

Chapter 2

Data Acquisition: Towards Optimal Use of Sensors

Abstract The main source of knowledge is processing data. Data comes from sensors. Within a limited budget, it is extremely important to make sure that the use of the sensors is optimized so that we get the largest possible amount of useful data from these sensors. Traditionally, most data comes from stationary sensors, i.e., sensors which we place at fixed locations. For such sensors, it is important to come up with the optimal placement, the placement which would lead to the largest amount of useful data. We analyze this problem in Sect. 2.1, on the example of placing bio-weapon detectors, and in Sect. 2.2, on the example of placing environmental sensors. The problem of optimal use becomes more technically challenging if we take into account the possibility of using mobile sensors, i.e., sensors which we can move along different trajectories. In this case, it is important to come up with optimal trajectories, i.e., the trajectories which would lead to the largest amount of useful data. We analyze this problem in Sect. 2.3, on the example of Unmanned Aerial Vehicles (UAVs) patrolling the border. In all these cases, it is important to make sure that not only we have an algorithm producing the optimal placement or optimal trajectory: we also need to make sure that the corresponding algorithms are computationally efficient, i.e., that the corresponding optimization algorithms can produce the resulting optimal setting in reasonable time. The more sensors we need to place, the more computations we need and therefore, the more important it is for the computation time to be reasonable. This is especially important in situations of *big data*, when the amount of data is so huge that the traditional numerical methods are not applicable [1–4]. We analyze this problem in Sect. 2.4, again on the example of security problems.

2.1 Optimal Use of Stationary Sensors: Case Study of Optimal Placement of Bio-Weapon Detectors

In this section, we analyze the problem of the optimal use of stationary sensors, on the example of optimal placement of bio-weapon detectors. Biological weapons are difficult and expensive to detect. Within a limited budget, we can afford a limited number of bio-weapon detector stations. It is therefore important to find the optimal

locations for such stations. A natural idea is to place more detectors in the areas with more population—and fewer in desert areas, with fewer people. However, such a commonsense analysis does not tell us how many detectors to place where. To decide on the exact placement of bio-weapon detectors, we formulate the placement problem in precise terms, and come up with an (almost) explicit solution to the resulting optimization problem.

The results from this section were first published in [5].

Formulation of the practical problem. Biological weapons are difficult and expensive to detect. Within a limited budget, we can afford a limited number of bio-weapon detector stations. It is therefore important to find the optimal locations for such stations.

Commonsense analysis of the problem. A natural idea is to place more detectors in the areas with more population—and fewer in areas with fewer people, e.g., in the desert areas. However, such a commonsense analysis does not tell us how many detectors to place where. To decide on the exact placement of bio-weapon detectors, we must formulate the placement problem in precise terms.

Objective function. The above commonsense idea is based on a (reasonable) assumption that the adversary's objective is to kill as many people as possible. Vice versa, our objective is to minimize the potential effect of a bio-weapon attack.

Comment. In this chapter, we mainly concentrate on the above objective function. This objective function may not always fully describe the adversary's objectives. For example, one of the objectives of political terrorism may be extra publicity for the cause. From this viewpoint, an adversary may prefer a scenario with a smaller number of victims if several of these victims are well-known. It is therefore desirable to formulate the objective functions that describe this (and similar) approaches, and extend our optimization analysis to the case of such more complex objective functions.

Towards precise formulation of the problem: what is known. Since the objective is to target as many people as possible, to analyze this situation, we need to know how many people live at different locations. In precise terms, we assume that we know, for every possible location x , the population density $\rho(x)$ in the vicinity of this location.

We assume that we know the number N of detectors that we can afford to place in the given territory.

We also assume that we know the efficiency of a bio-weapons detector station. We will estimate this efficiency by the distance d_0 at which this station can detect an outbreak of a disease.

For many diseases, $d_0 = 0$ —we can only detect a disease when the sources of this disease reach the detecting station.

However, it is quite possible that for some diseases, we have a super-sensitive equipment that is able to detect the concentration of the bio-weapons agent at a level below the threshold that makes this agent dangerous to the population. In this case,

we can detect the coming disease before it starts affecting people in the direct vicinity of the station—i.e., in effect, we have $d_0 > 0$.

For simplicity, we assume that the disease spreads equally fast in all directions.

Comment. This is also a somewhat simplifying assumption, since in reality, a disease spreads

- either with human movements—in which case in the vicinity of an interstate it spreads faster in the direction of the interstate,
- or with wind—in which case it spreads faster in the direction of the prevailing winds.

How we can describe the detector placement. On a large-scale basis, we need to decide how many detectors to place in different areas. In other words, we need to find the *density* $\rho_d(x)$ of detector placement—the number of detectors per unit of area (e.g., a square mile).

Under this description, the number of detectors in an area of size Δx is approximately equal to $\rho_d(x) \cdot \Delta x$, so the overall number of detectors can be obtained by adding these amounts, as $\int \rho_d(x) dx$. Thus, the constraint that we have exactly N detecting stations can be described as

$$\int \rho_d(x) dx = N. \quad (2.1.1)$$

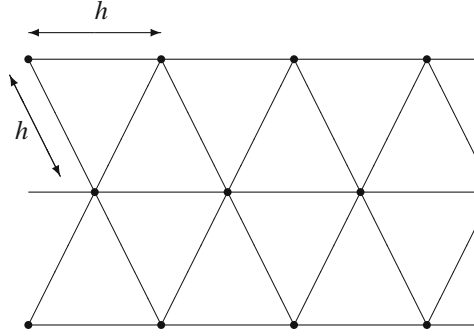
Optimal placement of sensors: at the vertices of a hexagonal grid. We want to place the sensors in such a way that the largest distance D to a sensor is as small as possible. Alternatively, if D is fixed, we want to minimize the number of sensors for which every point is at a distance $\leq D$ from one of the sensors. In geometric terms, this means that every point on a plane belongs to a circle of radius D centered on one of the sensors—and thus, the whole plane is covered by such circles. Out of all such coverings, we want to find the covering with the smallest possible number of sensors.

It is known that the smallest such number is provided by an equilateral triangle grid, i.e., a grid formed by equilateral triangles; see, e.g., [6]. Hence, in this section, we will select such a grid.

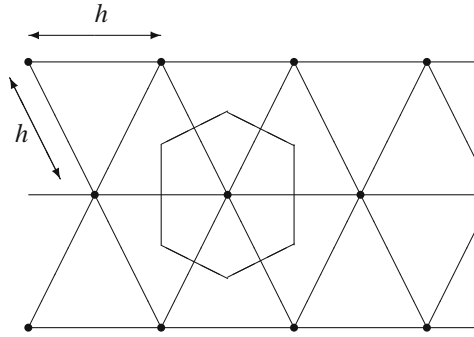
Locations of detector stations are assumed to be known to the adversary. Bio-weapon detector stations are not easily concealable. Thus, we assume that the adversary knows the locations of different stations.

How to estimate the effect of placing bio-weapons at a location x . Let us assume that we have already decided how many detectors to place in different regions, i.e., that we have already selected the density function $\rho_d(x)$.

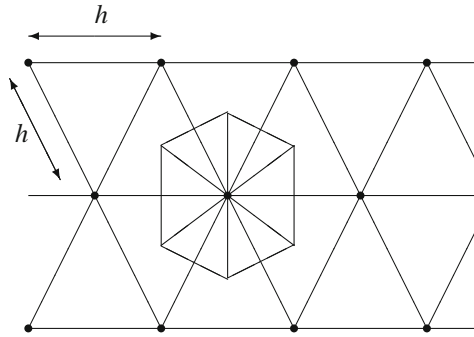
Within a small region of area A , we have $A \cdot \rho_d(x)$ detectors. Thus, if we, e.g., place these detectors on a grid with distance h between the two neighboring ones in each direction, we have:



For this placement, the set of all the points which are closest to a given detector forms a hexagonal area:



This hexagonal area consists of 6 equilateral triangles with height $h/2$:



In each triangle, the height $h/2$ is related to the size s by the formula

$$\frac{h}{2} = s \cdot \cos(60^\circ) = s \cdot \frac{\sqrt{3}}{2}, \quad (2.1.2)$$

hence

$$s = \frac{h}{\sqrt{3}} = h \cdot \frac{\sqrt{3}}{3}. \quad (2.1.3)$$

Thus, the area A_t of each triangle is equal to

$$A_t = \frac{1}{2} \cdot s \cdot \frac{h}{2} = \frac{1}{2} \cdot \frac{\sqrt{3}}{3} \cdot \frac{1}{2} \cdot h^2 = \frac{\sqrt{3}}{12} \cdot h^2. \quad (2.1.4)$$

So, the area A_s of the whole set is equal to 6 times the triangle area:

$$A_s = 6 \cdot A_t = \frac{\sqrt{3}}{2} \cdot h^2. \quad (2.1.5)$$

Each point from the region is the closest to one of the points from the detector grid, so the region of area A is thus divided into $A \cdot \rho_d(x)$ (practically) disjoint sets of area $\frac{\sqrt{3}}{2} \cdot h^2$. So, the area of the region is equal to the sum of the areas of these sets:

$$A = (A \cdot \rho_d(x)) \cdot \frac{\sqrt{3}}{2} \cdot h^2. \quad (2.1.6)$$

Dividing both sides of this equality by A , we conclude that

$$1 = \rho_d(x) \cdot \frac{\sqrt{3}}{2} \cdot h^2, \quad (2.1.7)$$

and hence, that

$$h = \frac{c_0}{\sqrt{\rho_d(x)}}, \quad (2.1.8)$$

where we denote

$$c_0 \stackrel{\text{def}}{=} \sqrt{\frac{2}{\sqrt{3}}}. \quad (2.1.9)$$

From the viewpoint of the adversary, it is desirable to place the bio-weapon at a location which is the farthest away from the detectors—so that it will take the longest time to be detected. For the grid placement, this location is at one of the vertices of the hexagonal zone—at which the distance from each neighboring detector is equal to $s = h \cdot \frac{\sqrt{3}}{3}$. By using formula (2.1.8), we can determine s in terms of $\rho_d(x)$, as

$$s = \frac{c_1}{\sqrt{\rho_d(x)}}, \quad (2.1.10)$$

where we denote

$$c_1 = \frac{\sqrt{3}}{3} \cdot c_0 = \frac{\sqrt[4]{3} \cdot \sqrt{2}}{3}. \quad (2.1.11)$$

Once the bio-weapon is placed at this location, it starts spreading until its spread area reaches the threshold distance d_0 from the detector. In other words, it spreads

for the distance $s - d_0$. During this spread, the disease covers the circle of radius $s - d_0$ and area $\pi \cdot (s - d_0)^2$.

By using the known population density $\rho(x)$, we can conclude that the number of affected people $n(x)$ is equal to

$$n(x) = \pi \cdot (s - d_0)^2 \cdot \rho(x). \quad (2.1.12)$$

Substituting the expression (2.1.10) into this formula, we conclude that

$$n(x) = \pi \cdot \left(\frac{c_1}{\sqrt{\rho_d(x)}} - d_0 \right)^2 \cdot \rho(x). \quad (2.1.13)$$

Adversary's choice of the location. According to our assumption about the adversary's objective function, the adversary wants to maximize the number of affected people. Thus, the adversary will select a location x for which this number $n(x)$ (as described by the expression (2.1.13)) is the largest possible. The resulting damage n is thus equal to the largest of the values $n(x)$:

$$n = \max_x \left(\pi \cdot \left(\frac{c_1}{\sqrt{\rho_d(x)}} - d_0 \right)^2 \cdot \rho(x) \right). \quad (2.1.14)$$

Our objective. Our objective is to minimize this overall damage, i.e., to select the detector placement $\rho_d(x)$ so as to minimize this value n .

In other words, we want to *minimize* the worst-possible (maximal) damage. This *minimax* formulation is typical for *zero-sum games*, in which the interests of the two sides are exactly opposite; see, e.g., [7].

Thus, we arrive at the following problem:

Resulting formulation of the problem in precise terms. We are given the population density $\rho(x)$, the value d_0 , and the total number of detectors N . We want to find a function $\rho_d(x)$ that minimizes the expression (2.1.14) under the constraint $\int \rho_d(x) dx = N$.

Analysis of the resulting optimization problem. The damage is determined by the maximum n of the function $n(x)$. Let us assume that we have already selected the optimal detector density function, i.e., the function $\rho_d(x)$ that minimizes the desired objective function n .

Let us show that the damage function $n(x)$ corresponding to this selection is constant. We will prove this by contradiction. If the function $n(x)$ is not constant, this means that at some locations x , the values $n(x)$ are smaller than the maximum n . In this case, we can slightly increase the detector density at the locations where $n(x) = n$, at the expense of slightly decreasing the location density at locations where $n(x) < n$.

The value of the expected damage $n(x)$ monotonically decreases with the detector density $\rho_d(x)$. This mathematical observation is in perfect accordance with common

sense: the more detectors we place at some location, the earlier we will be able to detect bio-weapons and thus, the smaller will be the resulting damage.

Thus, the above re-arrangement of detectors will decrease the value of $n(x)$ at all locations where $n(x) = n$ —and slightly increase at all other locations. As a result, after this detector relocation, the overall maximum $n = \max_x n(x)$ will decrease. This possibility contradicts to our initial assumption that the value n is the smallest possible. Thus, the function $n(x)$ is indeed constant.

Let us denote this constant by n_0 . Then, from the formula (2.1.13) for $n(x)$, we conclude that

$$n_0 = \pi \cdot \left(\frac{c_1}{\sqrt{\rho_d(x)}} - d_0 \right)^2 \cdot \rho(x). \quad (2.1.15)$$

Thus, we conclude that

$$\left(\frac{c_1}{\sqrt{\rho_d(x)}} - d_0 \right)^2 = \frac{n_0}{\pi \cdot \rho(x)}, \quad (2.1.16)$$

$$\frac{c_1}{\sqrt{\rho_d(x)}} - d_0 = \frac{c_2}{\sqrt{\rho(x)}}, \quad (2.1.17)$$

where we denote

$$c_2 \stackrel{\text{def}}{=} \frac{\sqrt{n_0}}{\sqrt{\pi}}. \quad (2.1.18)$$

Thus, we get

$$\frac{c_1}{\sqrt{\rho_d(x)}} = d_0 + \frac{c_2}{\sqrt{\rho(x)}}, \quad (2.1.19)$$

$$\sqrt{\rho_d(x)} = \frac{c_1}{d_0 + \frac{c_2}{\sqrt{\rho(x)}}}, \quad (2.1.20)$$

and

$$\rho_d(x) = \frac{c_1^2}{\left(d_0 + \frac{c_2}{\sqrt{\rho(x)}} \right)^2}, \quad (2.1.21)$$

From Eq. (2.1.11), we conclude that

$$c_1^2 = \frac{2 \cdot \sqrt{3}}{9}, \quad (2.1.22)$$

hence

$$\rho_d(x) = \frac{2 \cdot \sqrt{3}}{9} \cdot \frac{1}{\left(d_0 + \frac{c_2}{\sqrt{\rho(x)}}\right)^2}. \quad (2.1.23)$$

The value c_2 must be determined from the equation (2.1.1).

Thus, we arrive at the following solution:

Solution: The optimal detector location is characterized by the detector density

$$\rho_d(x) = \frac{2 \cdot \sqrt{3}}{9} \cdot \frac{1}{\left(d_0 + \frac{c_2}{\sqrt{\rho(x)}}\right)^2},$$

where the parameter c_2 must be determined from the equation

$$\int \frac{2 \cdot \sqrt{3}}{9} \cdot \frac{1}{\left(d_0 + \frac{c_2}{\sqrt{\rho(x)}}\right)^2} dx = N. \quad (2.1.24)$$

Case of $d_0 = 0$. As we have mentioned earlier, in some cases, we have $d_0 = 0$. In this case, the formula (2.1.23) takes a simplified form

$$\rho_d(x) = C \cdot \rho(x) \quad (2.1.25)$$

for some constant C . In this case, the detector density is exactly proportional to the population density.

Substituting the expression (2.1.25) into the constraint (2.1.1), we conclude that

$$N = C \cdot N_p, \quad (2.1.26)$$

where $N_p = \int \rho(x) dx$ is the total population. Thus, $C = \frac{N}{N_p}$ and the optimal detector placement Eq. (2.1.25) takes the form

$$\rho_d(x) = \frac{N}{N_p} \cdot \rho(x). \quad (2.1.27)$$

Towards more relevant objective functions. In our computations, we assumed that the main objective of the adversary is to maximize the number of people affected by the bio-weapon, i.e., to maximize the value $\int_A \rho(x) dx$, where A is the region where people become affected before the bio-weapon is detected.

As we have mentioned, the actual adversary's objective function may differ from this simplified objective function. For example, the adversary may take into account that different locations have different publicity potential. In this case, instead of

maximizing the total number of affected people, the adversary may want to maximize the weighted value $\int_A \tilde{\rho}(x) dx$, where $\tilde{\rho}(x) \stackrel{\text{def}}{=} w(x) \cdot \rho(x)$, and the weight $w(x)$ describes the publicity-related importance of the location x .

From the purely mathematical viewpoint, once we have fixed the weight functions $w(x)$, we get the exact same problem as before—with the only difference that we now have “effective population density” $\tilde{\rho}(x)$ instead of the original density $\rho(x)$. Thus, if we know the exact weight function $w(x)$, then we find the optimal detector density $\rho_d(x)$ by substituting the effective population density $\tilde{\rho}(x)$ instead of $\rho(x)$ into the above formulas.

2.2 Optimal Use of Stationary Sensors: Case Study of Optimal Placement of Environmental Sensors

In this section, we analyze the problem of the optimal use of stationary sensors, on another example: of optimal placement of environmental sensors. Specifically, we show that under reasonable assumption, the spatial variability of a field $f(x)$, i.e., the expected value

$$F(z) \stackrel{\text{def}}{=} E[(f(x+z) - f(x))^2],$$

has the form $F(z) = \left| \sum_{i=1}^n \sum_{j=1}^n g_{ij} \cdot z_i \cdot z_j \right|^\alpha$. We explain how to find g_{ij} and α from the observations, and how to optimally place sensors in view of this spatial variability.

The results of this section were first published in [8].

Need to describe spatial variability. To understand climate trends, we need to describe not only the values of temperature, humidity, wind speed and direction at a single location, we also need to know how these characteristics change from one location to the other. In other words, we need to describe spatial variability of the corresponding characteristics.

There is a similar need in other application areas. For example, to understand the brain activity within a region, in addition to describing brain activity at certain locations, we also need to describe how this brain activity changes from one location to the other, i.e., we need to describe spatial variability of the corresponding characteristics.

How to describe spatial variability: use of random variables. In general, we have a characteristic $f(x)$ that takes different values at different locations x . Since we cannot exactly predict the exact future values $f(x)$, it is reasonable to consider them random variables. Random variables $f(x)$ corresponding to different locations x form a *random field*.

How to describe spatial variability: use of normal distributions. The values $f(x)$ are determined by a large number of different factors. In statistics, the joint effect

of many small independent factors is—due to the Central Limit Theorem—well described by a normal distribution; see, e.g., [9]. Thus, it is reasonable to assume that the variables $f(x)$ are normally distributed.

A normal distribution is uniquely determined by its first two moments, i.e., by the expected values $E[f(x)]$ and $E[f(x) \cdot f(y)]$. The values $E[f(x)]$ and $E[(f(x))^2]$ describe the behavior at a single location. Thus, to describe spatial variability, it is sufficient to describe the values $E[f(x) \cdot f(y)]$ for $x \neq y$. Since we know the values $E[(f(x))^2]$ and $E[(f(y))^2]$, describing $E[f(x) \cdot f(y)]$ is equivalent to describing the following expected value:

$$C(x, y) \stackrel{\text{def}}{=} E[(f(y) - f(x))^2] = E[(f(y))^2] + E[(f(x))^2] - 2E[f(x) \cdot f(y)].$$

Homogeneity. Locally, the distribution is usually homogenous, i.e., does not change after a shift. Thus, if we change x to $x + z$ and y to $y + z$, we should get the same value $C(x, y)$: $C(x + z, y + z) = C(x, y)$. For $z = -z$, this leads to $C(x, y) = C(0, y - x)$. So, to describe spatial variability, it is sufficient to describe the function

$$F(z) \stackrel{\text{def}}{=} C(0, z) = E[(f(x + z) - f(x))^2].$$

Comment. For $z = 0$, the above definition leads to $F(0) = 0$.

Other natural requirements. It is reasonable to assume that $F(z)$ continuously depends on z .

It is also reasonable to assume that there is spatial variability, i.e., that $F(z) > 0$ for $z > 0$.

Another requirement is that $f(x)$ is very close to $f(y)$ only for close x and y . Formally, we will require that for some value $F_0 > 0$, the set $\{z : F(z) \leq F_0\}$ is bounded.

Comment. It should be mentioned that the spatial distribution is often *anisotropic*, i.e., depends on the direction. For example, a North-South oriented mountain range goes through the city of El Paso. The closeness to the mountain affects temperature, rainfall, wind, etc. As a result, the meteorological characteristics change much more when we move in the East-West direction than when we move in the North-South one.

We need to select a few-parametric family of functions $F(z)$. In different practical situations, we have different functions $F(z) \geq 0$. To describe all such situations, it is desirable to have a parametric family \mathcal{F} of possible functions $F(z)$.

Often, we only have a limited amount of data, so we can only statistically significantly determine a small number of parameters of the function $F(z)$. For example, in environmental sciences, we have a limited number of observations in remote areas such as most areas of Arctic and Antarctica. In brain research, we also often only have

limited data. To cover such situations, it is desirable to have simple, few-parametric families \mathcal{F} .

Desired properties of few-parametric families. The numerical value of a physical characteristic depends on the choice of a measuring unit. For example, for length, if we change from inches to cm, the numerical values increase by 2.54. In general, if we use a new unit which is λ times smaller than the previous one, then numerical values $f(x)$ increase by λ , and the resulting values of $F(z)$ increase by λ^2 . In principle, we can have an arbitrary positive value $C = \lambda^2$, so it is reasonable to require that the family \mathcal{F} contains, with every function $F(z)$, also all functions $C \cdot F(z)$ for every $C > 0$.

Another possible change is a change in spatial coordinates. In some applications, the usual coordinates work best, in other applications, polar, cylindrical, or other coordinates are more appropriate. Locally, each smooth coordinate transformation $x_i \rightarrow f_i(x_1, \dots, x_n)$ can be well approximated by a linear function $x_i \rightarrow \sum_{j=1}^n a_{ij} \cdot x_j + a_i$, i.e., in matrix terms, $x \rightarrow Ax + a$. Under this transformation, the difference $z = y - x$ is replaced with Az . It is therefore reasonable to require that the family \mathcal{F} contains, with every function $F(z)$, also all functions $F(Az)$ for all non-degenerate matrices A .

It turns out that these two requirements are sufficient to determine few-parametric families \mathcal{F} with the smallest possible number of parameters.

Proposition. Let \mathcal{F} be a $\frac{n \cdot (n+1)}{2}$ -parametric family of continuous functions $F(z)$ from \mathbb{R}^n to \mathbb{R} for which $F(z) = 0$, $F(z) > 0$ for $z \neq 0$, and for some $F_0 > 0$, the set $\{z : F(z) \leq F_0\}$ is bounded. Let us also assume that the family \mathcal{F} contains, with every function $F(z)$, also all functions $C \cdot F(z)$ for all $C > 0$ and all functions $F(Az)$ for all non-degenerate matrices A . Then, every function $F \in \mathcal{F}$ has the form

$$F(z) = \left| \sum_{i=1}^n \sum_{j=1}^n g_{ij} \cdot z_i \cdot z_j \right|^\alpha$$

for some real values α and g_{ij} .

Proof. In this proof, similarly to [10], we will use ellipsoids centered at 0, i.e., ellipsoids $E = \{z : \sum g_{ij} \cdot z_i \cdot z_j \leq 1\}$. We will call them *c-ellipsoids* (c for *centered*). To describe all such c-ellipsoids, we need to describe all symmetric matrices g_{ij} , so the family of c-ellipsoids is $\frac{n \cdot (n+1)}{2}$ -dimensional. The border $\{z : \sum g_{ij} \cdot z_i \cdot z_j = 1\}$ of an ellipsoid E will be denoted by ∂E .

1°. Let $F \in \mathcal{F}$. Let us first prove that there is a c-ellipsoid E_0 on whose border ∂E_0 we have $F(z) = F_0$ for all $z \in \partial E_0$.

1.1°. By definition of the class \mathcal{F} , the set $S \stackrel{\text{def}}{=} \{z : F(z) \leq F_0\}$ is bounded, and each function $F \in \mathcal{F}$ is continuous. Since $F(z)$ is continuous, the set S is closed.

Every bounded set can be enclosed into a c-ellipsoid. It is known (see, e.g., [11]) that, among all ellipsoids containing a given closed bounded set, there is exactly one ellipsoid with the smallest volume.

Let E_0 denote the c-ellipsoid with the smallest volume that contains the set S . We will say that this ellipsoid *corresponds* to the function $F(z)$.

Comment. The existence of the smallest-volume ellipsoid follows from the fact that every continuous function on a compact set attains its minimum. Uniqueness follows from the fact that if we have two c-ellipsoids E and E' of the same volume containing the same set, then we can select coordinates in which both matrices are diagonal, i.e., have the form $\sum g_i \cdot z_i^2 \leq 1$ and $\sum g'_i \cdot z_i^2 \leq 1$; then, for $g''_i = \frac{g_i + g'_i}{2}$, the ellipsoid $\sum g''_i \cdot z_i^2 \leq 1$ also contains the bounded set and, as can be easily shown, has a strictly smaller volume than E and E' .

1.2°. It is known that every c-ellipsoid E in appropriate affine coordinates becomes a unit ball $\{z : \sum z_i^2 \leq 1\}$. In other words, every ellipsoid can be obtained from a unit ball by an appropriate affine transformation. By combining the affine transformations corresponding to E and to E_0 , we conclude that E can be obtained from the ellipsoid E_0 by an affine transformation $z \rightarrow Az$.

Under an affine transformation, the ratio of volumes is preserved. So, since E_0 is the c-ellipsoid with the smallest volume that contains the set $S = \{z : F(z) \leq F_0\}$, E is the c-ellipsoid with the smallest volume containing the set $S' = \{z : F'(z) \leq F_0\}$, where $F'(z) \stackrel{\text{def}}{=} F(Az) \in \mathcal{F}$.

Different ellipsoids correspond to different functions $F'(z)$, so we have as many such functions $F'(z)$ as there are ellipsoids—i.e., a $\frac{n \cdot (n+1)}{2}$ -dimensional family.

1.3°. There are many affine transformations (rotations) that preserve the ball; in particular, for every two points on a unit sphere, there is a rotation that transforms one into another.

Thus, there are many affine transformations that preserve every ellipsoid E . In particular, for every two points $z, z' \in \partial E$ on this ellipsoid's border, there is an affine transformation that preserves ∂E and transforms z into z' .

For the ellipsoid E_0 , let us denote, by G_0 , the group of all affine transformations that preserve ∂E_0 .

1.4°. Let us show that the border ∂E_0 of the ellipsoid E_0 contains some points from the set $S \stackrel{\text{def}}{=} \{z : F(z) \leq z_0\}$.

We will prove this by contradiction. Let us assume that the border ∂E_0 of the ellipsoid E_0 does not contain any points from the set S . Then, we can proportionally shrink E_0 and get a new c-ellipsoid with the smaller volume that still contains S . This contradicts to the fact that E_0 has the smallest volume. The statement is proven.

1.5°. Let us prove that for all $z \in \partial E_0 \cap S$, we have $F(z) = F_0$.

Indeed, since $z \in S$, by definition of the set S , we have $F(z) \leq F_0$. On the other hand, since z belongs to the border ∂E_0 , the point z is a limit of points z_n from outside E_0 : $z_n \rightarrow z$. Outside E_0 , there are no points from S , so for all $z_n \notin E_0$, we

have $F(z_n) > F_0$. Since the function $F(z)$ is continuous, in the limit $z_n \rightarrow z$, we get $F(z) \geq F_0$. From $F(z) \leq F_0$ and $F(z) \geq F_0$, we conclude that $F(z) = F_0$.

1.6°. Finally, let us prove that every point $z \in \partial E_0$ belongs to the set S ; due to Part 1.5 of this proof, this will imply that $F(z) = F_0$ for all $z \in \partial E_0$.

We will prove this statement by contradiction. Let us assume that not every point $z \in \partial E_0$ belongs to the set S . Since transformations from G_0 transform every point $z \in \partial E_0$ into every other point $z' \in \partial E_0$, this means that not all transformations from G_0 preserve the intersection $\partial E_0 \cap S$. Thus, transformations that preserve the intersection form a subgroup $G'_0 \subset G_0$. Subgroups of the group of rotations are well known, they have smaller dimension than G_0 . Thus, we have a finite-parametric family of transformations (of dimension ≥ 1) that preserve ∂E_0 and turn the set $S = \{z : F(z) \leq F_0\}$ into a different set S' —i.e., which turn $F(z)$ into a different function $F'(z)$ for which the ellipsoid E_0 is the same. Thus, we have an at least 1-dimensional family of functions $F'(z)$ corresponding to E_0 .

By applying an affine transformation, we get a similar family of functions for every ellipsoid. The family of ellipsoids is already $\frac{n \cdot (n+1)}{2}$ -dimensional, and for each of them, there is an ≥ 1 -dimensional family of functions—thus, we get a $\geq \left(\frac{n \cdot (n+1)}{2} + 1\right)$ -dimensional family of functions $F'(z)$ —which contradicts to our assumption that the whole family \mathcal{F} is no more than $\frac{n \cdot (n+1)}{2}$ -dimensional. This contradiction shows that indeed $\partial E \subseteq S$.

2°. The ellipsoid E_0 corresponding to the function $F(z)$ has the form $\{z : \|z\|^2 \leq 1\}$, where $\|z\|^2 \stackrel{\text{def}}{=} \sum_{i,j} g_{ij} \cdot z_i \cdot z_j$. Let us prove that the function $F(z)$ has the form $F(z) = h(\|z\|)$ for some function $h(t)$ from real numbers to real numbers.

In other words, we need to prove that for every value v , the function $F(z)$ has a constant value on the border $\partial E_v \stackrel{\text{def}}{=} \{z : \|z\|^2 = v\}$ of the ellipsoid $E_v \stackrel{\text{def}}{=} \{z : \|z\|^2 \leq v\}$ which is obtained from E_0 by an appropriate dilation (homothety).

Indeed, if the function $F(z)$ had two different values on different points $z, z' \in \partial E_v$, then, similarly to Part 1.6 of this proof, we would be able to apply appropriate affine transformations and get a ≥ 1 -parametric family of functions $F'(z)$ corresponding to the same ellipsoid E_0 and thus, a $\geq \left(\frac{n \cdot (n+1)}{2} + 1\right)$ -dimensional family of functions $F'(z)$ —which contradicts to our assumption that $\dim(\mathcal{F}) \leq \frac{n \cdot (n+1)}{2}$.

3°. To complete the proof, let us show that $h(t) = \text{const} \cdot t^\alpha$.

Let us consider the functions $F(z)$ corresponding to all c -ellipsoids E which have the same volume $V(E)$ as E_0 : $V(E) = V(E_0)$. The dimension of the family of all such ellipsoids is $\frac{n \cdot (n+1)}{2} - 1$.

For every function $F(z) = h(\|z\|) \in \mathcal{F}$, and for every two real numbers $C > 0$ and $k > 0$, the family \mathcal{F} contains the function $C \cdot F(k \cdot z) = C \cdot h(k \cdot \|z\|)$. The corresponding transformations form a 2-dimensional multiplicative group.

The resulting family of functions cannot be fully 2-dimensional, since then, by considering such a family for every ellipsoid E with $V(E) = V(E_0)$, we would have a family of dimension

$$\geq \left(\frac{n \cdot (n+1)}{2} - 1 \right) + 2 = \frac{n \cdot (n+1)}{2} + 1 > \frac{n \cdot (n+1)}{2}$$

inside the family \mathcal{F} . Thus, in the 2-dimensional transformation group, there is a ≥ 1 -dimensional subgroup that keeps the function $h(t)$ invariant.

All subgroups of the 2-dimensional transformation group are well known, so we have $C(k) \cdot h(k \cdot t) = h(t)$ for some $C(k)$, and hence, $h(k \cdot t) = C^{-1}(k) \cdot h(t)$. It is known (see, e.g., [12]), that every continuous function that satisfies this functional equation has the form $h(t) = A \cdot t^\alpha$ for some A and α . The statement is proven, and so is our main result.

Mathematical comment: relation to Riemannian geometry. In general, the values g_{ij} describing spatial variability differ from one location to another. Thus, to describe spatial variability, we need to describe the values $g_{ij}(x)$ corresponding to different locations x . Mathematically, this is equivalent to describing a Riemannian metric.

How to determine g_{ij} and α from the empirical data? Based on the recorded values $f(x, t)$ at different locations x at different times $t = 1, \dots, T$, we can estimate $C(z) = E[(f(x+z) - f(x))^2]$ as

$$C(z) = \frac{1}{T} \cdot \sum_{t=1}^T (f(x+z, t) - f(x, t))^2.$$

We can then use the following iterative procedure to find g_{ij} and α . Initially, we take $g_{ij}^{(0)} = \delta_{ij}$, i.e., $g_{ii}^{(0)} = 1$ and $g_{ij}^{(0)} = 0$ when $i \neq j$. At each iteration k , we start with the values $g_{ij}^{(k-1)}$, and do the following.

First, we estimate $\alpha^{(k)}$ from the condition $C(z) \approx \left| \sum g_{ij}^{(k-1)} \cdot z_i \cdot z_j \right|^\alpha$. We can find this α by taking the logarithms of both sides and applying the Least Squares Method to the resulting system of linear equations with unknown α :

$$\ln C(z) \approx \alpha \cdot \ln \left(\sum_{i=1}^n \sum_{j=1}^n g_{ij}^{(k-1)} \cdot z_i \cdot z_j \right).$$

Once $\alpha^{(k)}$ is computed, we estimate $g_{ij}^{(k)}$ by applying the Least Squares Method to the following system of linear equations with unknown g_{ij} :

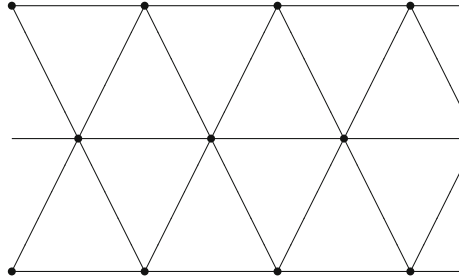
$$(C(z))^{1/\alpha^{(k)}} \approx \sum_{i=1}^n \sum_{j=1}^n g_{ij} \cdot z_i \cdot z_j.$$

Towards optimal sensor location. We want to place the sensors so as to reconstruct the value of $f(x)$ at all locations x with the desired accuracy ε . (Thus, in the spatial direction along which $f(x)$ changes faster we should place sensors more frequently.)

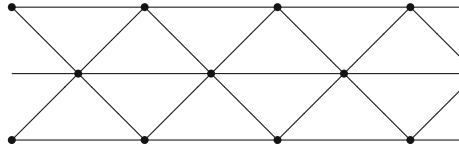
In precise terms, we want to place sensors in such a way that for each spatial location x , there is a sensor location s for which

$$E[(f(x) - f(s))^2] = \left| \sum_{i=1}^n \sum_{j=1}^n g_{ij} \cdot (x_i - s_i) \cdot (x_j - s_j) \right|^\alpha \leq \varepsilon^2.$$

For every symmetric matrix g_{ij} , there are affine coordinates—formed by its eigenvectors—in which this matrix become a unit matrix. In this case, the above condition simply means that every location must be ε -close to a sensor location. We have already mentioned, in Sect. 2.1, that under such condition, the asymptotically smallest number of sensors is provided by an equilateral triangle grid, i.e., a grid formed by equilateral triangles [6].



Hence, in general, the sensor grid can be obtained from the equilateral triangle one by an appropriate affine transformation.



In other words, we should place sensors along the grid parallel to eigenvectors of the matrix g_{ij} .

Mathematical comment: a similar problem of spatial distribution. Instead of spatial variation, we can consider a similar problem of spatial distributions, i.e., the problem of describing low-dimensional affine-invariant families of probability density functions—families that contain, with every function $\rho(x)$, the function $(\det(A))^{-1} \cdot \rho(Ax + a)$. Similar ellipsoid arguments—but with general ellipsoids instead of c-ellipsoids—show that in this case, every distribution from the corresponding family has the form $\rho(x) = h(\|x - a\|)$ for some function $h(t)$ and some vector a , where $\|z\|^2 = \sum_{i=1}^n \sum_{j=1}^n g_{ij} \cdot z_i \cdot z_j$ for some values g_{ij} .

2.3 Optimal Use of Mobile Sensors: Case Study of Unmanned Aerial Vehicles Patrolling the Border

In this section, we analyze the problem of the optimal use of mobile sensors, on the example of Unmanned Aerial Vehicles (UAVs) patrolling the border.

The results from this section were first published in [5]

Patrolling the border: a practical problem. Remote areas of international borders can be (and are) used by the adversaries: to smuggle drugs, to bring in weapons. It is therefore desirable to patrol the border, to minimize such actions.

Even with the current increase in the number of border patrol agents, it is not possible to effectively man every single segment of the border. It is therefore necessary to rely on other types of surveillance.

Unmanned Aerial Vehicles (UAVs) are an efficient way of patrolling the border:

- from every location along the border, they provide an overview of a large area, and
- if needed at a different location, they can move reasonably fast to the new location, without being slowed down by clogged roads or rough terrain.

However, while the area covered by the UAV is large, it is still limited. Due to resource limitations, we cannot have all the points on the border under a constant UAV surveillance. Thus, within a portion of the border that is covered by a UAV, it is necessary to keep the UAV moving.

How to describe UAV patrolling strategies. For simplicity, let us assume that the UAV can fly reasonably fast along the border, so that for each point, the interval between two consequent overflies does not exceed the time $2T$ needed to successfully cross the border area back-and-forth.

In the ideal case, this would mean that the UAV is capable of detecting all adversaries—and thus, preventing all border violations. In reality, however, a fast flying UAV can miss the adversary. It is therefore desirable to select a trajectory that would minimize the effect of this miss.

The faster the UAV goes pass a certain location, the less time it spends in the vicinity of this location, the more probable it is that the UAV will miss the adversary. From this viewpoint, an important characteristic of the trajectory is the velocity $v(x)$ with which the UAV passes through the location x . So, by a patrolling strategy, we will mean a function $v(x)$ that describes how fast the UAV flies at different locations.

This strategy must be selected in such a way that a total time for a UAV to go from one end of the area to another one is equal to the given value T . The time during which a UAV passes from the location x to the location $x + \Delta x$ is equal to

$$\Delta t = \frac{\Delta x}{v(x)}. \quad (2.3.1)$$

Thus, the overall flight time is equal to the sum of these times, i.e., to

$$T = \int \frac{dx}{v(x)}, \quad (2.3.2)$$

where the integral is taken over the whole length of the border segment.

From the mathematical viewpoint, an arbitrary non-negative function $v(x)$ can describe the velocity at different locations. In practice, not every function $v(x)$ can be implemented, since the UAV has the largest possible velocity V , so we must have $v(x) \leq V$ for all x .

From the computational viewpoint, it is convenient, instead of the velocity $v(x)$, to use its reciprocal

$$s(x) \stackrel{\text{def}}{=} \frac{1}{v(x)}. \quad (2.3.3)$$

In the geosciences, this reciprocal is called *slowness*; see, e.g., [13] and references therein; we will use this term in this section as well.

In terms of slowness, the requirement that the overall time be equal to T has a simpler form

$$T = \int s(x) dx. \quad (2.3.4)$$

In terms of slowness $s(x)$, the velocity limitation

$$v(x) \leq V \quad (2.3.5)$$

takes the form $s(x) \geq S$, where $S \stackrel{\text{def}}{=} \frac{1}{V}$. Since $s(x) \geq S$, the value $s(x)$ can be represented as $S + \Delta s(x)$, where $\Delta s(x) \stackrel{\text{def}}{=} s(x) - S$ satisfy the simpler constraint $\Delta s(x) \geq 0$.

In terms of $\Delta s(x)$, the requirement that the overall time be equal to T has a simpler form

$$T = S \cdot L + \int \Delta s(x) dx, \quad (2.3.6)$$

where L is the total length of the piece of the border that we are defending, or, equivalently,

$$T_0 = \int \Delta s(x) dx, \quad (2.3.7)$$

where $T_0 \stackrel{\text{def}}{=} T - S \cdot L$.

Probability of detection. In order to select a reasonable patrolling strategy, we must find out, for each strategy, what is the probability that under this strategy, the adversary can still cross the border.

Let h denote a distance at which the UAV can still see. This means that when the adversary is trying to cross at location x , a UAV can, in principle, observe this adversary when it is located in the zone between $x - h$ and $x + h$. The width of this

zone is equal to

$$(x + h) - (x - h) = 2h. \quad (2.3.8)$$

We have denoted the UAV's velocity at location x by $v(x)$. So, the time that it takes for a UAV to cross the zone of width $2h$ is equal to

$$t_{\text{obs}} = \frac{2h}{v(x)}. \quad (2.3.9)$$

In terms of slowness, this expression takes a simpler form

$$t_{\text{obs}} = 2h \cdot s(x). \quad (2.3.10)$$

Let Δt denote the time during which a UAV takes one snapshot of the underlying area. In these terms, during the crossing time t_{obs} , the UAV can take

$$n(x) = \frac{t_{\text{obs}}}{\Delta t} = \frac{2h}{\Delta t} \cdot s(x) \quad (2.3.11)$$

snapshots.

Let p_1 be the probability that an adversary can avoid detection based on a single snapshot. Then, to avoid detection during several snapshots means to avoid detection during the first snapshot, during the second snapshot, etc. It is reasonable to assume that the misses corresponding to different snapshots are statistically independent. Under this assumption, the probability $p(x)$ to be missed under $n(x)$ snapshots is equal to the product of $n(x)$ probabilities of a miss corresponding to different snapshots, i.e., equal to

$$p(x) = p_1^{n(x)}. \quad (2.3.12)$$

Substituting the above expression for $n(x)$ in terms of $s(x)$, we conclude that

$$p(x) = p_1^{(2h/\Delta t) \cdot s(x)}, \quad (2.3.13)$$

i.e., that

$$p(x) = \exp(-k \cdot s(x)), \quad (2.3.14)$$

where we denoted

$$k \stackrel{\text{def}}{=} \frac{2h}{\Delta t} \cdot |\ln(p_1)|. \quad (2.3.15)$$

Relative importance of different locations. We also need to take into account that different locations along the border have different importance.

For example, if smugglers succeed in bringing drugs to the vicinity of the city of El Paso, they can store in a safe place and distribute it without exposure. On the other hand, if they bring the same shipment in the remote desert area, they still need to

bring it close to a town or a city, and risk being detected while they are transporting this shipment.

In the case of smugglers, this importance can be described in monetary terms: a shipment available in city can be sold for a much larger amount than a shipment available at some remote location from which it still has to be transported to a city. The corresponding price $w(x)$ of the shipment successfully transported across the border at a point with coordinate x can be used as a measure of potential benefit, for the adversary, of penetrating the border at this particular location.

For other types of border penetration, we can also similarly estimate the potential benefit to the adversary.

We will start our analysis with a simplified case when we know the exact value of $w(x)$ for all x . After that, we will explain how to deal with a more realistic case, when we only know $w(x)$ with uncertainty.

Decision making: reminder. We assume that the adversary has observed the UAV, so the adversary knows the slowness function $s(x)$ and is, thus, capable of computing the probability $p(x)$ of avoiding detection. How does an adversary make decisions based on this knowledge?

A standard way to describe preferences of a decision maker is to use the notion of *utility*; see, e.g., [14–18]. To describe the utility of an outcome A , we need to select two extreme outcomes: a very unfavorable alternative A_- and a very favorable outcome A_+ .

We assume that all outcomes A in which we are interested are better than A_- and worse than A_+ . If we denote the relation “the decision maker prefers A' to A ” by $A \leq A'$, then we can describe this assumption as $A_- \leq A \leq A_+$.

Then, for each probability $p \in [0, 1]$, we can consider a *lottery* $L(p)$ in which we have A_+ with probability p and A_- with the remaining probability $1 - p$.

For $p = 1$, the lottery $L(p)$ coincides with A_+ , so we have $A \leq A(1)$. For $p = 0$, the lottery $L(p)$ coincides with A_- , so we have $A(0) \leq A$. The larger p , i.e., the larger the probability of a beneficial event A_+ , the more beneficial is the lottery $L(p)$ for the decision maker. So, if $p < q$, then $L(p) < L(q)$.

Let p_0 be the infimum (greatest lower bound) of the set of all the values p for which $A \leq L(p)$. Then:

- When $p < p_0$, then for $\tilde{p} = (p + p_0)/2$, we have $\tilde{p} < p_0$ and thus, by definition of the infimum, we cannot have $A \leq L(\tilde{p})$. Thus, we have $L(\tilde{p}) \leq A$. Since $p < \tilde{p}$, we have $L(p) < L(\tilde{p}) \leq A$ and thus, $L(p) < A$.
- When $p > p_0$, then, since p_0 is the greatest lower bound, p is not a lower bound, i.e., there exists a value \tilde{p} for which $A \leq L(\tilde{p})$ and $\tilde{p} < p$. Since $\tilde{p} < p$, we have $L(\tilde{p}) < L(p)$ hence $A < L(p)$.

Thus, we have the value p_0 that has the following property:

- when $p < p_0$, the corresponding lottery is worse than the event A :

$$L(p) < A; \quad (2.3.16)$$

- when $p > p_0$, the corresponding lottery is better than the event A :

$$L(p) > A. \quad (2.3.17)$$

This threshold value p_0 is called the *utility* of the event A . The utility is usually denoted by $u(A)$.

We can simplify the above somewhat complicated relation between A and p_0 by saying that the event $L(p_0)$ is *equivalent* to A . We will denote this equivalence by $A \sim L(p_0)$.

The notion of utility depends on the choice of the outcomes A_- (for which utility is 0) and A_+ (for which utility is 1). In principle, we select different outcomes A'_- and A'_+ . One can show that the new value $u'(A)$ is linearly related to the old one: $u'(A) = a \cdot u(A) + b$, where:

- $b = u'(A_-)$ is the utility of A_- in the new scale, and
- $a + b = u'(A_+)$ is the utility of A_+ in the new scale, so we can determine a as $u'(A_+) - u'(A_-)$.

In other words, utility is defined modulo an arbitrary linear transformation

$$u(A) \rightarrow u'(A) = a \cdot u(A) + b. \quad (2.3.18)$$

In practice, we can rarely predict the exact consequences of each decision. The consequences depend on the circumstances. For example, if we decide whether to take an umbrella or not, the consequences of this decision depend on whether it will rain or not. In the ideal situation, we know the probabilities p_1, \dots, p_n of different possible consequences E_1, \dots, E_n . In other words, the action leads to E_1 with probability p_1 , to E_2 with probability p_2 , ..., and to E_n with probability p_n .

By definition of the utility, the event E_1 is equivalent to a lottery $L(u(E_1))$ in which we get A_+ with probability $u(E_1)$, the event E_2 is equivalent to a lottery $L(u(E_2))$ in which we get A_+ with probability $u(E_2)$, etc. Thus, the original action is equivalent to the composite lottery, in which:

- with probability p_1 , we get a lottery that results in A_+ with probability $u(E_1)$, and in A_- otherwise;
- with probability p_2 , we get a lottery that results in A_+ with probability $u(E_2)$, and in A_- otherwise;
- ...

In this composite lottery, we get either A_+ or A_- , and the probability of getting A_+ can be easily computed as

$$u \stackrel{\text{def}}{=} p_1 \cdot u(E_1) + p_2 \cdot u(E_2) + \dots + p_n \cdot u(E_n). \quad (2.3.19)$$

Thus, the original action is equivalent to the lottery $L(u)$. By definition of the utility, this means that the utility of the action is equal to u .

From the mathematical viewpoint, u is the expected value of the utility of different consequences, so we can conclude that the utility of an action is the expected value of utilities of its consequences.

Strategy selected by the adversary. We have already mentioned that utility is defined modulo an arbitrary linear transformation. For convenience, let us select the utility scale in such a way that for the adversary, the utility of not being able to cross the border is 0.

In this scale, let $w(x)$ denote the utility of the adversary succeeding in crossing the border at location x . We have assumed that we know the exact value of $w(x)$ for every location x .

According to decision theory, the adversary will select a location x at which the expected utility

$$u(x) = p(x) \cdot w(x) = \exp(-k \cdot s(x)) \cdot w(x) \quad (2.3.20)$$

is the largest possible.

Thus, for each slowness function $s(x)$, the adversary's gain $G(s)$ is equal to

$$G(s) = \max_x u(x) = \max_x [\exp(-k \cdot s(x)) \cdot w(x)]. \quad (2.3.21)$$

Towards an optimal strategy for patrolling the border. Our objective is to select a strategy $s(x)$ for which the gain $G(s)$ is the smallest possible.

Let x_m be the location at which the utility $u(x) = \exp(-k \cdot s(x)) \cdot w(x)$ attains its largest possible value. If close to x_m , we have a point x_0 for which $u(x_0) < u(x_m)$ and $s(x_0) > S$, then we can slightly decrease the slowness $s(x_0)$ at the vicinity of x_0 (i.e., go faster in this vicinity) and use the resulting time to slow down (i.e., to go slower) at all locations x at which $u(x) = u(x_m)$. As a result, we slightly decrease the value $u(x_m) = \exp(-k \cdot s(x_m)) \cdot w(x_m)$.

Yes, we also slightly increase the value

$$u(x_0) = \exp(-k \cdot s(x_0)) \cdot w(x_0), \quad (2.3.22)$$

but for small changes, this value is still smaller than $u(x_m)$ and thus, does not affect the maximum $\max_x u(x)$. As a result, the gain $G(s)$ decreases (this argument is similar to the one presented in [19]).

So, when the adversary's gain is minimized, we get

$$u(x) = u_0 = \text{const} \quad (2.3.23)$$

hence

$$\exp(-k \cdot s(x)) = \frac{u_0}{w(x)}, \quad (2.3.24)$$

thence

$$s(x) = \frac{1}{k} \cdot (\ln(w(x)) - \ln(u_0)) \quad (2.3.25)$$

and

$$\Delta s(x) = \frac{1}{k} \cdot \ln(w(x)) - \Delta_0, \quad (2.3.26)$$

where

$$\Delta_0 \stackrel{\text{def}}{=} -\frac{1}{k} \cdot \ln(u_0) - S. \quad (2.3.27)$$

When this value gets to $s(x) = S$ and $\Delta s(x) = 0$, we get $\Delta s(x) = S$. Thus, we conclude that

$$\Delta s(x) = \max \left(\frac{1}{k} \cdot \ln(w(x)) - \Delta_0, 0 \right). \quad (2.3.28)$$

The value Δ_0 can be determined from the condition that

$$\begin{aligned} \int \Delta s(x) dx = \\ \int \max \left(\frac{1}{k} \cdot \ln(w(x)) - \Delta_0, 0 \right) dx = T_0. \end{aligned} \quad (2.3.29)$$

Since this integral monotonically decreases with Δ_0 , we can use bisection to find the appropriate value Δ_0 ; see, e.g., [20].

Towards taking fuzzy uncertainty into account . The above algorithm is based on the assumption that we know the exact value of the adversary's gain $w(x)$ at different locations. In reality, as we have mentioned, we only have expert estimates for $w(x)$. To formalize these estimates, we can use fuzzy techniques; see, e.g., [21, 22].

Once we have the fuzzy values $w(x)$, we can apply Zadeh's extension principle to the above crisp formulas and thus, come up with fuzzy recommendations about the slowness, such as "go somewhat slow here", "go fast", etc. It is well known (see, e.g., [21, 22]) that Zadeh's extension principle is equivalent to processing α -cuts. Specifically, if we know a relation $y = f(x_1, \dots, x_n)$ between the inputs x_1, \dots, x_n and the desired value y , and we know the fuzzy values X_1, \dots, X_n of the inputs, then the resulting fuzzy value Y of the output can be obtained as follows: for every $\alpha \in (0, 1]$, we have

$$\begin{aligned} Y(\alpha) &= f(X_1(\alpha), \dots, X_n(\alpha)) = \\ &\{f(x_1, \dots, x_n) : x_1 \in X_1(\alpha), \dots, x_n \in X_n(\alpha)\}, \end{aligned} \quad (2.3.30)$$

where for each fuzzy value Z with a membership function $\mu_Z(z)$, its α -cut $Z(\alpha)$ is defined as

$$Z(\alpha) \stackrel{\text{def}}{=} \{z : \mu_Z(z) \geq \alpha\}. \quad (2.3.31)$$

When a fuzzy value is a fuzzy number, each α -cut is an interval $Z(\alpha) = [\underline{Z}(\alpha), \overline{Z}(\alpha)]$. When all the inputs are fuzzy numbers, the above formula takes the simplified form

$$[\underline{Y}(\alpha), \overline{Y}(\alpha)] = \{f(x_1, \dots, x_n) : x_i \in [\underline{X}_i(\alpha), \overline{X}_i(\alpha)]\}. \quad (2.3.32)$$

When the function $y = f(x_1, \dots, x_n)$ is an increasing function of all its variables, then its largest value is attained when all its inputs attain their largest values, and its smallest value is attained when all its inputs attain their smallest values. In other words, the desired α -cut has the form $[\underline{Y}(\alpha), \overline{Y}(\alpha)]$, where

$$\underline{Y}(\alpha) = f(\underline{X}_1(\alpha), \dots, \underline{X}_n(\alpha)); \quad (2.3.33)$$

$$\overline{Y}(\alpha) = f(\overline{X}_1(\alpha), \dots, \overline{X}_n(\alpha)). \quad (2.3.34)$$

Similarly, when the function $y = f(x_1, \dots, x_n)$ is an increasing function of the variables x_1, \dots, x_k and decreasing in x_{k+1}, \dots, x_n , then the α -cut has the form $[\underline{Y}(\alpha), \overline{Y}(\alpha)]$, where

$$\underline{Y}(\alpha) = f(\underline{X}_1(\alpha), \dots, \underline{X}_k(\alpha), \overline{X}_{k+1}(\alpha), \dots, \overline{X}_n(\alpha)); \quad (2.3.35)$$

$$\overline{Y}(\alpha) = f(\overline{X}_1(\alpha), \dots, \overline{X}_k(\alpha), \underline{X}_{k+1}(\alpha), \dots, \underline{X}_n(\alpha)). \quad (2.3.36)$$

In our case, for each location x , we know the fuzzy value $W(x)$ of the corresponding gain. This means that for each degree α , we know the corresponding α -cut $W(x)(\alpha) = [\underline{W}(x)(\alpha), \overline{W}(x)(\alpha)]$.

In the crisp case, based on the gains $w(x)$, we first compute the value Δ_0 and then the corresponding changes $\Delta s(x)$ in the UAV's slowness. Thus, in the fuzzy case, we need to find the α -cuts for Δ_0 and then, α -cuts for $\Delta s(x)$.

According to the above formula for Δ_0 , its value is an increasing function of all the inputs $w(x)$. Thus, we conclude that for every α , the α -cut for Δ_0 has the form $[\underline{\Delta}_0(\alpha), \overline{\Delta}_0(\alpha)]$, where $\underline{\Delta}_0(\alpha)$ can be determined from the condition that

$$\int \max \left(\frac{1}{k} \cdot \ln(\underline{W}(x)(\alpha)(x)) - \underline{\Delta}_0(\alpha), 0 \right) dx = T_0, \quad (2.3.37)$$

and $\overline{\Delta}_0(\alpha)$ can be determined from the condition that

$$\int \max \left(\frac{1}{k} \cdot \ln(\overline{W}(x)(\alpha)(x)) - \overline{\Delta}_0(\alpha), 0 \right) dx = T_0. \quad (2.3.38)$$

The value $\Delta s(x)$ is increasing in $w(x)$ and decreasing in Δ_0 . Thus,

- the smallest value $\underline{\Delta s}(x)(\alpha)$ is attained when $w(x)$ is the smallest and Δ_0 is the largest, and

- the largest value $\overline{\Delta s(x)}(\alpha)$ is attained when $w(x)$ is the largest and Δ_0 is the smallest:

$$\underline{\Delta s(x)}(\alpha) = \max \left(\frac{1}{k} \cdot \ln(\underline{W(x)}(\alpha)) - \underline{\Delta_0}(\alpha), 0 \right); \quad (2.3.39)$$

$$\overline{\Delta s(x)}(\alpha) = \max \left(\frac{1}{k} \cdot \ln(\overline{W(x)}(\alpha)) - \underline{\Delta_0}(\alpha), 0 \right). \quad (2.3.40)$$

The resulting recommendations can be used either as a guidance for a human controller, or—by using fuzzy control—in the design of the automatic controller.

Comment. Fuzzy techniques can be similarly used in other problems related to security, e.g., in finding optimal placement for bio-weapon detectors [5] as described in Sect. 2.1.

2.4 Efficient Algorithms for Optimizing Sensor Use: Case Study of Security Problems

In this section, we analyze the problem of designing efficient algorithms for optimizing sensor use, on the example of security problems.

The results from this section first appeared in [23].

Formulation of the problem. Security problems typically involve making strategic resource allocation decisions in order to prevent or mitigate attacks. Game theory has been used to model decision-making in a variety of security situations, including the protection of critical infrastructure from terrorist attacks [24, 25], computer network security [26–28], robot patrolling [29–31], and scheduling [32]. Recently, research on security games has been deployed to make real-world homeland security decision, including the ARMOR system in use at the LAX airport [33], the IRIS system used by the Federal Air Marshals Service [34], and the GUARDS system developed for the Transportation Security Administration [35].

A key research direction has been the development of faster algorithms to scale to increasingly large and complex instances of security games [19, 36–38]. Faster algorithms that exploit the structure of security games have been key in enabling new applications of these methods. We present new algorithms for one of the most basic classes of security games: Stackelberg security games with multiple, identical defender resources. This class of games was described by Kiekintveld et al. [19], which also gave a polynomial-time ($O(n^2)$) algorithm for computing Stackelberg equilibrium of these games.

In this section, we present two new algorithms for Stackelberg security games with identical resources. The first solves a special case in worst-case linear time ($O(n)$), and the second solves the general case in $O(n \cdot \log(n))$. In addition to improving on

the theoretical complexity of the best known methods for this class of security games, our algorithms are based on a detailed analysis of the structure of the solutions for these games, which may lead to faster algorithms or heuristics for more complex variants of security games.

Security game model: general case. In this section, we adopt the general model of security games described in [19]. A security game has two players, a *defender*, Θ , and an *attacker*, Ψ . There is a set of n targets $t_i \in T$ that the attacker wishes to attack and the defender wishes to protect. In our model, the attacker can choose to attack exactly one target from this set. The defender has a limited number of resources, $m < n$, that can be deployed to protect the targets. We assume throughout that these resources are identical, and that at most one resource can be used to protect each target.

If the attacker chooses to attack target t_i , we call the attack *successful* if the target is left uncovered by a defender, and *unsuccessful* if the target is covered by a defender. The defender's payoff for an uncovered attack is denoted $U_\Theta^u(t)$, and for a covered attack $U_\Theta^c(t)$. Similarly, $U_\Psi^u(t)$ and $U_\Psi^c(t)$ denote the attacker's payoffs in each case. We will make the standard assumptions for security games that $U_\Theta^u(t) < U_\Theta^c(t)$ and $U_\Psi^u(t) > U_\Psi^c(t)$ for all targets t . In other words, the attacker receives a higher payoff for attacking an undefended target than a defended one, and vice versa for the defender. Note that this does not imply that the games are zero-sum (or even strategically zero-sum).

The attacker's possible strategies consist of attacking each of the n targets. The defender's space of possible strategies consists of all possible ways to assign the m resources to the n targets. However, we can conveniently summarize the defender's strategy by defining the *coverage vector* which gives the probability that there is a defender resource assigned to each individual target. Denote these probabilities by c_i , so that $\sum_{i=1}^n c_i = m$. The attacker's expected utility for an attack on target t_i can then be written as $(1 - c_i) \cdot U_\Psi^u(t_i) + c_i \cdot U_\Psi^c(t_i)$, and similarly for the defender. Because of our assumptions, for each target t_i , the defender's expected payoff decreases when the probability c_i of defending this target increases. We also assume that all defender resources are identical and can be deployed to any target.

We model the game as a *Stackelberg game* [39] in which the attacker can observe the defender's strategy (c_1, \dots, c_n) before planning an attack (modeling the capability of attackers to use surveillance to learn security policies). The standard solution concept for these games is Strong Stackelberg Equilibrium (SSE) [40, 41]. In an SSE, the leader first selects a mixed strategy, and then the follower chooses an optimal pure strategy in response, breaking ties in favor of the leader. This behavior can be induced by the leader selecting a strategy arbitrarily close to the equilibrium that causes the the follower to strictly prefer the desired strategy [42], but in practice we compute the limit point where ties are broken in favor of the leader.

Case of fully protective resources: description and analysis of the problem. Let us first consider a practically important case of fully protective resources. When a single resource is deployed at a target, the target is fully protected. For now, we

restrict the attacker's payoff for attacking a covered resource to 0: $U_\psi^c(t) = 0$. We begin with a basic analysis that describes the structure of the solution.

According to the analysis in [19], in our game, the objective of the defender is equivalent to minimizing the expected utility of the attacker, as long as the solution has the largest possible set of targets that are optimal for the attacker to select. Using the Strong Stackelberg Equilibrium assumption, the attacker will select breaking ties in favor of the defender. Therefore, in most cases we will not need to take into account the defender's payoffs directly; the defender's payoff will be maximized implicitly by finding a set of coverage probabilities c_i so as to minimize the expected payoff of the attacker.

The attacker seeks to maximize the expected value of a successful attack:

$$\arg \max_i (1 - c_i) \cdot U_\psi^u(t_i), \quad (2.4.1)$$

while the defender chooses a coverage vector to minimize the attacker's expected payoff.

Let t_{i_o} denote the optimal target to attack, so we have for every target t_i :

$$(1 - c_{i_o}) \cdot U_\psi^u(t_{i_o}) \geq (1 - c_i) \cdot U_\psi^u(t_i). \quad (2.4.2)$$

Now, assume that for some i this inequality is strict and that $c_i > 0$. In this case we could decrease c_i and increase the probability c_j for all j such that

$$(1 - c_j) \cdot U_\psi^u(t_j) = (1 - c_{i_o}) \cdot U_\psi^u(t_{i_o}), \quad (2.4.3)$$

thus decreasing the expected payoff of the attacker.

Therefore, for the minimizing coverage vector, all targets can be divided into two groups:

- either the expected value for attacking the target is equal to the optimal value,
- or the expected value is less than the optimal value and the coverage probability assigned to the target is 0.

In other words, the optimal solution will have the property that the attacker's expected value for all targets with positive coverage probability is equal to some constant q :

$$(1 - c_i) \cdot U_\psi^u(t_i) = q. \quad (2.4.4)$$

For any target t_i with $c_i > 0$ we can thus calculate the necessary value of c_i as:

$$c_i = 1 - \frac{q}{U_\psi^u(t_i)}. \quad (2.4.5)$$

For all other targets $U_\psi^u(t_i) < q$, and therefore

$$1 - \frac{q}{U_\psi^u(t_i)} < 0. \quad (2.4.6)$$

Summarizing: once we know q , we can find all the probabilities c_i by using the formula

$$c_i = \max \left(1 - \frac{q}{U_\psi^u(t_i)}, 0 \right). \quad (2.4.7)$$

For each target t_i , this formula requires a constant number of computational steps. Therefore, after q is computed, we can therefore compute all the probabilities c_i in time $O(n)$.

So, to find the optimal covering vector, it is sufficient to find the constant q . This constant can be found from the condition that $\sum_{i=1}^n c_i = m$, i.e., that

$$\sum_{i=1}^n \max \left(1 - \frac{q}{U_\psi^u(t_i)}, 0 \right) = m. \quad (2.4.8)$$

The left-hand side of this equality decreases as q increases. So, if for some q , the resulting sum is smaller than m , this means that the optimal value q_o is smaller than q : $q_o < q$; similarly, if for some q , the resulting sum is larger than m , this means that the optimal value q_o is larger than q : $q_o > q$.

The structure of the optimal covering vector can be clarified if we sort the targets in order of descending attacker payoffs for successful attacks, so that:

$$U_\psi^u(t_{(1)}) \geq \dots \geq U_\psi^u(t_{(n-1)}) \geq U_\psi^u(t_{(n)}). \quad (2.4.9)$$

We can also add $U_\psi^u(t_{(0)}) \stackrel{\text{def}}{=} +\infty$ and $U_\psi^u(t_{(n+1)}) \stackrel{\text{def}}{=} 0$, then

$$U_\psi^u(t_{(0)}) \geq \dots \geq U_\psi^u(t_{(n)}) \geq U_\psi^u(t_{(n+1)}). \quad (2.4.10)$$

The values $U_\psi^u(t_{(i)})$ divide the real line into intervals $[U_\psi^u(t_{(i+1)}), U_\psi^u(t_{(i)})]$, so the threshold constant q must be in one of these intervals, i.e., between $U_\psi^u(t_{(k+1)})$ and $U_\psi^u(t_{(k)})$ for some k . In this case, according to the above formula for c_i , all targets with a value greater than q (i.e., the targets $t_{(1)}, t_{(2)}, \dots, t_{(k)}$ in the above ordering) will be protected with positive probability, and all targets with value smaller than q (i.e., targets $t_{(k+1)}, t_{(k+2)}, \dots$) are left unprotected. Let k denote the index of last target that has positive probability. Given the constraint that the coverage probabilities add to m , we can write:

$$\sum_{i=1}^k \left(1 - \frac{q}{U_\psi^u(t_{(i)})} \right) = m, \quad (2.4.11)$$

hence

$$k - m = q \cdot \sum_{i=1}^k \frac{1}{U_{\Psi}^u(t_{(i)})}, \quad (2.4.12)$$

and

$$q = \frac{k - m}{\sum_{i=1}^k \frac{1}{U_{\Psi}^u(t_{(i)})}}. \quad (2.4.13)$$

So, instead of selecting q , we can simply select a threshold value k .

Once we have found this k , we can then compute the threshold value q by using the above formula and then use this q to find the optimal coverage probabilities.

For the optimal value $k = k_o$, the corresponding value q is located in the interval $[U_{\Psi}^u(t_{(k+1)}), U_{\Psi}^u(t_{(k)})]$. If for some k , the value q computed by the above formula is smaller than $U_{\Psi}^u(t_{(k+1)})$, this means that we are trying to cover too few targets, with the same q , we can cover more, so the optimal value k_o should be larger: $k > k_o$.

Similarly, if for some k , the value q computed by the above formula is larger than $U_{\Psi}^u(t_{(k)})$, this means that we are trying to cover too many targets, so the optimal value k_o should be smaller: $k_o < k$.

Let us show that this argument can lead to a linear-time algorithm for finding the optimal coverage vector.

Case of fully protective resources: linear-time algorithm. On each stage of this iterative algorithm, we have three lists of targets:

- the list T^c of the targets t_i about which we are sure that in the optimal coverage, these targets will be covered with a positive probability $c_i > 0$;
- the list T^u of the targets t_i about which we are sure that in the optimal coverage, these targets will not be covered ($c_i = 0$);
- the list $T^?$ of the targets t_i about which we have not yet found out whether they will be covered or not in the optimal coverage.

In the beginning, we set $T^c = T^u = \emptyset$ and

$$T^? = \{t_1, t_2, \dots, t_n\}. \quad (2.4.14)$$

At each stage, we will also update the value

$$S^c = \sum_{t_i \in T^c} \frac{1}{U_{\Psi}^u(t_i)}. \quad (2.4.15)$$

In the beginning, since $T^c = \emptyset$, we take $S^c = 0$.

At each iteration, we do the following:

- First, we compute the median m of the values $U_{\Psi}^u(t_i)$ corresponding to all “undecided” targets $t_i \in T^?$.

- Then, by analyzing the elements of the undecided set $T^?$ one by one, we divide them into two subsets

$$T^+ = \{t_i : U_\psi^u(t_i) \geq m\}, T^- = \{t_i : U_\psi^u(t_i) < m\}. \quad (2.4.16)$$

In the set T^+ , we find the target t_{k^+} with the smallest value of $U_\psi^u(t_i)$; in the set T^- , we find the target t_{k^-} with the largest value of $U_\psi^u(t_i)$.

- We then compute

$$S^+ = \sum_{t_i \in T^+} \frac{1}{U_\psi^u(t_i)}, \quad (2.4.17)$$

$$s^+ = S^c + S^+, \text{ and } q = \frac{k - m}{s^+}.$$

- If $q < U_\psi^u(t_{k^-})$, then, as we have argued in our analysis, this means that we are trying to cover too few targets, so definitely all the elements from the set T^+ should be covered. Thus, we replace T^c with $T^c \cup T^+$, $T^?$ with T^- , and S^c with s^+ .
- If $q > U_\psi^u(t_{k^+})$, this means that we are trying to cover too many targets, so definitely all the elements from the set T^- should not be covered. Thus, we replace T^u with $T^u \cup T^-$ and $T^?$ with T^+ (and keep S^c unchanged).
- Finally, if $U_\psi^u(t_{k^-}) \leq q \leq U_\psi^u(t_{k^+})$, this means that this q is optimal.

Iterations continue until we find the optimal value q . Once we get the optimal value q , we can then find the optimal covering probabilities as $c_i = \max \left(1 - \frac{q}{U_\psi^u(t_i)}, 0 \right)$.

Let us prove that this algorithm indeed takes linear time. Indeed, at each iteration, we can compute the median in linear time [20], and all other operations with the set $T^?$ also take time \mathcal{T} linear in the number of elements $|T^?|$ of this set $T^?$: $\mathcal{T} \leq C \cdot |T^?|$ for some C . We start with the set $T^?$ of size n . On the next iteration, we have a set of size $n/2$, then $n/4$, etc. Thus, the overall computation time is

$$\leq C \cdot (n + n/2 + n/4 + \dots) \leq C \cdot 2n,$$

i.e., linear in n .

General case: analysis of the problem. Let us now go back to the general case, when defense resources are not necessarily fully protective. In this general case, the attacker seeks to maximize the expected value of a successful attack:

$$\arg \max_{\ell} e_{\ell}(c_{\ell}), \quad (2.4.18)$$

where

$$e_{\ell}(c_{\ell}) \stackrel{\text{def}}{=} (1 - c_{\ell}) \cdot U_{\psi}^u(t_{\ell}) + c_{\ell} \cdot U_{\psi}^c(t_{\ell}), \quad (2.4.19)$$

while the defender chooses a coverage vector to minimize the attacker's expected payoff

$$e(c) = \max_{\ell} e_{\ell}(c_{\ell}). \quad (2.4.20)$$

Once we select a coverage vector, we thus divide all the targets into three groups:

- the first group is formed by targets t_i for which $c_i = 1$; these targets that will be guarded with certainty;
- the second group is formed by targets t_j for which $0 < c_j < 1$; these targets with some probability will be guarded and with some probability will not be guarded;
- the third group is formed by targets t_k for which $c_k = 0$; these targets will be not guarded.

Intuitively, this division makes sense:

- the most important targets must be guarded no matter what,
- the least valuable targets will not be guarded at all if we do not have enough resources, and
- intermediate targets will be guarded with some probability.

Let us prove that this intuitive meaning is indeed true. To be more precise, let us prove that in this game, there exists a minimizing vector (c_1, \dots, c_n) that has the following properties:

- The expected payoff $e_i(c_i)$ of each target t_i of the first group (with $c_i = 1$) is larger than or equal to the expected payoff $e_j(c_j)$ of each target t_j of the second group (with $0 < c_j < 1$):

$$e_i(c_i) \geq e_j(c_j). \quad (2.4.21)$$

- The expected payoff $e_j(c_j)$ of all target $t_j, t_{j'}$ from the second group (with $0 < c_j < 1$) is the same:

$$e_j(c_j) \geq e_{j'}(c_{j'}). \quad (2.4.22)$$

- The expected payoff $e_j(c_j)$ of each target t_j from the second group (with $0 < c_j < 1$) is larger than or equal to the expected payoff of each target t_k from the third group (with $c_k = 0$):

$$e_j(c_j) \geq e_k(c_k). \quad (2.4.23)$$

Intuitively, this makes sense: if the attacker's expected payoff from a target t_i that we guard absolutely is smaller than the expected payoff from some other target t_j that we guard with a certain probability, then it makes sense to switch some probability from target t_i to target t_j . In this case, the attacker's expected value for t_j decreases; for t_i it somewhat increases, but since it was smaller than for the target t_j , it remains smaller, and the maximum of these values $e_i(c_i)$ does not increase.

To prove this result more formally, let us start with any minimizing vector and show that by appropriate transformations it can be transformed into a minimizing vector with the desired properties.

First, let us show how we can satisfy the first property. For that, let us show that we can decrease the number of targets t_i for which $c_i = 1$ and for which, for some j , we have $0 < c_j < 1$ and $e_i(c_i) < e_j(c_j)$. Indeed, out of all such targets, let us pick a target for which the value $e_i(c_i)$ is the smallest, and let j be the corresponding target from the second group. Then, for some $\Delta > 0$, we replace c_i with $c'_i = c_i - \Delta$ and c_j with $c'_j = c_j + \Delta$. When Δ is small enough, we have $c'_i > 0$, $c'_j < 1$, and $e_i(c'_i)$ is still smaller than all the values $e_\ell(c_\ell)$ for which we had $e_i(c_i) < e_\ell(c_\ell)$.

Let us keep all the other probabilities the same: $e'_\ell = c_\ell$ for all $\ell \neq i, j$. This replacement does not change the sum $\sum c_i$, so while $c'_i \geq 0$ and $c'_j \leq 1$, we still get a coverage vector. As we have mentioned, the expected value of a target decreases with the increase in the probability that this target will be guarded. Thus, when Δ increases, the value $e_i(c_i - \Delta)$ increases while the value $e_j(c_j + \Delta)$ decreases. So, while $e_i(c_i - \Delta) \leq e_j(c_j + \Delta)$, we have $e_i(c_i) < e_i(c_i - \Delta) \leq e_j(c_j + \Delta) < e_j(c_j)$. Thus, $e_i(c'_i) < e_j(c_j) \leq e(c) = \max_{\ell} c_\ell(e_\ell)$ and similarly $e_j(c'_j) < e_j(c_j) \leq e(c) = \max_{\ell} c_\ell(e_\ell)$. For all other targets ℓ , we have $c'_\ell = c_\ell$ hence $e_\ell(c'_i) = e_\ell(c_\ell) \leq e(c)$. Thus,

$$e(c') = \min \left(e_i(c'_i), e_j(c'_j), \min_{\ell \neq i, j} e_\ell(c'_\ell) \right) \leq e(c). \quad (2.4.24)$$

Since the original vector c is a minimizing vector, the value $e(c)$ is the smallest possible value, we conclude that c' is also a minimizing vector.

Let us show that in the new minimizing vector, the number of targets ℓ from the first group for which the expected value is smaller than for some target from the second group is smaller than the same number computed based on the original minimizing vector. Indeed, in the new minimizing vector, the target t_i is no longer from group one, it is now from group two, so it is sufficient to check that this addition of a new group-two target does not lead to the appearance of a new “wrong-order” target of group one. Indeed, if for some target $t_{i'}$ from group one, we have $e_{i'}(c_{i'}) < e_i(c'_i)$, then we could not have $e_i(c_i) < e_{i'}(c_{i'})$ —because we selected Δ so small that all such inequalities remain. Thus, we have $e_{i'}(c_{i'}) \leq e_i(c_i)$ but in this case $e_i(c_i) < e_j(c_j)$ implies that $e_{i'}(c_{i'}) < e_j(c_j)$ —and thus, $t_{i'}$ was the wrong-order target already in the original minimizing vector.

By applying this procedure again and again, we arrive at the new minimizing vector for which the number of wrong-order targets of group one is 0, i.e., in which the expected payoff for every target from group one is larger than or equal to the expected payoff for every target from group two.

Similarly, we can get a new minimizing vector in which the expected payoff for every target from group two is larger than or equal to the expected payoff of every target of group three.

Let us now show that we can arrive at the minimizing vector for which for all targets from group two, the expected payoff is the same. Let us show how an appropriate procedure can minimize the number of pairs $(t_j, t_{j'})$ of targets from group two for which $e_j(c_j) < e_{j'}(c_{j'})$. Indeed, let us sort all the corresponding values $e_j(c_j)$ into an increasing sequence, and let us take two neighboring values from this sequence.

Similarly to the above case, we replace c_j with $c'_j = c_j - \Delta$ and $c_{j'}$ with $c'_{j'} = c_{j'} + \Delta$. Both expected values $e_j(c_j - \Delta)$ and $e_{j'}(c_{j'} + \Delta)$ linearly depend on Δ , so, by solving the corresponding linear equation, we can find Δ for which $e_j(c_j - \Delta) = e_{j'}(c_{j'} + \Delta)$. If this value Δ satisfies the conditions $c'_j = c_j - \Delta \geq 0$ and $c'_{j'} = c_{j'} + \Delta \leq 1$, we get a new minimizing vector in which strict inequality holds for one fewer pair of targets from group two. If this value Δ does not satisfy one of these inequalities, this means that for some smaller value $\Delta' < \Delta$, we have either $c'_j = 0$ or $c'_{j'} = 1$. In both cases, the pairs stops being a wrong-order pair of targets from group two. One can check that no other wrong-order pairs appear after this transformation.

Let us now take the minimizing vector with the desired properties. In particular, this means that for all targets from group two, the attacker's expected value is the same. Let us denote this common value by q . Then, for every target t_j with $0 < c_j < 1$, we have

$$(1 - c_j) \cdot U_\Psi^u(t_j) + c_\ell \cdot U_\Psi^c(t_j) = q. \quad (2.4.25)$$

So, we can calculate c_j as

$$c_j = \frac{U_\Psi^u(t_j) - q}{U_\Psi^u(t_j) - U_\Psi^c(t_j)}. \quad (2.4.26)$$

For targets for which $U_\Psi^u(t_k) < q$, we have $c_k = 0$ —and the above ratio is negative. For targets for which $U_\Psi^c(t_i) > q$, we have $c_i = 1$ —and the above ratio is larger than 1. Thus, if the ratio is smaller than 0, we take $c_i = 0$, and if the ratio is larger than 1, we take $c_i = 1$.

So, once we know q , for all targets t_i , we can find all the covering probabilities c_i by using the following formula:

$$c_i = \min \left(\max \left(\frac{U_\Psi^u(t_i) - q}{U_\Psi^u(t_i) - U_\Psi^c(t_i)}, 0 \right), 1 \right). \quad (2.4.27)$$

For each target t_i , this formula requires a constant number of computational steps. Therefore, after q is computed, we can therefore compute all the probabilities c_i in time $O(n)$.

So, to find the optimal covering vector, it is sufficient to find the constant q . This constant can be found from the condition that $\sum_{i=1}^n c_i = m$, i.e., that

$$\sum_{i=1}^n \min \left(\max \left(\frac{U_\Psi^u(t_i) - q}{U_\Psi^u(t_i) - U_\Psi^c(t_i)}, 0 \right), 1 \right) = m. \quad (2.4.28)$$

The left-hand side of this equality decreases as q increases. So:

- If for some q , the resulting sum is smaller than m , this means that the optimal value q_o is smaller than q : $q_o < q$.
- Similarly, if for some q , the resulting sum is larger than m , this means that the optimal value q_o is larger than q : $q_o > q$.

Here, the target t_i is covered with probability $c_i > 0$ if and only if $q < U_\psi^u(t_i)$, and the target t_i is covered with probability $c_i = 1$ if and only if $U_\psi^c(t_i) \geq q$. Thus, the above formula for determining q can be rewritten as follows:

$$k(q) + \sum_{i: U_\psi^c(t_i) < q \leq U_\psi^u(t_i)} \frac{U_\psi^u(t_i) - q}{U_\psi^u(t_i) - U_\psi^c(t_i)} = m, \quad (2.4.29)$$

where

$$k(q) \stackrel{\text{def}}{=} \#\{i : U_\psi^c(t_i) \geq q\}. \quad (2.4.30)$$

Thus, if we know the place of q with respect to all the values $U_\psi^u(t_i)$ and $U_\psi^c(t_i)$, we can determine q by explicitly solving the above linear equation.

If we sort all $2n$ values $U_\psi^u(t_i)$ and $U_\psi^c(t_i)$ into a decreasing sequence

$$z_0 = +\infty \geq z_1 \geq z_2 \geq \dots \geq z_{2n-1} \geq z_{2n} \geq z_{2n+1} = 0, \quad (2.4.31)$$

we thus subdivide the real line into $2n + 1$ zones $[z_{k+1}, z_k]$, within each of which the relation between q and the values $U_\psi^u(t_i)$ and $U_\psi^c(t_i)$ is fixed. Thus, within each zone, we can find the corresponding q and check whether this value is indeed within the corresponding zone. As a result, in order to find q , it is sufficient to find the corresponding value k .

Since the order is decreasing, the larger k , the smaller q , and the more targets we cover. The selection of the zone means that we select which targets we cover fully, and which targets we cover with a positive probability. Similar to the case of fully protective resources:

- If based on this selection, we need more than m resources—i.e., if the value q obtained from solving the above linear equation is smaller than all the values from this zone—this means that we are trying to cover too many targets, so we need to decrease k .
- Similarly, if it turns out that based on this selection, we need fewer than m resources—i.e., that the value q obtained from solving the above linear equation is larger than all the values from this zone—this means that we are trying to cover too few targets, so we can increase k .

Thus, we can use bisection to find the appropriate zone, and we arrive at the following algorithm.

General case: $O(n \cdot \log(n))$ algorithm. First, we sort all $2n$ values $U_\psi^u(t_i)$ and $U_\psi^c(t_i)$ into a decreasing sequence:

$$z_1 \geq z_2 \geq \dots \geq z_{2n-1} \geq z_{2n}. \quad (2.4.32)$$

We then take $z_0 = +\infty$ and $z_{2n+1} = 0$, so that we get:

$$z_0 \geq z_1 \geq z_2 \geq \dots \geq z_{2n-1} \geq z_{2n} \geq z_{2n+1}. \quad (2.4.33)$$

Then, we use bisection to find the value k for which $z_k \geq q \geq z_{k+1}$. At each stage of this bisection procedure, we keep two values ℓ and u such that $z_\ell \geq q \geq z_u$. In the beginning, we have $\ell = 0$ and $u = 2n + 1$. At each iteration, we do the following:

- First, we compute the midpoint $m = (\ell + u)/2$.
- Then, under the assumption that $q \in [z_{m+1}, z_m]$, we compute

$$k_m = \#\{i : U_\psi^c(t_i) \geq z_{m+1}\}, \quad (2.4.34)$$

then $m_0 = m - k_m$, and find q from the resulting linear equation

$$\sum_{i: U_\psi^c(t_i) \leq z_m \leq z_{m+1} \leq U_\psi^u(t_i)} \frac{U_\psi^u(t_i) - q}{U_\psi^u(t_i) - U_\psi^c(t_i)} = m_0. \quad (2.4.35)$$

- If the resulting value q is smaller than z_m , then, according to our analysis, this means that the optimal k is larger than m , so we replace the original value ℓ with m .
- If the resulting value q is larger than z_{m+1} , then, according to our analysis, this means that the optimal k is smaller than m , so we replace the original value u with m .

The algorithm stops when $u = \ell + 1$, in which case we have the desired q . Based on this q , we can compute all coverage probabilities by using the above formula

$$c_i = \min \left(\max \left(\frac{U_\psi^u(t_i) - q}{U_\psi^u(t_i) - U_\psi^c(t_i)}, 0 \right), 1 \right). \quad (2.4.36)$$

There is one more special case the must be considered to ensure that this solution is in fact a Strong Stackelberg Equilibrium. This case occurs when at least one target has coverage $c_i = 1$. In this case, we must ensure that the target that gives maximum payoff for the defender has an optimal payoff for the attacker (so far, we have considered only the payoffs for the attacker). This can be done by first finding the maximal covered payoff for the attacker $U_\psi^c(t)$ for any target that has coverage probability 1. Denote this target by t_{max} . We then loop through each of the targets to determine whether the defender would achieve a higher payoff if the coverage probability was reduced so that the attackers expected payoff was equal to $U_\psi^c(t_{max})$.

We can compute the necessary coverage for each target using the equation:

$$c_i = \frac{U_{\psi}^c(t_{max}) - U_{\psi}^u(t_i)}{U_{\psi}^c(t_i) - U_{\psi}^u(t_i)} \quad (2.4.37)$$

If the defender's expected payoff for target t_i is greater than $U_{\psi}^c(t_{max})$ given c_i , then we reduce the coverage probability to this new value c_i for target t_i . Note that this can only reduce the total coverage probability required. The additional coverage can either be left unallocated or assigned arbitrarily to any target for which the attacker has an expected payoff less than $U_{\psi}^c(t_{max})$.

Let us prove that this algorithm indeed takes time

$$O(n \cdot \log(n)). \quad (2.4.38)$$

Indeed, sorting can be done in time $O(n \cdot \log(n))$ [20]. At each stage of the bisection method, we handle each target once, so each stage takes $O(n)$ computational steps. We start with an interval $[\ell, u]$ of size $2n$. At each stage, we replace it with a half-size interval $[\ell, m]$ or $[m, u]$. Thus, after the first iteration, we get an interval of size n , after the second, of size $n/2$, ..., and after k -th iteration, an interval of size $(2n)/2^k$. Thus, this procedure stops after $\log_2(2n)$ iterations. So, the overall computation time is indeed

$$O(n \cdot \log(n)) + O(n) \cdot \log(2n) = O(n \cdot \log(n)). \quad (2.4.39)$$

The final stage or analysis for the special case where at least one target coverage $c_i = 1$ requires two loop through each target. The first identifies the fully-covered target with maximum payoff for the attacker $U_{\psi}^c(t_{max})$. The second calculates the required reduction in coverage probability to make the attacker indifferent between t_{max} and any other target, and replaces the coverage probability if a reduction is beneficial for the defender. Since this requires time $O(2 \cdot n)$, the overall complexity remains $O(n \cdot \log(n))$.

Summary of the results. In this section, we have presented two new algorithms for a fundamental class of Stackelberg security games. These algorithms operate in linear time for a restricted case, and $O(n \cdot \log(n))$ for the general case, both improvements over the best known algorithms for this class of games. The algorithms are based on new analysis of the structure of the game-theoretic solutions of these games, which may provide insights to improve the efficiency of algorithms for additional classes of security games.

References

1. B. Baesens, *Analytics in a Big Data World: The Essential Guide to Data Science and its Applications* (Wiley, Hoboken, New Jersey, 2014)
2. B. Marr, *Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance* (Wiley, Chichester, UK, 2015)
3. N. Marz, J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems* (Manning Publication, Shelter Island, New York, 2015)
4. V. Mayer-Schönberger, K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think* (John Murray Publication, London, UK, 2013)
5. C. Kiekintveld, O. Lerma, Towards optimal placement of bio-weapon detectors, in *Proceedings of the 30th Annual Conference of the North American Fuzzy Information Processing Society NAFIPS'2011*, El Paso, Texas, 18–20 Mar 2011
6. R. Kershner, The number of circles covering a set. *Am. J. Math.* **61**(3), 665–671 (1939)
7. R.B. Myerson, *Game Theory: Analysis of Conflict* (Harvard University Press, Harvard, Massachusetts, 1997)
8. M. Koshelev, O. Lerma, C. Tweedie, Towards optimal few-parametric representation of spatial variation: geometric approach and environmental applications. *Geoinformatics* **21**(1), 15–24 (2011)
9. D.J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures* (Chapman & Hall/CRC, Boca Raton, Florida, 2011)
10. S. Li, Y. Ogura, V. Kreinovich, *Limit Theorems and Applications of Set Valued and Fuzzy Valued Random Variables* (Kluwer Academic Publishers, Dordrecht, 2002)
11. H. Busemann, *Geometry of Geodesics* (Dover, New York, 2005)
12. J. Aczel, *Lectures on Functional Equations and Their Applications* (Dover, New York, 2006)
13. M.G. Averill, K.C. Miller, G.R. Keller, V. Kreinovich, R. Araiza, S.A. Starks, Using expert knowledge in solving the seismic inverse problem. *Int. J. Approx. Reason.* **45**(3), 564–587 (2007)
14. P.C. Fishburn, *Utility Theory for Decision Making* (Wiley, New York, 1969)
15. P.C. Fishburn, *Nonlinear Preference and Utility Theory* (The John Hopkins Press, Baltimore, Maryland, 1988)
16. R.L. Keeney, H. Raiffa, *Decisions with Multiple Objectives* (Wiley, New York, 1976)
17. R.D. Luce, R. Raiffa, *Games and Decisions: Introduction and Critical Survey* (Dover, New York, 1989)
18. H. Raiffa, *Decision Analysis* (Addison-Wesley, Reading, Massachusetts, 1970)
19. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordonez, M. Tambe, Computing optimal randomized resource allocations for massive security games, in *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'09*, Budapest, Hungary, 10–15 May 2009
20. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms* (MIT Press, Cambridge, Massachusetts, 2009)
21. G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications* (Prentice Hall, Upper Saddle River, New Jersey, 1995)
22. H.T. Nguyen, E.A. Walker, *First Course on Fuzzy Logic* (CRC Press, Boca Raton, Florida, 2006)
23. O. Lerma, V. Kreinovich, C. Kiekintveld, Linear-time resource allocation in security games with identical fully protective resources, in *Proceedings of the AAAI Workshop on Applied Adversarial Reasoning and Risk Modeling AARM'11*, San Francisco, California, 7 Aug 2011
24. T. Sandler, D.G. Arce, Terrorism and game theory, in *Simulation and Gaming*, vol. 34, no. 3 (2003), pp. 319–337
25. V.M. Bier, Choosing what to protect. *Risk Analysis* **27**(3), 607–620 (2007)
26. T. Alpcan, T. Basar, A game theoretic approach to decision and analysis in network intrusion detection, in *Proceedings of the 42nd IEEE Conference on Decision and Control CDC'2003*, Maui, Hawaii, Maui, 9–12 Dec 2003, pp. 2595–2600

27. K.C. Nguyen, T.A.T. Basar, Security games with incomplete information, in *Proceedings of IEEE International Conference on Communications ICC'09*, Dresden, Germany, 14–18 June 2009
28. V. Srivastava, J. Neel, A.-B. MacKenzie, R. Menon, L.A. Dasilva, J.E. Hicks, J.H. Reed, R.P. Gilles, Using game theory to analyze wireless ad hoc networks. *IEEE Commun. Surv. Tutor.* **7**(4), 46–56 (2005)
29. N. Gatti, Game theoretical insights in strategic patrolling: model and algorithm in normal-form, in *Proceedings of the 18th European Conference on Artificial Intelligence ECAI'08*, Patras, Greece, 21–25 July 2008, pp. 403–407
30. N. Agmon, S. Kraus, G.A. Kaminka, V. Sadov, Adversarial uncertainty in multi-robot patrol, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI'09*, Pasadena, California, 11–17 July 2009
31. E. Halvorson, V. Conitzer, R. Parr, Multi-step multi-sensor hide-seeker games, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI'09*, Pasadena, California, 11–17 July 2009
32. T. Roughgarden, Stackelberg scheduling strategies. *SIAM J. Comput.* **33**(2), 332–350 (2004)
33. J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, P. Parachuri, Deployed ARMOR protection: the application of a game-theoretic model for security at the Los Angeles International Airport, in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'08*, Estoril, Portugal, 12–16 May 2008
34. J. Tsai, S. Rath, C. Kiekintveld, F. Ordóñez, M. Tambe, IRIS—A tool for strategic security allocation in transportation networks, in *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems AAMAS'09*, Budapest, Hungary, 10–15 May 2009
35. J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, E. Steigerwald, Guards—game theoretic security allocation on a national scale, in *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems AAMAS'11*, Taipei, Taiwan, 2–6 May 2011
36. V. Conitzer, T. Sandholm, Computing the optimal strategy to commit to, in *Proceedings of the 7th ACM Conference on Electronic Commerce EC'06*, Ann Arbor, Michigan, 11–15 June 2006, pp. 82–90
37. P. Paruchuri, J.P. Pearce, J. Marecki, M. Tambe, F. Ordonez, S. Kraus, Playing games with security: an efficient exact algorithm for Bayesian Stackelberg games, in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'08*, Estoril, Portugal, 12–16 May 2008, pp. 895–902
38. M. Jain, E. Kardes, C. Kiekintveld, M. Tambe, F. Ordonez, Security games with arbitrary schedules: a branch and price approach, in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence AAAI'10*, Atlanta, Georgia, 11–15 July 2010
39. H. von Stackelberg, *Marktform und Gleichgewicht* (Springer Verlag, Vienna, 1934)
40. G. Leitmann, On generalized Stackelberg strategies. *Optim. Theory Appl.* **26**(4), 637–643 (1978)
41. T. Basar, G.J. Olsder, *Dynamic Noncooperative Game Theory* (Academic Press, San Diego, California, 1995)
42. B. von Stengel, S. Zamir, *Leadership with commitment to mixed strategies*, London School of Economics LSE, Computational, Discrete, and Applied Mathematics (CDAM) Report Series, Technical Report LSE-CDAM-2004-01 (2004)

Towards Analytical Techniques for Optimizing Knowledge
Acquisition, Processing, Propagation, and Use in
Cyberinfrastructure and Big Data

Lerma, L.O.; Kreinovich, V.

2018, VIII, 141 p., Hardcover

ISBN: 978-3-319-61348-2