that including fairness constraints in a detailed description of the population inside each target might not be implementable. That means, that in average solutions satisfy those constraints but the pure strategies that implement this solution could not satisfy them. We will show some models and algorithms to retrieve pure strategies minimizing the violations of such constraints.

The questions that we aim to answer are the following:

- Is it possible to model coverage distributions including fairness considerations?
- Are they implementable? If not, how can we include those considerations in practical settings?
- How much does the defender lose, in terms of expected utility, by including these considerations?

Our first contribution is to model fairness constraints in the coverage probabilities in SSGs. We present two models, one focused on a detailed description of the population, the second one based on labels on the targets. Our second contribution is to show that the model based on labels is implementable. Our third contribution is to provide a methodology for implementing schedules allocating low probability to strategies that violate more the set of non-implementable constraints.

The rest of the paper is as follows. In Section 2 we introduce the main concepts related to SSGs, Implementability and Random Allocations. We give an introductory example for this problem. In Section 3, we provide the models that are discussed in this work. In Section 4 we provide a discussion about the implementability of the coverage distribution returned by these models. Also, some extensions are presented. In Section 5 computational results are shown. Our models are tested in a realistic instance presented in Section 6. Finally, our conclusions are presented in Section 7.

2 Problem Statement and Notation

2.1 Stackelberg Security Games and compact formulations

In SSGs, the leader, named in this context *defender*, must protect from the followers, named *attackers*, a set of targets J. The payoffs for the players only depend on whether a target is protected or not. In consequence, several strategies have identical payoffs. Thus, we denote by $D^k(j|p)$ 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 $A^k(j|p)$. We denote π_k the probability of the defender facing attacker k.

In the heterogeneous resources setting there is a set Ω of resources, $|\Omega| = m$, in which each one can be allocated to a possible subset of targets J_ω. If $J = J_\omega$ for each $\omega \in \Omega$, we call it homogeneous resources. A pure strategy $i \in I$ for the leader is an allocation of resources to targets. That is

$$I = \left\{ \{a_\omega \in \{0,1\}^{|J_\omega|}\}_{\omega\in\Omega} : \sum_{j\in J_\omega} a_{\omega j} \ \leq\ 1 \ \forall\ \omega \in \Omega \right\}$$

in the case of heterogeneous resources, or

$$I = \left\{ a \in \{0,1\}^{|J|} : \sum_{j\in J} a_j \leq m \right\}$$

in the case of homogeneous resources. In this context, in the homogeneous case $|I| = \sum_{k=1}^{m} \binom{n}{k}$. Authors in [8] provide a compact formulation using the transformation

$$c_j = \sum_{a\in I: a_j = 1} x_a, \tag{1}$$

where c_j represents the frequency coverage of target j and x_a represents the probability of playing strategy $a \in I$. The formulation for computing a SSE in the homogeneous case (HOM) is stated as follows:

$$\begin{array}{llll}
\text{(HOM)} \ \max\limits_{c,q,f,s} & \sum_{k\in K} \pi_k f_k & & (2) \\
\text{s.t.} & f_k \leq D^k(j|p)c_j + D^k(j|u)(1-c_j) + M(1-q_j^k) & j \in J,\ k \in K & (3) \\
& 0 \leq c_j \leq 1 & j \in J & (4) \\
& \sum_{j\in J} c_j = m & & (5) \\
& \sum_{j\in J} q_j^k = 1 & j \in J,\ k \in K & (6) \\
& 0 \leq s^k - A^k(j|p)c_j - A^k(j|u)(1-c_j) \leq M(1-q_j^k) & j \in J,\ k \in K & (7) \\
& f^k, s^k \in \mathbb{R} & & (8) \\
& q_j^k \in \{0,1\} & j \in J,\ k \in K & (9)
\end{array}$$

where c_j represents the frequency with which target j is protected. Variables f^k represent the expected utility for the defender of facing attacker k and s^k the expected utility of attacker k. The objective function (2) to be maximised represents the expected utility of the defender. Expression (3) states an upper bound for f^k which is tight for the target selected by each attacker k. Expression (4) and (5) define the coverage probabilities c_j. Expression (6) states that each attacker selects a pure strategy. For each type of attacker, expression (7) states that the target attacked maximizes their expected utility.

An extension to the heterogeneous case is stated by introducing variables $c_{\omega j}$ satisfying

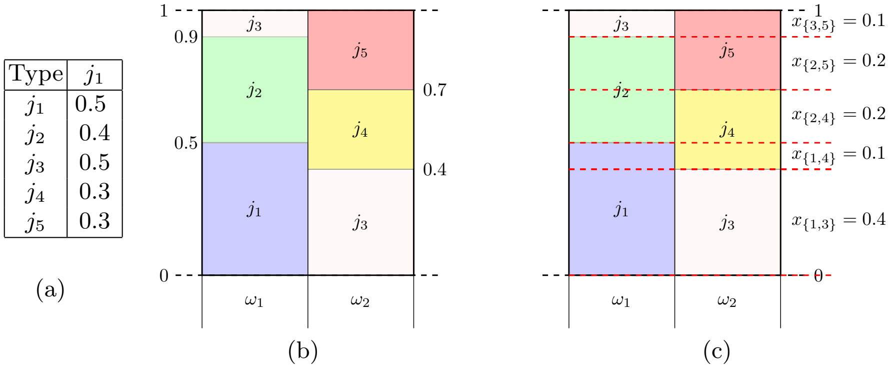
$$c_j = \sum_{\omega \in \Omega : j \in J_\omega} c_{\omega j} \quad j \in J, \tag{10}$$

and adding constraints limiting the amount of coverage for every single resource:

$$\sum_{j \in J_\omega} c_{\omega j} \leq 1 \quad \omega \in \Omega. \tag{11}$$

By solving these optimization problems, a coverage distribution is obtained. In order to obtain a mixed strategy x in the original space I that fits with (1) the following method described in Algorithm 1 could be applied. An example is presented in Figure 1.

```
Algorithm 1 Box Method
Require: $c \in \mathbb{R}^{|J|}$ feasible coverage.
  Step 1: For each resource $\omega \in \Omega$, consider a column of height 1.
  Step 2: For each resource $\omega$, fill up the column with the values of $c_j$, with $j \in J_\omega$
  Step 3: Define $x$ by extending each rectangle line into a horizontal line crossing all
  columns. The area between two horizontal lines represents a defender strategy. This
  area identifies a set of targets protected, at most one for each resource $\omega$. The height
  of this area represents the probability of the corresponding strategy.
```



Type	j_1
j_1	0.5
j_2	0.4
j_3	0.5
j_4	0.3
j_5	0.3

(a)

Fig. 1. Example of Algorithm 1 with 5 targets and two homogeneous resources. (a) Coverage probability (b) Step 1 and 2. (c) Step 3.

2.2 Population and unfair allocations

The optimal solution found through the optimisation problem stated above relies on how the payoffs matrices are computed. Anyway, discrimination issues can be performed by the security agency, putting less (or more) resources over some groups. We study two models to avoid such situation.

The first model considers that in each target $j \in J$ there is a population p_{jt} of type t between a set of possible population types T. Examples of T can include different races, religions, income level, among others. In order to avoid discrimination issues, it could be desirable that the total coverage allocated to each population type is proportional to the amount of inhabitants.

A second model can be developed with a slightly different description of the population. Instead of considering a fraction of population in each target, consider that each target has a label $\mathcal{L}$. Each label $\ell \in \mathcal{L}$ denotes the most representative population in the target. For example, the Latin, Asian and African areas in the main cities in Europe. Also, in some cities there is a division in High income, Medium class and Low income areas.

In the following example, we show that even in small instances, the problem of unfairness may occur. We will use this example along the article to introduce the main concepts of this work.

Example 1. Consider the following instance with five targets, $J = \{j_1, j_2, j_3, j_4, j_5\}$, three attacker types, $K = \{k_1, k_2, k_3\}$ and the total population divided in three types, $T = \{t_1, t_2, t_3\}$. Payoffs of the game and the description of the population in each target is represented in Table 1. We consider $\pi = (\pi_{k_1}, \pi_{k_2}, \pi_{k_3}) = (0.5, 0.3, 0.2)$ the distribution of probability over the set of the attackers. We aim to allocate two homogeneus resources, i.e., $m = 2$.

	k_1		k_2		k_3	
j	D^{k_1}	A^{k_1}	D^{k_2}	A^{k_2}	D^{k_3}	A^{k_3}
j_1	0 -6	34 -30	42 -9	15 -44	32 -29	49 -33
j_2	11 -26	9 -16	47 -3	12 0	37 -39	16 -48
j_3	2 -4	11 -39	9 -11	15 -16	25 -47	26 -5
j_4	0 -35	3 -11	37 -6	22 -32	29 -23	21 -48
j_5	9 -28	20 -48	47 -33	7 -14	42 -25	30 -40

(a)

Type	j_1	j_2	j_3	j_4	j_5	Total
t_1	10	50	10	70	0	14%
t_2	100	100	100	10	50	36%
t_3	270	0	50	0	180	50%
Label	l_3	l_2	l_2	l_1	l_3	

(b)

Table 1. Description of Example 1. a) Payoffs: Each row represent a target j. For each attacker type it is described $D^k = (D^k(j|p), D^k(j|u))$ and $D^k = (A^k(j|p), A^k(j|u))$. b) Description of the population detailed by types and labels.

In the last row of Table 1.(b) a label ℓ for each target is stated. In this example, a target is labeled by l_i if the population type t_i is the most representative. We label each target in $\mathcal{L} = \{\ell_1, \ell_2, \ell_3\}$ corresponding with the most representative population type. In other words, $\ell^j = \ell_i$ if $i = \arg_{t_i \in T} \max p_{jt_i}$. It would be desirable that the coverage that each part of the population receives is proportional to the total population that they represent.

Table 2.(a) shows the coverage given by model (HOM) and Table 2.(b) shows how this coverage is allocated to each part of the population. The proportional

Target	j_1	j_2	j_3	j_4	j_5
c_j	0.416	0.675	0.358	0.335	0.216

(a)

Type	c_t	% resources	Deviation
t_1	0.552	27.6 %	97.143%
t_2	0.872	43.6%	21.111%
t_3	0.576	28.8%	-42.4%

(b)

Label	c_l	% resources	Deviation
ℓ_1	0.335	16.75 %	-16.25%
ℓ_2	1.033	51.65%	29.125%
ℓ_3	0.632	28.8%	-21.0%

(c)

Table 2. Description of the solution in Example 1 in terms of population: (a) Coverage in the Strong Stackelberg equilibrium. (b) Coverage received by each type of population and the deviation from the fraction of population that they represent. (c) Coverage received by each labeled target.

coverage on population type t is computed as $c_t = \sum_{j \in J} c_j \frac{p_{jt}}{\sum_{t' \in T} p_{jt'}}$. Population type t_1 receives proportionally 97.143% more coverage than the population that they represent while population type t_3 receives a -42.4% less. Note that ℓ_1 represents the 20% of the targets, and ℓ_2 and ℓ_3 the 40% each one. In Table 2.(c), we show what is the proportion of resources allocated to each label. In this instance, both evaluations give an unbalanced allocation of the resources.

2.3 Random Allocations, Implementability and Bi-hierarchies.

We now introduce some mathematical notions that allow us to develop the models and results of this work. The problem of finding an SSE can be seen as finding a specific set of random allocations between resources Ω and a set of targets J under some considerations. Then, the coverage vector $c_{\omega j}$ can be seen as a random allocation as presented in [3]. They study constraints of the type:

$$q_S^L \leq \sum_{(\omega,j) \in S} c_{\omega j} \leq q_S^U \quad S \subseteq \Omega \times J,$$

where S is called constraint set and q_S^L, q_S^U are positive integers named quotas on S. The full of set of constraints is named constraint structure $\mathcal{H}$.

A random allocation c is *implementable* under the constraint structure $\mathcal{H}$ and quotas $\mathbf{q} = \{(q_S^L, q_S^U)\}_{S \in \mathcal{H}}$ if it can be written as a convex combination of pure allocations feasible under $\mathcal{H}$ and quotas $\mathbf{q}$. Constraint structure $\mathcal{H}$ is *universally implementable* if, for any integer quotas $\mathbf{q}$, every random allocation satisfying constraints in $\mathcal{H}$ is implementable under $\mathbf{q}$.

A constraint structure is a *hierarchy* if for any pair S, S' either $S \subseteq S'$ or $S' \subseteq S$ or $S \cap S' = \emptyset$. A constraint structure $\mathcal{H}$ is a *bihierarchy* if it can be partitioned in two hierarchies $\mathcal{H}_1$ and $\mathcal{H}_2$, that is, $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$ and $\mathcal{H}_1 \cup \mathcal{H}_2 = \mathcal{H}$.

Necessary and sufficient conditions over the constraint structure and implementability are given by [3] through the following two theorems:

Theorem 1. *(Sufficiency) If a constraint structure is a bihierarchy and quotas* $\mathbf{q}$ *are integers, then it is universally implementable.*

Theorem 2. *(Necessity) If a constraint structure contains all the rows and columns constraints and is not a bihierarchy, then it is not universally implementable.*

In the SSG context, the implementability of a set of constraints means that each coverage distribution feasible can be decomposed in a convex combination of pure strategies, or pure patrols, all of them satisfying the constraints. Note that if we only restrict to allocate resources to targets, the set of constraints forms a bihierarchy and all the coverage distributions c are implementable.

3 Models

In this section we discuss about how to restrict the coverage distributions taking into account the issues described in Section 2.2. We describe two ways of modelling fairness constraints: First we model constraints with a detailed description of the population; Then, we restrict the possible coverage considering aggregated information in terms of labels in each target. By doing this, we generate coverage probabilities that are not significantly correlated with sensitive features, as race, income level, etc.

3.1 Focus on the population

In this setting we assume a description of the population in each target j, given by the percentage of population of type $t \in T$ denoted by p_{jt} . In order to avoid discrimination issues, we might consider to restrict the amount of coverage that each population receives.

That is, the coverage vector c should satisfy

$$q_t^L \leq \sum_{j \in J} c_j \tilde{p}_{jt} \leq q_t^U \quad t \in T. \tag{12}$$

where $q = (q_t^L, q_t^U)$ are the quotas of the total coverage performed over the population $t \in T$.We denote $\tilde{p}_{jt} = \frac{p_{jt}}{\sum_{t' \in T} p_{jt'}}$ the fraction of population type t in target j. In this work, we use quotas of the form:

$$q_t^L = (1-\alpha) m \sum_{j \in J} \tilde{p}_{jt} \qquad q_t^U = (1+\alpha) m \sum_{j \in J} \tilde{p}_{jt} \tag{13}$$

where α is the maximum acceptable percentage of deviation from the total fraction of the population t multiplied by the number of the resources available (e.g., 10%).

In constraints (12), we assume that the total coverage inside a target is distributed proportionally to each population type. This assumption can be relaxed, through introducing some nonlinear relationship between the coverage and the probability of being covered given the composition of the population inside target j. Anyway, this topic is out of the scope of this paper, but it is still an interesting research question.

As we mentioned before, constraints (12) are not universally implementable. This is mainly because these constraints do not induce integer extreme points. In consequence, the vector coverage c under its form can not be decomposed in terms of pure strategies satisfying constraints (12). We will discuss how to deal with this issue in Section 4.

3.2 Label focus

A different approach can be stated as follows. There exists for each target a label $\ell \in \mathcal{L}$, representing a type of population representative in that target. In that case, we should think about protecting targets with an amount of coverage proportional to the percentage of the population they represent. Information about population is aggregated in each target, and in consequence, it is a more relaxed way of modelling fairness constraints.

Formally, define J_ℓ the targets labeled by ℓ. Note that $\{J_\ell\}_{\ell \in \mathcal{L}}$ defines a partition of J. For each label ℓ, there is a minimum and a maximum number of resources assigned to protect zones in J_ℓ, denoted again by q_ℓ^L and q_ℓ^U respectively. Then, constraints on the amount of coverage can be stated as:

$$q_\ell^L \leq \sum_{j \in J_\ell} c_j \leq q_\ell^U \quad \ell \in \mathcal{L}. \tag{14}$$

where the quotas $\mathbf{q}$ are given by:

$$q_\ell^L = \left\lfloor (1-\alpha) m \frac{|J_\ell|}{|J|} \right\rfloor \qquad q_\ell^U = \left\lceil (1+\alpha) m \frac{|J_\ell|}{|J|} \right\rceil \tag{15}$$

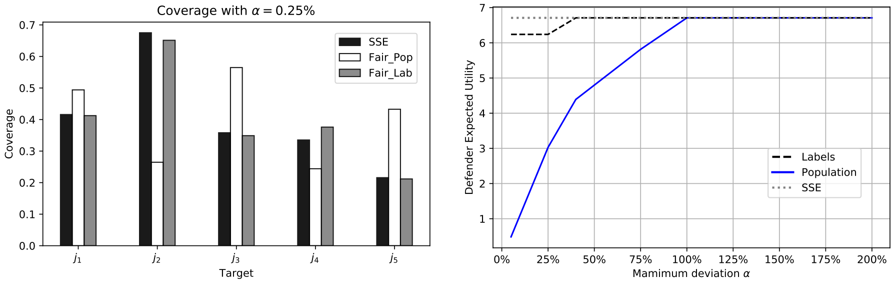
In this case, we use integer quotas to establish the implementability result in Theorem 3. In the model focused on the population, even using integer quotas the implementability result does not hold.

Example 1 continued Consider $\alpha = 25\%$. In Table 3, we show the bounds for each type of population and labels for both models respectively. Figure 2(a) shows the optimal coverage in each target by including the fairness constraints in both models and the coverage given by (HOM). For a fixed α, the model on labels is less restrictive than the one focused on the population. This explains the difference between the optimal coverage given by (HOM). By the same reason, the difference in terms of Defender Expected utility for different values of α for the model of population is greater than the model with labels, as is shown in Figure 2(b).

The optimal coverage, in both cases should be implemented by sampling pure strategies. We showed in Table 4, one possible decomposition of these solutions

Type	q_t^L	q_t^U	Label	q_ℓ^L	q_ℓ^U
t_1	0.21	0.35	ℓ_1	0	1
t_2	0.54	0.9	ℓ_2	0	1
t_3	0.75	1.25	ℓ_3	0	1

Table 3. Lower and upper bounds in the coverage for both models when ($\alpha = 0.25$)



(a) (b)

Fig. 2. (a) Comparison of the optimal coverage without any fairness consideration and the models focused on Labels and Population. (b) Defender expected utility in function of α.

using Algorithm 1. Note that strategies in model focused on population are not implementable. In particular, the third strategy in Table 4.(a) that covers targets j_2 and j_4 allocates 0 resources to the population type t_3.

		λ_1	λ_2	λ_3	λ_4	λ_5
Target	c^{pop}	0.324	0.17	0.074	0.191	0.241
j_1	0.494	1	1	0	0	0
j_2	0.265	0	0	1	1	0
j_3	0.565	1	0	0	0	1
j_4	0.244	0	1	1	0	0
j_5	0.432	0	0	0	1	1

(a)

		λ_1	λ_2	λ_3	λ_4
Target	c^{lab}	0.063	0.349	0.376	0.212
j_1	0.412	1	1	0	0
j_2	0.651	1	0	1	1
j_3	0.349	0	1	0	0
j_4	0.376	0	0	1	0
j_5	0.212	0	0	0	1

(b)

Table 4. Decomposition for the optimal coverage in (a) model focused on population and (b) model focused on labels, using Algorithm 1. In both cases we use $\alpha = 25\%$.

4 Implementability and extensions

In this section we discuss the implementability of both models. First, we show that if quotas used in the model of labels are integers, then every coverage satisfying fairness constraints are implementable. Then, we discuss how to find pure strategies that are the closest to the set of fairness constraints in the model of population and that fits with the coverage distribution. Finally, we discuss about some extensions of the models and when they preserve this property.

4.1 Labels are implementable, Population constraints do not

Now we show that the model focused on labels is implementable.

Theorem 3. *If for each $\ell \in \mathcal{L}$, q_ℓ^L, q_ℓ^U are integers, then conditions (14) are universally implementable.*

Proof. We prove that the set of constraints forms a bihierarchy in the problem of allocation targets to resources. In particular, any vector coverage satisfies:

$$
\begin{aligned}
0 &\leq c_{\omega j} \leq 1 && \omega \in \Omega, j \in A(\omega) && (16)\\
0 &\leq c_{\omega j} \leq 0 && \omega \in \Omega, j \notin A(\omega) && (17)\\
0 &\leq \textstyle\sum_{j \in J} c_{\omega j} \leq 1 && \omega \in \Omega && (18)\\
0 &\leq \textstyle\sum_{\omega \in \Omega} c_{\omega j} \leq 1 && j \in J && (19)\\
q_\ell^L &\leq \textstyle\sum_{\omega \in \Omega} \sum_{j \in J_\ell} c_{\omega j} \leq q_\ell^U && \ell \in \mathcal{L}. && (20)
\end{aligned}
$$

Conditions (16) and (17) represent singletons $\{(\omega, j)\} \in \Omega \times J$. Condition (18) can be represented as the sets $\{\omega\} \times J$. Condition (19) is represented by $\Omega \times \{j\}$. Finally, conditions (20) can be represented as the sets $\{(\omega, j) : j \in J_\ell, \omega \in \Omega\}$. We show that by grouping conditions (16), (17) and (18), and grouping conditions (19) and (20), they form a bihierarchy. Formally, we define the following sets:

$$
\begin{aligned}
\mathcal{H}_1 &= \{\{(\omega, j)\} \in \Omega \times J\} \cup \{\{\omega\} \times J : \omega \in \Omega\}\\
\mathcal{H}_2 &= \{\Omega \times \{j\} : j \in J\} \cup \{\{(\omega, j) : j \in J_\ell, \omega \in \Omega\} : \ell \in \mathcal{L}\}
\end{aligned}
$$

First, we show $\mathcal{H}_1$ is a hierarchy: Clearly each pair of singleton are disjoint. The same occurs with each pair of elements of $\{\{\omega\} \times J : \omega \in \Omega\}$. On the other hand, each singleton (ω, j) is, either included in the set $\{\omega\} \times J$ and it has empty intersection with any other set $\{\omega'\} \times J$, with $\omega' \neq \omega$. Then, $\mathcal{H}_1$ is a hierarchy.

Now we prove that $\mathcal{H}_2$ is a hierarchy. Each pair of elements in $\{\Omega \times \{j\} : j \in J\}$ are disjoint. The same occurs with each pair of elements in $\{\{(\omega, j) : j \in J_\ell, \omega \in \Omega\} : \ell \in \mathcal{L}\}$ because $\{J_\ell\}$ induces a partition over J. Now, if we take a pair of element of each group of conditions, for index j and ℓ, there are two cases: Either $j \in J_\ell$, in that case $\Omega \times \{j\} \subseteq \{(\omega, j) : j \in J_\ell, \omega \in \Omega\}$; or $j \notin J_\ell$, in that case $\Omega \times \{j\} \cap \{(\omega, j) : j \in J_\ell, \omega \in \Omega\} = \emptyset$. Then, $\mathcal{H}_2$ is also a hierarchy.

Clearly $\mathcal{H}_1 \cap \mathcal{H}_2 = \emptyset$. Then, the set of conditions $\mathcal{B} = \mathcal{H}_1 \cup \mathcal{H}_2$, forms a bihierarchy, and then under any integer quotas, the expected allocation is implementable and the result follows. A graphical representation of the proof is shown in Figure 3.

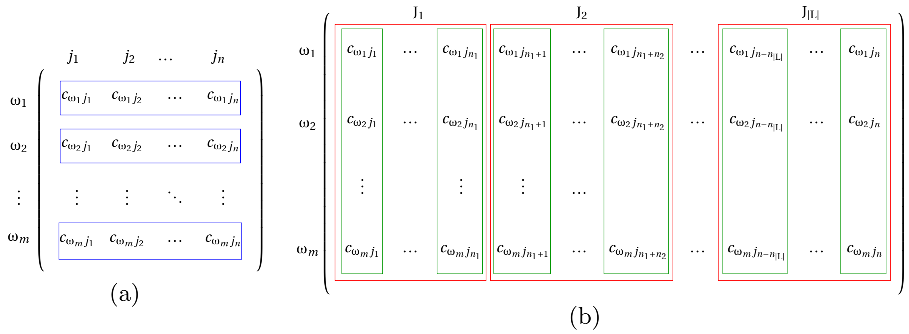


Fig. 3. Representation of the bihierarchy. (a) $\mathcal{H}_1$ consisting in constraints (18) and singletons. (b) $\mathcal{H}_2$ consisting in constraints (19) and (20).

4.2 Approximating the Implementability

Coverage frequencies in the model focused on population are not implementable in general. That means that they cannot be decomposed in pure strategies satisfying the fairness constraints (12). In any case, they can be decomposed in pure strategies in the original set of strategies, that is, pure strategies covering at most m targets. This can be preformed by Algorithm 1. On one hand, this algorithm generates different decompositions by considering different orders in which the set of targets are included. For instance, in Figure 1, targets were included in lexicographic order. If the algorithm considers order j_2, j_3, j_1, j_4, j_5, the output probabilities x will be different. On the other hand, Algorithm 1 does not take into account if the strategies produced satisfy conditions such as fairness, or are close to satisfy them. We would like to get a decomposition such that is the fairest as possible, allocating low probability to strategies that are unfair and high probabilities to the fairest ones.

Formally, we have a polyhedron P_1 as the convex hull of the binary encoding of the set I of pure strategies. Let P_2 be the polyhedron of all the coverage vectors satisfying the fairness constraint in the model focused on population, that is $P_2 = \{c \in P_1 : c \text{ satisfies } (12)\}$. We want to find a convex decomposition of a point in P_2 in terms of vertices of P_1 such that the weighted sum of the violations of constraints (12) of each strategy is minimised. The weights used in this optimisation problem come from the convex decomposition. By doing this, we aim to achieve a set of strategies implementing the optimal fair coverage, but at the same time, allocating low probability to the strategies that are unfair.

Now we present some models to find such decomposition. We formulate the following non-linear model in order to get a decomposition in pure strategies where each pure strategy minimises the violation of the constraint (12). Let $\mathcal{M} = \{1, \ldots, U\}$ be a set of indices, where U is an upper bound on the number of strategies needed to decompose c in terms of pures strategies in I. In our first model, we create a vector of variables a_i as the binary encoding of strategy

$i \in \mathcal{M}$, where $a_{ij} = 1$ if the target j is covered by strategy i. The formulation is as follow:

$$\min_{a,\lambda,\epsilon} \qquad \sum_{i\in\mathcal{M}} \sum_{t\in T} \lambda_i \epsilon_{it} \tag{21}$$

$$\sum_{i\in\mathcal{M}} \lambda_i = 1 \tag{22}$$

$$\sum_{i\in\mathcal{M}} \lambda_i a_{ij} = c_j \qquad j \in J \tag{23}$$

$$\sum_{j\in J} a_{ij} \leq m \qquad i \in \mathcal{M} \tag{24}$$

$$-\epsilon_{it} + q_t^L \leq \sum_{j\in J} a_{ij}\tilde{p}_{jt} \leq q_t^U + \epsilon_{it} \; t \in T, i \in \mathcal{M} \tag{25}$$

$$a \in \{0,1\}^{|J||\mathcal{M}|},\ \lambda \in \mathbb{R}^{|\mathcal{M}|} \tag{26}$$

$$\epsilon \geq 0 \tag{27}$$

where λ_i is the weight in the decomposition and ϵ_t measures the violation of the fairness constraint for each strategy. The objective function (21) minimizes the wheigted violation of constraints (25). Equation (22) states that the weights must sum to 1 and Equation (23) ensures that the convex combination fits with c. constraints (24) and (26) define the pure strategy i. Expression (25) and the fact that $\epsilon_t \geq 0$ defines the maximum deviation for each population type.

This formulation is a non-convex mixed integer non-linear problem untractable even for small instances. We linearize this model by introducing variables $\gamma_{ij} = \lambda_i a_{ij}$ and $\mu_{it} = \lambda_i \epsilon_{it}$, and re-scaling constraints (25) by λ_i. This mixed integer linear problem (MILP) is:

$$\text{(DEC)} \ \min_{a,\gamma,\mu} \qquad \sum_{i\in\mathcal{M}} \sum_{t\in T} \mu_{it} \tag{28}$$

$$\text{constraints (22), (24), (26)}$$

$$\sum_{i\in\mathcal{M}} \gamma_{ij} = c_j \qquad j \in J \tag{29}$$

$$-\mu_{it} + \lambda_i q_t^L \leq \sum_{j\in J} \gamma_{ij}\tilde{p}_{jt} \leq \lambda_i q_t^U + \mu_{it} \; t \in T, i \in \mathcal{M} \tag{30}$$

$$\gamma_{ij} \leq \lambda_i \qquad j \in J,\ i \in \mathcal{M} \tag{31}$$

$$\gamma_{ij} \leq a_{ij} \qquad j \in J,\ i \in \mathcal{M} \tag{32}$$

$$a_{ij} + \lambda_i - 1 \leq \gamma_{ij} \qquad j \in J,\ i \in \mathcal{M} \tag{33}$$

$$\mu \geq 0 \tag{34}$$

where constraints (31), (32) and (33) defines the product $\lambda_i a_{ij}$. We name this MILP (DEC). This formulation has two main drawbacks. First, we have to know a priori an upper-bound of the number of strategies to achieve the best decomposition. Secondly, the linear relaxation has always optimal value equal to zero. This means the formulation is a weak formulation and in consequence algorithms for solving MILP implemented in commercial optimization software might perform very poorly. We will show this issue in the next section.

In order to solve this problem in an efficient way, we propose the following column generation algorithm. Consider a set of feasible strategies. For a given strategy, it is straightforward to compute the violation of fairness constraints. We denote the violation of strategy a^i by $v(a^i)$. Then, we state the following linear optimisation problem (MP):

$$\begin{aligned} \text{(MP)} \min_{\lambda} \quad & \textstyle\sum_{i \in \mathcal{M}} \lambda_i v(a^i) && (35)\\ \text{s.t.} \quad & \textstyle\sum_{i \in \mathcal{M}} \lambda_i a^i = c && (36)\\ & \textstyle\sum_{i \in \mathcal{M}} \lambda_i = 1 && (37)\\ & \lambda_i \geq 0 && (38) \end{aligned}$$

with the difference, that here a^i and $v(a^i)$ are parameters known for the problem. We denote $\alpha \in \mathbb{R}^{|J|}$ and $\beta \in \mathbb{R}$ the dual variables associated to constraints (36) and (37) respectively. The column generation algorithm works as follow: First choose a set of feasible set of pure strategies. They can be retrieved using Algorithm 1. Solve (MP) and get the dual variables. Then, compute the most negative reduced cost of a possible new strategy. This optimization problem, called column generator (CG), is stated as follow:

$$\begin{aligned} \text{(CG)} \min_{a} \quad & \textstyle\sum_{t \in T} v_t(a) - \sum_{j \in J} \alpha_j a_j - \beta && (39)\\ \text{s.t.} \quad & \textstyle\sum_{j \in J} a_j \leq m && (40)\\ & -v_t(a) + q_t^L \leq \textstyle\sum_{j \in J} a_j \tilde{p}_{jt} \leq q_t^U + v_t(a) \; t \in T, i \in \mathcal{M} && (41)\\ & a \in \{0,1\} && (42) \end{aligned}$$

where the objective function minimizes the reduced cost of the new strategy generated. If the new strategy generated by (CG) has a positive reduced cost, then the algorithm stops and the optimal solution of the master is the optimal solution of the whole problem. Otherwise, a new strategy is added to $\mathcal{M}$ with cost $v(a) = \sum_{t \in T} v_t(a)$.

Example 1 continued. Now we decompose the coverage c^{pop} using formulation (DEC). The decomposition is showed in Table 5. The distance decreases from 0.284 to 0.258. Note that the strategy that covers targets j_2 and j_4 (allocating 0 resources to the population type t_3) is not present anymore.

		λ_1	λ_2	λ_3	λ_4	λ_5
Target	c^{pop}	0.491	0.168	0.265	0.073	0.003
j_1	0.494	1	0	0	0	1
j_2	0.265	0	0	1	0	0
j_3	0.565	1	0	0	1	0
j_4	0.244	0	1	0	1	1
j_5	0.432	0	1	1	0	0

Table 5. Decomposition of c^{pop} using (DEC).

4.3 Some Extensions

Here we present two extensions to the discussion in this topic:

Multiple Labels: We consider the setting where each target has multiple labels representing different dimensions of analysis. It could be the case, where each target is characterised by race, religion, wealth, etc. Consider the set of attributes labels $\mathcal{A} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_{|\mathcal{A}|}\}$, , with its corresponding set of labels. Similarly to (15), quotas $(q^L_{\ell^k}, q^U_{\ell^k})$ on each label are stated, for each label ℓ^k in the k-th attribute.

We would like to state an extension of Theorem 3 for this setting. If the labels satisfy that $J_{\ell^k} \subseteq J_{\ell^{k'}}$ or $J_{\ell^k} \cap J_{\ell^{k'}} = \emptyset$ for each $\ell^k \in \mathcal{L}_k$ and $\ell^{k'} \in \mathcal{L}_{k'}$, then constraints forms a hierarchy. Then, Theorem 3 applies directly.

If it is not the case, we define $\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2 \times \ldots \times \mathcal{L}_{|\mathcal{A}|}$ which clearly induces a partition over J. If we set quotas as in (15) the result will be implementable under labels $\mathcal{L}$, but is a relaxation of the original problem. We would need to find quotas (q^L_ℓ, q^U_ℓ) for each ℓ, that produces the same constraints that quotas $(q^L_{\ell^k}, q^U_{\ell^k})$. If we are able to find integer quotas satysfying this, then Theorem 3 applies again.

Penalizing violations with different weights: As in the Example 1, maybe it could be the case that decision makers would not prefer strategies that do not cover one type of population at all. The model (DEC) and the column generation, penalizes in the same way strategies that cover more than the quotas and the ones that cover less. This symmetry can be broken by introducing different variables to measure the violation, ϵ^- and ϵ^+, and replacing in the objective function (21) and constraints (25) by:

$$\min_{a,\lambda,\epsilon^-,\epsilon^+} \quad \textstyle\sum_{i\in\mathcal{M}} \sum_{t\in T} \lambda_i(\kappa_t \epsilon^-_{it} + \epsilon^+_{it}) \tag{43}$$

$$-\epsilon^-_{it} + q^L_t \leq \textstyle\sum_{j\in J} a_{ij}\tilde{p}_{jt} \leq q^U_t + \epsilon^+_{it} \; t \in T, i \in \mathcal{M} \tag{44}$$

where $\kappa_t > 1$ is a parameter assigning more weight to under allocate protection to population type t. Linearization techniques and column generation can be straight applied as before.

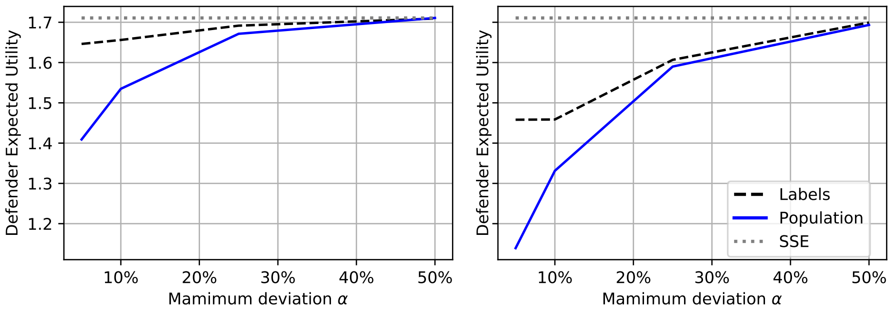
5 Computational Experiments

In our computational results we investigate three questions: First, the impact in the defender expected utility by including fairness considerations in SSGs. Secondly, we test the computational performance of (DEC) and the column generation approach in order to get strategies close to be implementable. Finally, we test how much we win by decomposing coverage probabilities in almost-implementable pure strategies with our method instead with the Box method presented in Algorithm 1.

Experimental Setting: We test our methods in randomly generated games. For each $n \in \{20, 30, 40\}$, $m \in \{5, 10\}$ and $\{1, 3\}$ we generate 10 instances. Payoff matrices were generated uniformly such that $D(j|p), A(j|u) \sim U(0, 100)$ and $D(j|u), A(j|p) \sim U(-100, 0)$. For each game generated, an amount of 1000 "inhabitants" were allocated among the targets. Finally, we divided that population in $|T| = 3$ or $|T| = 7$ types, by running random partitions that sum the total population allocated in each target. With this setting, we aim to obtain defender's expected rewards comparables when population is divided in 3 or 7 types.

All experiments have been carried out using CPLEX 12.8 and Python 3.6, in a single thread on a server with a 3.40Ghz Intel i7 processor and 64 GB of memory.

Defender's Expected Utility: In order to measure the impact, we run the models of Population and Labels with different values of $\alpha \in 5\%, 10\%, 25\%, 50\%$. In Figure 4, we show the average defender's expected utility in function of α, and separated by T. The model that consider labels in the targets is less restrictive, thus for a fixed α returns a bigger defender expected utility. Logically, both of them are upper bounded by the SSE returned by (HOM). Also, if we consider a more detailed description of the population (i.e. more types), the defender expected utility decreases. Finally, as we increase α, the defender expected utility for the three models converges to the SSE value.



(a)

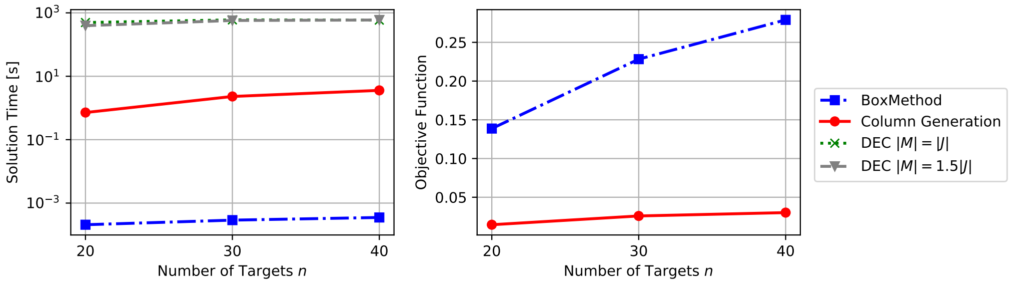
(b)

Fig. 4. Average defender's expected utility in function of α. (a) $|T| = 3$ (b) $|T| = 7$

Efficiency of Decomposition Methods: For $\alpha = 25\%$, we solve all the instances with the model focused on population. We test the MILP (DEC) and the column generation. For the formulation (DEC), we use an upper bound on the number of strategies necessaries to decompose the coverages equal to $U = |J|$ and $U = 1.5|J|$. We compare them with the output of Algorithm 1, using a lexicographic order.

We use a time limit of 600 seconds for the algorithms. Figure 5.a shows the average runtime in logarithmic scale. (DEC) considering $U = |J|$, respectively $U = 1.5|J|$ hits the time limit in 97%, resp. 99%, of the instances. Algorithm 1 takes less than 1 mili-second in being performed. The Column Generation takes between 0.75 seconds and 4 seconds.

In figure 5.b, we compare the minimimum weighted violation and the weighted violation of the solution generated by Algorithm 1. Algorithm 1 returns consistently a solution that violates the more compared to the solution returned by the Column Generation. We note that the minimum weighted violation decreases as the size of the instance increase. From the practical point of view, the models that miminizes the weighted violation generates fairer allocations in reasonable time.



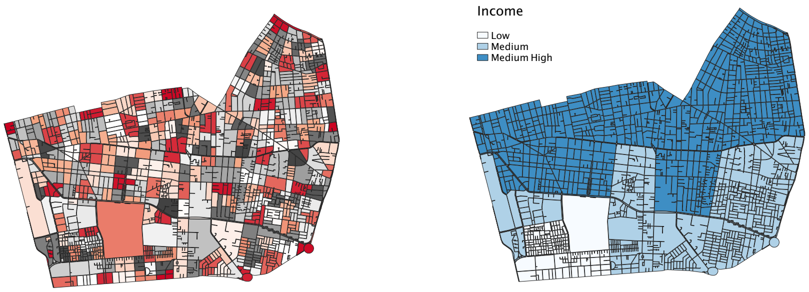
(a) (b)

Fig. 5. (a) Solution time for different methods to decompose coverage c in implementable strategies x. (b) Weighted violation of fairness constraints for each method.

6 Case Study

In this section we study the performance of our models in a realistic instance, based on data retrieved from Ñuñoa, a municipality in Santiago of Chile. This instance consists in 1266 census blocks where the data considered is level of income (medium high, medium, low), the demand of policial resources DEM, the population in each census block and a measure of the criminal activity (amount of reported crime RC). The demand of policial resources is computed by Carabineros de Chile, the national police of Chile, as the amount of resources necessary to do general deployments, court orders, monitoring establishments and extraordinary services as in [2]. To consider one target for each block is expensive to solve in terms of computational times, so we apply the clustering algorithm integrated in QGIS [14], and we aggregate the data to reduce the size of the problem. We finally consider 250 targets represented in Figure 6.(a).

In this municipality there are three types of population in terms of level of education and income (strongly correlated). People with low income, medium income and high income, who receive 432US\$, 750US\$ and 1400\$ per month. The geographical distribution of the income level is represented in Figure 6.(b).



(a) Targets (b) Income level

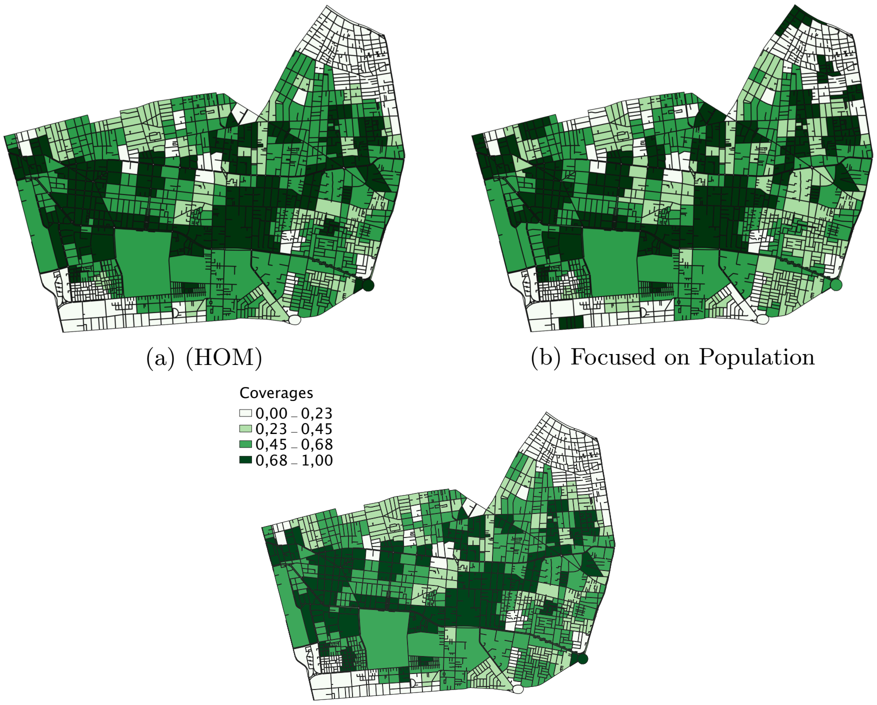
Fig. 6. Data from Ñuñoa

We consider two types of attackers: the first one, named k_1, related to the criminal activity and the second one, named k_2, related to the general activities that Carabineros perform. Both of them will appear with the same frequency, so $\pi_{k_1} = \pi_{k_2} = 0.5$. Payoffs are built proportionally to DEM and RC. The penalties for the attacker where used as a constant for each type of attacker (the punishment will be equal no matter where the attacker is caught). The specific parameters are shown in Table 6.

$D^{k_1}(j\|p)$	$D^{k_1}(j\|u)$	$A^{k_1}(j\|p)$	$A^{k_1}(j\|u)$	$D^{k_2}(j\|p)$	$D^{k_2}(j\|u)$	$A^{k_2}(j\|p)$	$A^{k_2}(j\|u)$
0	$-DEM_j$	DEM_j	-100	0	$-RC_j$	RC_j	-300

Table 6. Payoff matrices built in the case study.

We test the model (HOM) (without any fairness consideration) and the models focused on population and labels. We deploy 120 homogoneus policial resources and we use the fairness parameter $\alpha = 0.1$. All models took less than 1 minute to return the optimal coverage distribution. Results are shown in Figure 7. The model focused on the population allocates resources where the model without fairness consideration does not. Both, high and low income areas which, in model HOM, were not covered now present patrol assigments. The model focused on labels, on the other hand, presents a result with similar behaviour as HOM. Even when results are comparable, these models become a useful tool to ensure a fair distribution of police resources without having significant impact on its optimal allocation. This could be important for both police decision making and governmental policies.



(a) (HOM)

(b) Focused on Population

(c) Focused on Labels

Fig. 7. Coverage distribution in Ñuñoa.

7 Conclusions

In this paper, we have studied the impact of fairness constraints in SSG. We have presented two models: one imposing fairness constraints in a detailed description of the population, the other, imposing constraints in targets that are labeled. These models aim to allocate fair distribution of resources amongst the population, avoiding discrimination issues from a tactical point of view. We study our model on a realistic instance.

We have shown that imposing constraints with labels on the targets is implementable, meaning that each coverage distribution satisfying these constraints can be decomposed in pure strategies, all of them satisfying these constraints. This is not the case with the model with the detailed description of the population. In this case, we propose a MILP formulation to find the decomposition that is closest to the set of fairness constraints. Also we propose a column generation method to solve this problem efficiently. Computational tests have shown that the column generation approach finds efficiently the best decomposition.

References

1. Bucarey, V., Casorrán, C., Figueroa, Ó., Rosas, K., Navarrete, H., Ordóñez, F.: Building real stackelberg security games for border patrols. In: International Conference on Decision and Game Theory for Security. pp. 193–212 (2017)
2. Bucarey, V., Ordóñez, F., Bassaletti, E.: Shape and balance in police districting. In: Applications of Location Analysis, pp. 329–347. Springer (2015)
3. Budish, E., Che, Y.K., Kojima, F., Milgrom, P.: Designing random allocation mechanisms: Theory and applications. American Economic Review **103**(2), 585–623 (2013)
4. Casorrán, C., Fortz, B., Labbé, M., Ordóñez, F.: A study of general and security stackelberg game formulations. European Journal of Operational Research (2019)
5. Correa, J., Harks, T., Kreuzen, V.J., Matuschke, J.: Fare evasion in transit networks. Operations Research **65**(1), 165–183 (2017)
6. Hargreaves, J., Linehan, C., Husband, H.: Police Powers and Procedures, England and Wales, Year Ending 31 March 2017. Home Office (2017)
7. Jain, M., Kardes, E., Kiekintveld, C., Ordonez, F., Tambe, M.: Security games with arbitrary schedules: A branch and price approach. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
8. Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., Tambe, M.: Computing optimal randomized resource allocations for massive security games. In: AAMAS 2008. pp. 689–696 (2009)
9. Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordonez, F., Kraus, S.: Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In: AAMAS 2008. pp. 895–902 (2008)
10. Pierson, E., Simoiu, C., Overgoor, J., Corbett-Davies, S., Ramachandran, V., Phillips, C., Goel, S.: A large-scale analysis of racial disparities in police stops across the united states. arXiv preprint arXiv:1706.05678 (2017)
11. Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., Kraus, S.: Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport. In: AAMAS 2008. pp. 125–132 (2008)
12. Rights, E.U.A.F.F.: Bigdata: Discrimination in data-supported decision making. FRA FOCUS (2018), https://fra.europa.eu/en/publication/2018/big-data-discrimination
13. 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: AAMAS 2012. pp. 13–20 (2012)
14. Team, Q.D., et al.: Qgis geographic information system. Open Source Geospatial Foundation Project (2016)
15. Zafar, M.B., Valera, I., Gomez-Rodriguez, M., Gummadi, K.P.: Fairness constraints: A flexible approach for fair classification. Journal of Machine Learning Research **20**(75), 1–42 (2019), http://jmlr.org/papers/v20/18-262.html