Scheduling Bodyguards

Loe Schlicher*[1], Kyle Y. Lin[2], and Moshe Kress[2]

[1]*Eindhoven University of Technology, Department of Industrial Engineering and Innovation Sciences , P.O. Box 513, 5600 MB, Eindhoven, The Netherlands*
[2]*Naval Postgraduate School, Operations Research Dept., CA 93943, United States*

November 27, 2024

Abstract

Security agencies throughout the world use bodyguards to protect government officials and public figures. In this paper, we consider a two-person zero-sum game between a defender who allocates such bodyguards to protect several targets and an attacker who chooses one target to attack. Because the number of feasible bodyguard allocations grows quickly as either the number of targets or the number of bodyguards increases, solving the game by brute force with a linear program becomes computationally intractable for problems of practical size. By assuming that the marginal benefit of each additional bodyguard assigned to a target is non-increasing, we show that we can solve the game with a different linear program whose size is linear in the number of targets and the number of bodyguards, respectively. Next, we extend the allocation game to a scheduling game, which allows a bodyguard to report to multiple targets if their schedules allow. We develop an algorithm to compute a bound for the value of this bodyguard scheduling game and present a mixed strategy that achieves this bound in all numerical experiments.

1 Introduction

Security agencies throughout the world—such as the Secret Service in the US, the Protected Persons Service in the United Kingdom, or the Dienst Koninklijke en Diplomatieke Beveiliging in The Netherlands—are in charge of protecting government officials (e.g., cabinet ministers, lawyers, and judges) as well as public figures (e.g., television hosts, crime journalists, or accomplished scientists) from violent events such as terrorist attacks or political violence. The security agencies typically do so by assigning bodyguards to individuals with the mission to protect them against any threat.

In the Netherlands, the number of protected individuals has increased tenfold in the last twenty years (Start, 2023) and it appears that this number is only increasing, while the capacity to protect these individuals is lagging behind (Nachtegael, 2024). As stated in an official report by the Dutch Ministry of Justice and Security (Zouridis, 2023), this potential lack of protection might already have led to the assassination of a prominent Dutch crime journalist, a Dutch lawyer in charge of defending a key witness as well as a family member of the key witness. Based on this report, the Dutch cabinet decided to increase the total budget of the Dutch Protection and Security program with 112 million euros yearly, to hire and train new personnel, but also to fund scientific research to improve the effectiveness and efficiency of the program (Kaag, 2023).

*Corresponding Author: l.p.j.schlicher@tue.nl

In this paper, we develop a game theoretical model that can assist the Dutch Protection and Security program assigning bodyguards to targets (i.e., individuals under threat). This game is zero-sum and played between an *attacker* and a *defender*. The defender allocates a number of bodyguards among several targets to protect them, and the attacker chooses—simultaneously—one target to attack. By assigning more bodyguards to a target, the defender reduces the damage caused by an attack, where damage is broadly construed as any consequence undesirable for the defender, such as people getting hurt, damaged properties, chaos, among other things. We assume that the defender wants to minimize the expected damage from an attack while the attacker wants to maximize it.

The reason for studying a *two*-person *zero*-sum game—as opposed to a *multi*-player *non-zero* sum game between a security agency and several adversaries, each interested in one subject—stems from the fact that it is difficult for security agencies to identify all these adversaries, let alone accurately assessing their views/damage of the different targets. For instance, in the Netherlands, thousands of (digital) reports of threats were made against Dutch politicians in 2022, of which many are received via social media (such as 'X') in anonymous form. It is thus most practical to model the adversaries as a *single* strategic opponent whose interest is opposite to that of the security agency. The reason for studying a *simultaneous-move* game—as opposed to a *sequential-move* game—is to capture the fact that the security agency can randomize their security plan each morning and an attack typically requires days or weeks to plan. Moreover, a sequential-move game would make the adversaries unrealistically strong by allowing them to steal the defender's security plan, choose the least protected target, and plan an attack in just hours of time.

In the first part of the paper, we study a *bodyguard allocation game*. In this game, the defender allocates a number of bodyguards among several targets such that each bodyguard is assigned to one target. We assume that the damage incurred when a target is attacked is convex and non-increasing in the number of bodyguards assigned to the target. In other words, assigning more bodyguards to a target reduces the damage caused by an attack, but the marginal damage reduction of each additional bodyguard is decreasing. Because each player has a finite number of pure strategies, linear programming can be used to compute the value of the game and an optimal mixed strategy for each player. Solving the game by linear programming, however, requires us to first enumerate all pure strategies of both players. Whereas the number of pure strategies for the attacker is just the number of targets, the number of pure strategies for the defender grows quickly in the number of targets and bodyguards. For instance, if there are 10 targets and 30 bodyguards, the number of pure strategies for the defender is $\binom{30+10-1}{10-1}$, which is more than 211 million. A direct implementation of the linear program can present computational challenges for problems of practical sizes.

For the bodyguard allocation game, we demonstrate that it is possible to solve the game without enumerating all pure strategies for the defender. To do so, we first show that the best way to implement a defender's mixed strategy –with an expected number $x \in \mathbb{R}_+$ of bodyguards allocated to a target– is to allocate either $\lfloor x \rfloor$ bodyguards or $\lceil x \rceil$ bodyguards with appropriate probabilities. By taking advantage of this property, we are able to solve the bodyguard allocation game by formulating a different linear program, whose size grows linearly in the number of targets and in the number of bodyguards. This approach allows us to solve a bodyguard allocation game with 10 targets and 30 bodyguards within a few seconds.

In the second part of the paper, we study a *bodyguard scheduling game*. Each target is associated with a location, and a start time and an end time. If the locations of two targets are nearby, and the end time of one target is sufficient earlier than the start time of the other target, then a bodyguard can be assigned to protect both targets. For example, a bodyguard can report to a courthouse at 9:00–12:00, and then report to a press conference in the same city at 14:00–15:00. If we represent each target by a node, then we can draw a directional arc from node 1 to node 2, if a bodyguard assigned to target 1 can be assigned next to target 2. A feasible bodyguard schedule is then analogous to a feasible flow in the network consisting of these nodes and arcs to observe all appropriate constraints.

In the bodyguard scheduling game, each player has a finite number of pure strategies, so in theory one can again compute the entire payoff matrix and solve the game by a linear program. For example, if there are 10 targets and 30 bodyguards, then the attacker has 10 pure strategies—one for attacking each target—and the number of the defender's pure strategies is the number of different feasible, integer-valued, flows in the network described above. In the worst case, we may need to screen up to $(30 + 1)^{10} \approx 8.19 \times 10^{14}$ potential assignments to determine which ones are feasible, because in theory each target can have any number of bodyguards between 0 and 30. Our contribution to the bodyguard scheduling game is to develop a much more efficient way—by leveraging our findings for the bodyguard allocation game—to approach the bodyguard scheduling game. Specifically, we develop an algorithm to compute a bound for the value of this bodyguard scheduling game and present a mixed strategy that achieves this bound in all numerical experiments.

The rest of the paper proceeds as follows. Section 2 provides an overview of the major advancements in the research disciplines related to this paper. Section 3 concerns the bodyguard allocation game and Section 4 concerns the bodyguard scheduling game. Section 5 concludes the paper.

2 Overview of related literature

Our work belongs to the stream of literature that uses quantitative modelling to assist police departments and security agencies to make better decisions about resource allocation. Pioneers in this stream of literature are Kolesar et al. (1975) and Chaiken and Dormont (1978). These authors were asked by the New York Police Department to come up with patrol car schedules meeting specified service standards. Variations of these models can be found in Green and Kolesar (1984), Green (1984), Schaack and Larson (1989), and Kolesar and Green (1998). It is worth pointing out that some of these models are in use by police departments in the US (Green and Kolesar, 2004). Some other decisions that are investigated are the dispatching of police cars (Dunnett et al., 2019), the joint decision of dispatching and locating police cars (Adler et al., 2014), the partitioning of a city center into police patrol sectors (Curtin et al., 2010; Camacho-Collados and Liberatore, 2015) and the allocation of police officers and cameras to fight pickpocketing (Schlicher and Lurkin, 2024).

The aforementioned papers do not explicitly model strategic behavior of opponents (e.g., criminals anticipating on police decisions). This is in sharp contrast to the defender-attacker models that expanded significantly after 9/11 (see reviews of Gupta et al. (2020), Hausken (2024), and Hunt and Zhuang (2024)). A central question in these works is how a limited budget should be allocated over a number of potential attack locations, while taking into account the strategic behavior of attackers (see, e.g., Azaiez and Bier (2007), Bier et al. (2007), Bier et al. (2008), Zhuang and Bier (2007), Hausken (2008), Shan and Zhuang (2013), Guan et al. (2017), Baron et al. (2018), and Musegaas et al. (2022)). Similar to these works, we also model our setting as a defender-attack game, but we assume the resources to be *countable objects* (cf. Dahan et al. (2022)), which is in contrast to the other papers that see the resources as financial budget. To the best of our knowledge, papers that (i) are inspired by police operations, (ii) apply game theory, and (iii) consider resources to be countable objects, are limited. Below, we list some exceptions.

A recent example is the work of Wu et al. (2020). The authors investigate how to assign a limited number of police teams over a set of regions in the city center of San Francisco, with the aim to minimize the number of criminal targets. Because criminals can behave strategically, the authors model the interaction between the police department and criminals in each region as a 2x2 zero-sum game, where the police department has to decide to allocate a single police team or not in each region, while the criminals in that region make the decision to commit crime(s) or not. Using data from the San Francisco Police in 2016, the authors showed the potential of their model.

Another example stems from the work of Pita et al. (2009). This work is inspired by a resource allocation problem faced by the police at the Los Angeles international (LAX) airport. The police at LAX uses several security barriers to prevent terrorist attacks, consisting of road checkpoints, police units patrolling the roads to the terminals, as well as security screening and bag checks for passengers. Because resources (i.e., police officers) are scarce, the police need to make choices about the locations to which they allocate resources, while taking into account that adversaries can learn over time (i.e., can learn resource allocation schedules). For that reason, the authors developed a non-zero sum game that randomizes between where to allocate resources (i.e., randomizes between roadways entering the airport and canine patrol routes within the airport terminals). The authors formulated a mixed-integer linear program that is able to identify an optimal randomized allocation strategy for real-life instances. Notably, the authors also developed a support system (based on the non-zero sum game) that is already in use at LAX for more than a decade.

As a final example, we discuss the work of Jain et al. (2012), which is an extension/variation of the work of Pita et al. (2009), applied to the allocation of air marshals to flights. This air marshal application comes close to our bodyguard allocation/scheduling game, because similar to bodyguards, air marshals are also subject to travel constraints. For instance, if an air marshal is assigned to a flight from Los Angeles to San Francisco, then its next flight should have San Francisco as its departure airport. Likewise, if a bodyguard is assigned to a certain target it is not possible to reassign it to another target at a different location immediately, because bodyguards need to travel first. Dealing with such travel constraints substantially complicates the analysis of a game and this holds for the air marshal game as well. Instead of solving this game via brute force naively, Jain et al. (2012) propose a mixed integer linear program, where decision variables reflect the percentage of time air marshals are allocated to schedules—a feasible combination of consecutive flights. Because these schedules consists of 2 to 3 flights per day only, their mixed integer linear program is able to solve real-life instances daily (i.e., instances with hundreds of air marshals and thousands of flights).

Despite the apparent overlap with our work, we do not believe that Jain et al. (2012)'s results can be easily applied to our setting, mainly for two reasons. First, in contrast to the work of Jain et al. (2012) where at most *one* air marshal is allocated to a single flight, we allow for an assignment of *multiple* bodyguards to a single target. Second, it seems that the solution method of Jain et al. (2012) is tailor made—and so leveraged—for a setting where (i) the number of air marshals is much smaller than the number of flights and (ii) the defender and the attacker have different utility functions. This is in sharp contrast to our setting where many bodyguards can be assigned to protect the same target and the defender and the attacker have opposite interests.

3 The bodyguard allocation game

Consider a two-person zero-sum game $\mathscr{G}$ between an attacker and a defender. The defender has a total of $k \in \mathbb{N}_+$ agents to allocate among a set $N = \{1, 2, \ldots, n\}$ of $n \in \mathbb{N}_+$ targets. If $z \in \mathbb{N}_{\geq 0}$ bodyguards are assigned to target $i \in N$, then the attacker can cause damage $g_i(z)$ by attacking target i, with $g_i : \mathbb{N}_{\geq 0} \to \mathbb{R}_{\geq 0}$. We assume that each additional bodyguard assigned to target i reduces the damage the attacker can cause by attacking target i, but the marginal reduction is diminishing, i.e., g_i is convex and non-increasing. In addition, there exists a $b_i \in \mathbb{N}_+$ such that the marginal benefit for additional bodyguards beyond b_i is zero. In other words, $g_i(z) = g_i(b_i)$ for all $z \geq b_i$, for $i \in N$. This assumption is not restrictive, because we can set $b_i = k$ for all $i \in N$.

A pure strategy for the defender can be delineated by $(z_1, z_2, \ldots, z_n)$, where $\sum_{i=1}^{n} z_i = k$, with the interpretation that $z_i \in \mathbb{N}_{\geq 0}$ is the number of bodyguards assigned to target $i \in N$. The number of defender's pure strategies is *at most* $\binom{k+n-1}{n-1}$, which equals the number of nonnegative integer solutions to $\sum_{i=1}^{n} z_i = k$. A pure strategy for the attacker is an integer in N, which corresponds to the target they choose to attack. The number of the attacker's pure strategies is n. The attacker chooses which target to attack in order to maximize the expected damage, while the defender chooses how to allocate the bodyguards to minimize it.

Because the number of pure strategies in $\mathscr{G}$ for each player is finite, the two-person zero-sum game $\mathscr{G}$ has a finite payoff matrix. Linear programming can be used to compute the value of the game and an optimal mixed strategy for each player. We next demonstrate the game $\mathscr{G}$ with an example with $n = 2$ targets and $k = 2$ bodyguards.

Example 1. *Consider a setting with* $n = 2$ *targets,* $k = 2$ *bodyguards,* $b_1 = b_2 = 2$*, and* $g_1(0) = 0.8, g_1(1) = 0.6, g_1(2) = 0.5, g_2(0) = 0.6, g_2(1) = 0.4$ *and* $g_2(2) = 0.3$*. The payoff matrix is given below, where the rows correspond to the attacker's pure strategies and the columns correspond to the defender's pure strategies. The attacker wants to maximize the expected damage, while the defender wants to minimize it.*

$$\begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} (2,0) & (1,1) & (0,2) \end{array} \\ \begin{bmatrix} 0.5 & 0.6 & 0.8 \\ 0.6 & 0.4 & 0.3 \end{bmatrix} \end{array}$$

It is straightforward to verify that it is optimal for the attacker to attack target 1 with probability $2/3$ *and attack target 2 with probability* $1/3$*. It is optimal for the defender to use* $(2, 0)$ *with probability* $2/3$ *and* $(1, 1)$ *with probability* $1/3$*. The value of the game is* $8/15$*.*

While in theory it is possible to solve $\mathscr{G}$ by first laying out the entire payoff matrix and then solving a linear program, as demonstrated in Example 1, this approach quickly becomes computationally intractable as n and k increase. For example, if there are $n = 10$ targets and $k = 20$ bodyguards, then the defender has more than 10 million pure strategies.

The rest of this section presents a method to solve $\mathscr{G}$ with computational effort that is orders of magnitude smaller than what is required to solve it via the entire payoff matrix.

3.1 Analysis of the bodyguard allocation game

In this section, we analyze the bodyguard allocation game. We identify an attacker's and defender's optimal strategy and develop a method to compute the value of the game without having to compute the entire payoff matrix.

To do so, we first introduce some new concepts and definitions. For any defender's mixed strategy, we can compute a corresponding vector $x = (x_1, \ldots, x_n)$, where $x_i \in \mathbb{R}_{\geq 0}$ represents the expected number of bodyguards assigned to target $i \in N$.[1] We call a defender's mixed strategy *consistent* if

- it always assigns x_i bodyguards to target i if x_i is an integer, and otherwise
- (so if x_i is not integer) it assigns $\lfloor x_i \rfloor$ bodyguards to target i with probability $\lceil x_i \rceil - x_i$ and assigns $\lceil x_i \rceil$ bodyguards to target i with probability $x_i - \lfloor x_i \rfloor$.

For any given vector $(x_1, \ldots, x_n)$ with $\sum_{i=1}^{n} x_i = k$, where x_i is the expected number of bodyguards assigned to target $i \in N$, it is always possible to construct a consistent mixed strategy. One way to do it is to first divide $[0, k]$ into n subintervals with lengths $x_1, x_2, \ldots, x_n$ as follows:

$$[0, x_1), [x_1, x_1 + x_2), \cdots, \left[\sum_{i=1}^{n-2} x_i, \sum_{i=1}^{n-1} x_i\right), \left[\sum_{i=1}^{n-1} x_i, k\right],$$

with subinterval i corresponding to target $i \in N$. Next, draw a number u from the uniform distribution over the interval $(0, 1)$. Find the points $u, u+1, u+2, \ldots, u+k-1$, and identify the subintervals each of these points belongs to. Finally, assign the k bodyguards to the targets corresponding to these subintervals. It is straightforward to verify that this mixed strategy is consistent.

Example 2. *Reconsider the setting of Example 1 and let vector $x = (1\frac{2}{3}, \frac{1}{3})$. As illustrated in Figure 1 below, if $u \in (0, \frac{2}{3})$ we assign two bodyguards to target 1. On the other hand, if $u \in (\frac{2}{3}, 1)$ we assign one bodyguard to target 1 and one bodyguard to target 2.*

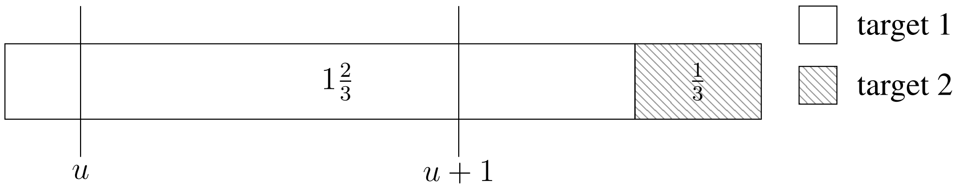

Figure 1: Visualisation of setting with $n = 2$, $k = 2$ and $x = (1\frac{2}{3}, \frac{1}{3})$

We now show that to find an optimal defender's strategy, it is sufficient for the defender to consider only consistent mixed strategies. We give a lemma before proving this result.

Lemma 1. *Let W be a nonnegative integer-valued random variable with $\mathbb{E}[W] = c \in \mathbb{R}_{>0}$. If c is integer and $\mathbb{P}\{W = c\} = 1$ or if c is non-integer and*

$$\mathbb{P}\{W = \lfloor c \rfloor\} = \lceil c \rceil - c,$$
$$\mathbb{P}\{W = \lceil c \rceil\} = c - \lfloor c \rfloor,$$

then for any convex and non-increasing function $g(\cdot)$, $\mathbb{E}[g(W)] \leq \mathbb{E}[g(W')]$ for all nonnegative integer-valued random variables W' with $\mathbb{E}[W'] = c$.

Proof. Define $d_z = g(z-1) - g(z)$, for $z \in \mathbb{N}_{>0}$. Because g is convex and non-increasing, we have

$$d_1 \geq d_2 \geq \cdots \geq 0.$$

[1] Note that $\sum_{i \in N} x_i = k$.

Moreover, the objective function $\mathbb{E}[g(W'))]$ can be computed as follows:

$$\begin{aligned}\mathbb{E}[g(W')] &= \sum_{z=0}^{\infty} g(z)\mathbb{P}\{W'=z\} \\ &= \sum_{z=0}^{\infty}\left(g(0)-\sum_{j=1}^{z} d_j\right)\mathbb{P}\{W'=z\} \\ &= g(0)-\sum_{z=0}^{\infty}\sum_{j=1}^{z} d_j\mathbb{P}\{W'=z\}\end{aligned}$$

To minimize the preceding, it is equivalent to maximize

$$\begin{aligned}\sum_{z=0}^{\infty}\sum_{j=1}^{z} d_j\mathbb{P}\{W'=z\} &= \sum_{j=1}^{\infty}\sum_{z=j}^{\infty} d_j\mathbb{P}\{W'=z\} \\ &= \sum_{j=1}^{\infty} d_j\mathbb{P}\{W'\geq j\}. \end{aligned} \tag{1}$$

Recall that the expected value of a nonnegative integer-valued random variable can be computed by $\mathbb{E}[W'] = \sum_{j=1}^{\infty}\mathbb{P}\{W'\geq j\}$. Writing $y_j = \mathbb{P}\{W'\geq j\}$ as decision variables with $j\in\mathbb{N}_{>0}$, we can formulate the following linear programming model to maximize (1).

$$\begin{aligned}\max \quad & \sum_{j=1}^{\infty} d_j y_j, \\ \text{subject to} \quad & \sum_{j=1}^{\infty} y_j = c, \\ & 1\geq y_1\geq y_2\geq\cdots\geq 0.\end{aligned}$$

Because d_j is non-increasing in j, to maximize the preceding an optimal solution is to let

$$y_j = \begin{cases} 1, & j=1,\ldots,\lfloor c\rfloor, \\ c-\lfloor c\rfloor, & j=\lfloor c\rfloor+1, \\ 0, & j\geq\lfloor c\rfloor+2.\end{cases}$$

Because $\mathbb{P}\{W'\geq 1\} = \mathbb{P}\{W'\geq 2\} = \cdots = \mathbb{P}\{W'\geq c\} = 1$, $\mathbb{P}\{W'=n'\}=0$ for $n'<c$. Moreover, because $\mathbb{P}\{W'\geq c+2\}=0$, we have $\mathbb{P}\{W=n'\}=0$ for all $n'\geq c+2$. Thus, $\mathbb{P}\{W=c\}=1$ or $\mathbb{P}\{W=c\}=1-c+\lfloor c\rfloor$ and $\mathbb{P}\{W=c+1\}=c-\lfloor c\rfloor$ maximizes the objective of (1) and thus $\mathbb{E}[g(W)]\leq\mathbb{E}[g(W')]$ for all nonnegative integer-valued random variables W' with $\mathbb{E}[W']=c$. □

Theorem 1. *There is always an optimal strategy that is a consistent mixed strategy.*

Proof. Consider an arbitrary mixed strategy for the defender and let X_i be a non-negative integer-valued random variable, denoting the number of bodyguards assigned to target $i\in N$ with $\mathbb{E}[X_i]=x_i$. Because g_i is a nonnegative integer-valued function that is non-increasing and convex for all $i\in N$, applying Lemma 1 it follows that the expected damage if target i is attacked—namely, $E[g_i(X_i)]$—is minimized if X_i takes on the two integers surrounding x_i, or just x_i if it happens to be an integer, for $i\in N$. Consequently, any of the defender's mixed strategy that is not consistent is dominated by a consistent mixed strategy, which completes the proof. □

3.2 The defender's optimal strategy

According to Theorem 1, to find an optimal defender strategy it is sufficient to consider only consistent mixed strategies. Recall that $g_i(z)$ is the damage from an attack of target $i \in N$ if it is protected by z bodyguards. Without loss of generality, we assume that the targets are labeled in such a way that $g_1(0) \geq g_2(0) \geq \cdots \geq g_n(0)$. In other words, without any bodyguards, target 1 has the highest value and target n the lowest value. We define $h_i(x_i)$ as the expected damage if the attacker attacks target $i \in N$ when the defender uses a consistent mixed strategy (with induced vector x) that assigns an expected number of x_i bodyguards to target i, for $x_i \in [0, b_i]$. In other words, the function $h_i : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ for all $i \in N$ is defined by

$$h_i(x_i) = \begin{cases} g_i(x_i), & \text{if } x_i \text{ is an integer,} \\ (\lceil x_i \rceil - x_i) g_i(\lfloor x_i \rfloor) + (x_i - \lfloor x_i \rfloor) g_i(\lceil x_i \rceil), & \text{if } x_i \text{ is not an integer.} \end{cases}$$

Consequently, the defender's game can be formulated as

$$\min_{x \in \mathscr{X}} \big\{ \max_{i \in N} h_i(x_i) \big\}, \tag{2}$$

with $\mathscr{X} = \{(x_i)_{i \in N} | x_i \geq 0 \, \forall i \in N, \sum_{i \in N} x_i = k\}$. The game in (2) can be solved by a greedy algorithm: keep allocating *fractional* bodyguards to targets having the highest present expected damage until the defender runs out of bodyguards, or until target i has received b_i bodyguards for some $i \in N$ so the objective function cannot be reduced further. [2] Consequently, an optimal solution, which we denote by x^*, has the property that for some $t \in N$, we have $h_1(x_1^*) = h_2(x_2^*) = \cdots = h_t(x_t^*)$ and $h_j(x_j^*) = h_j(0)$ for $j \geq t + 1$. We denote the optimal value by r^*.

Example 3. *Reconsider the setting of Example 1. The defender wants to solve*

$$\min_{(x_1, x_2) \in \mathscr{X}} \left\{ \max \big\{ \max\{0.8 - 0.2x_1, 0.7 - 0.1x_1, 0.5\}, \max\{0.6 - 0.2x_2, 0.5 - 0.1x_2, 0.3\} \big\} \right\}$$

The optimal solution is $x^* = \left(1\frac{2}{3}, \frac{1}{3}\right)$ with $r^* = \frac{8}{15}$. If, however, the defender has only $k = 1$ bodyguard, we would end up with optimal solution $x^* = (1, 0)$ and $r^* = 0.6$. In Figure 2, we also visualize the progression of the greedy algorithm with 0, 1, and 2 bodyguards.

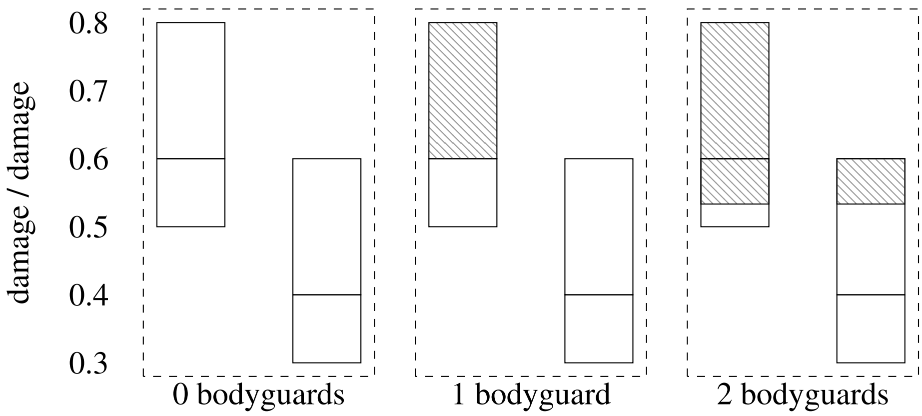

Figure 2: Allocation for 0, 1 and 2 bodyguards.

[2] For instance, this might happen if $b_i = 1$ for all $i \in N$, $k = 2$ and $g_1(1) \geq g_i(0)$ for all $i \in N$. For this setting, there is no need to allocate more than one bodyguard, which is assigned to target 1.

Because the defender has a mixed strategy that guarantees the expected damage to be at most r^* regardless of which target the attacker chooses to attack, r^* is an upper bound for the value of the game. We now give an expression for this upper bound r^*. Recall that in an optimal solution to (2), there exists $t \in \mathbb{N}_+$ such that the first t targets receive some bodyguards, while all others targets do not receive any bodyguard. Write $T = \{1, 2, \ldots, t\}$ for convenience. Because $x_i^* = 0$ for $i \notin T$, it follows that

$$k = \sum_{i=1}^{n} x_i^* = \sum_{i \in T} x_i^*.$$

For $i \in T$, write $x_i^* = m_i + y_i$, where $m_i = \lfloor x_i^* \rfloor$ is the integral part and $y_i = x_i^* - m_i \in [0, 1)$ is the fractional part. We next develop a formula to compute the optimal value of (2), r^*, as a function of T and m_i, $i \in T$, so that we can compute r^* directly once we identify T and m_i, $i \in T$.

Since $h_i(x_i^*) = r^*$ for $i \in T$, by the definition of h_i we must have

$$r^* = h_i(x_i^*) = (1 - y_i) g_i(m_i) + y_i g_i(m_i + 1),$$

for $i \in T$. Solving for y_i yields

$$y_i = \begin{cases} \frac{g_i(m_i) - r^*}{g_i(m_i) - g_i(m_i+1)} & \text{if} \quad m_i \le b_i - 1 \\ 0 & \text{if} \quad m_i \ge b_i \end{cases} \tag{3}$$

for $i \in T$. Note that $y_i = 0$ follows from the fact that $g_i(m_i) = r^*$ if $m_i = b_i$.

Define $s \equiv \sum_{i \in T} y_i$, which is the sum of all the fractional parts of the bodyguard allocations, so

$$k = \sum_{i \in T} x_i^* = \sum_{i \in T} m_i + \sum_{i \in T} y_i = \sum_{i \in T} m_i + s.$$

Note that $s < t$ because $y_i < 1$ for $i \in T$. Using (3), we have that

$$\begin{aligned} s \equiv \sum_{i \in T} y_i &= \sum_{i \in T} \frac{g_i(m_i)}{g_i(m_i) - g_i(m_i + 1)} - r^* \sum_{i \in T} \frac{1}{g_i(m_i) - g_i(m_i + 1)} \\ &= \sum_{i \in T} \frac{g_i(m_i)}{g_i(m_i) - g_i(m_i + 1)} - \frac{r^*}{\lambda}, \end{aligned} \tag{4}$$

where we have defined

$$\lambda_T = \left(\sum_{i \in T} \frac{1}{g_i(m_i) - g_i(m_i + 1)} \right)^{-1}. \tag{5}$$

Solving r^* from (4) gives

$$r^* = \left(\sum_{i \in T} \frac{g_i(m_i)}{g_i(m_i) - g_i(m_i + 1)} - s \right) \lambda_T. \tag{6}$$

The next subsection shows that r^* in (6) is also a lower bound for the value of the game $\mathscr{G}$, so that r^* is the value of the game.

3.3 The attacker's optimal strategy

We now present an attacker's strategy that guarantees an expected damage at least r^* for the attacker regardless of what the defender does, which proves that r^* is also a lower bound for the value of the game.

A mixed strategy for the attacker can be delineated by $(p_1, \ldots, p_n)$ with $\sum_{i=1}^n p_i = 1$, where $p_i \geq 0$ is the probability of attacking target $i \in N$. Consider the attacker's strategy with

$$p_i = \begin{cases} \dfrac{\lambda_T}{g_i(m_i) - g_i(m_i+1)}, & i \in T, \\ 0, & i \notin T. \end{cases} \tag{7}$$

where λ is defined in (5). We will show that this attacker's strategy guarantees expected damage at least r^* regardless of what the defender does. Given the attacker's strategy in (7), what can the defender do to minimize the expected damage? The defender chooses $(z_i)_{i\in N} \in \mathbb{N}_{\geq 0}^N$ in order to minimize

$$\sum_{i=1}^n p_i g_i(z_i) = \sum_{i\in T} p_i g_i(z_i), \tag{8}$$

with the constraint $\sum_{i=1}^n z_i = k$. The equality in the preceding is due to $p_i = 0$ for $i \notin T$. Because $p_i g_i(z_i)$ are convex functions, $i \in T$, it follows that the preceding optimization problem can be solved by a greedy algorithm (see, for example, Lemma 1 in Subelman (1981) or Appendix in Ross and Lin (2001)). That is, to achieve optimality, the defender can allocate the bodyguards one at a time to the target that provides the most reduction in the objective function in (8) at the moment.

Because $g_i(z_i)$ is a convex decreasing function, $i \in T$, we can see that each of the first m_i bodyguards allocated to target i will reduce the objective function in (8) for at least

$$p_i g_i(m_i - 1) - p_i g_i(m_i) = \frac{\lambda_T}{g_i(m_i) - g_i(m_i+1)}(g_i(m_i-1) - g_i(m_i)) > \lambda_T.$$

In addition, after allocating m_i bodyguards to target i, $i \in T$, the (m_i+1)st bodyguard allocated to target i would reduce the objective function by exactly λ_T. Since $\sum_{i\in T} m_i \leq k$, it follows that with the greedy algorithm, after the first $\sum_{i\in T} m_i$ iterations, exactly m_i bodyguards will go to target i, for $i \in T$.

After the first $\sum_{i\in T} m_i$ bodyguards allocated in the greedy algorithm, with m_i bodyguards going to target i, for $i \in T$, the defender still has $k - \sum_{i\in T} m_i = s$ bodyguards to allocate. Because of the choice of p_i in (7), allocating one additional bodyguard to any target $i \in T$ will reduce the objective function by exactly λ_T. Since $s < t$, to minimize the objective function in (8), it is optimal for the defender to choose a subset of s targets from T and allocate one additional bodyguard to each target in this subset. The minimized expected damage achieved in (8) is

$$\sum_{i\in T} p_i g_i(z_i) = \sum_{i\in T} p_i g_i(m_i) - s\lambda_T = \sum_{i\in T} \frac{\lambda_T g_i(m_i)}{g_i(m_i) - g_i(m_i+1)} - s\lambda_T = r^*,$$

where the last equality is due to (6). In other words, the best the defender can do against the attacker's strategy in (7) is to reduce the expected damage to r^*. In other words, the attacker's mixed strategy in (7) guarantees expected damage for at least r^* regardless of what the defender does. Consequently, r^* is a lower bound for the value of our game.

3.4 Solving the game with a linear program

We have shown that r^* in (6) is an upper bound for the value of the game $\mathscr{G}$ in Section 3.2 and also a lower bound in Section 3.3. Consequently, we have proved that r^* is the value of the game $\mathscr{G}$. To compute r^*, one could solve the optimization problem in (2). Another way to compute r^* is to recognize that $h_i(x_i)$ is a piecewise-linear non-increasing function in x_i, for $i \in N$. Therefore, the optimization problem in (2) can be transformed into a linear program as follows:

$$
\begin{aligned}
\min_{x_1,\ldots,x_n,v} \quad & v \\
\text{s.t. } & v \geq (g_i(j+1) - g_i(j)) \cdot (x_i - j) + g_i(j) \; \forall j = 0, 1, \ldots, b_i - 1 \,, \qquad \forall i \in N \\
& v \geq g_i(b_i) \qquad \forall i \in N \\
& \sum_{i \in N} x_i = k \\
& v \geq 0 \\
& x_i \geq 0 \qquad \forall i \in N,
\end{aligned} \tag{9}
$$

The optimal value from the preceding linear program is equal to r^* and an optimal solution $(x_i^*)_{i \in N}$ describes an induced vector of an optimal mixed consistent strategy. This consistent mixed strategy can be obtained by following the procedure at the beginning of Section 3.1. For the attacker it is optimal to use the mixed strategy in (7). We demonstrate this result via an example.

Example 4. *Consider a setting with $n = 4$ targets, $k = 3$ bodyguards, $b_1 = b_2 = 2$, $b_3 = b_4 = 3$ and functions g_i for all $i \in N$ as depicted in Table 1 below.*

z	0	1	2	3
$g_1(z)$	0.9	0.7	0.6	0.6
$g_2(z)$	0.8	0.6	0.5	0.5
$g_3(z)$	0.7	0.5	0.4	0.3
$g_4(z)$	0.6	0.4	0.3	0.2

Table 1: Damage functions

Solving optimization problem (9) *leads to optimal value $r^* = \frac{5}{8}$ with $x = (1\frac{3}{4}, \frac{7}{8}, \frac{3}{8}, 0)$. Hence, we have $T = \{1, 2, 3\}$, $m_1 = 1$, $m_2 = m_3 = m_4 = 0$, $y_1 = \frac{3}{4}, y_2 = \frac{7}{8}$, $y_3 = \frac{3}{8}$, and $y_4 = 0$. For the defender, it is optimal to always allocate one (out of the three bodyguards) to target 1. The remaining two bodyguards could be mixed as follows: defend target 1 and target 2 with probability $5/8$, defend target 1 and target 3 with probability $1/8$ and defend target 2 and target 3 with probability $1/4$ (see also Figure 3). For the attacker, we can use* (7) *to compute $p_1 = 1/2$, $p_2 = p_3 = 1/4$, and $p_4 = 0$. In other words, it is optimal for the attacker to attack target 1 with probability $1/2$, and target 2 and target 3 each with probability $1/4$, and attack target 4 not at all.*

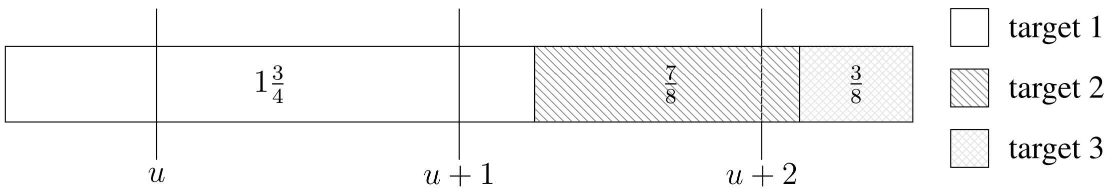

Figure 3: Visualisation of setting with $n = 4$, $k = 3$ and $x = (1\frac{3}{4}, \frac{7}{8}, \frac{3}{8}, 0)$

Recall that in $\mathscr{G}$, the attacker has n pure strategies and the defender has up to $\binom{k+n-1}{n-1}$ pure strategies. To compute an optimal mixed strategy for the attacker via the payoff matrix, the linear program requires $n + 1$ variables and up to $\binom{k+n-1}{n-1}$ constraints. For example, if there are $n = 10$ targets and $k = 20$ bodyguards, then that linear program has 11 variables and more than 10 million constraints. By comparison, the linear program in (9) has $n + 1$ variables and up to $nk + n + 1$ constraints, so its size is linear in both n and k. If $n = 10$ and $k = 20$, then the linear program in (9) has 11 variables and only 211 constraints.

3.5 Some special cases

As discussed in the previous section, we can identify r^*, as well as the associated optimal strategies of both the attacker and defender by solving optimization problem (9). In some special cases, however, it is not necessary to solve problem (9). In this section, we discuss three of them.

3.5.1 At most one bodyguard for each target

In this section, we discuss the special case $b_i = 1$ for all $i \in N$. This setting could, for instance, represent a setting where a security agency believes that the probability of an attack is already low per target (e.g., $g_i(0) \ll 1$ for all $i \in N$) and so at most one bodyguard per target suffices. It could also represent a setting where the security agency wants to limits the amount of input required. That is, if $b_i = 1$ for all $i \in N$ only two data points ($g_i(0)$ and $g_i(1)$) needs to be estimated per target.

It turns out that for this special case, we need to compare $n+2$ values to identify r^*. If $x_i^* \geq 1$ for some $i \in N$ then $i \in \text{argmax}\{g_i(1)\}$ due to the structure of an optimal solution. Consequently, $r^* = \max_{i \in N}\{g_i(1)\}$. Hence, the first value that we need in our comparison is $\max_{i \in N}\{g_i(1)\}$.

Now, suppose that $x_i^* < 1$ for all $i \in N$ and thus $m_i = 0$ for all $i \in N$ and $s = k$. For this setting, we focus on $n+1$ candidate optimal solutions of optimization problem (9), namely those $(x_i)_{i \in N}$ for which $h_i(x_i) = h_j(x_j)$ for all $i, j \in T'$ with $T' \in \{\{1\}, \{1,2\}, \ldots, \{1,2,\ldots,n\}\}$. Using the derivations of (3)–(6), we know that for each $T' \in \{\{1\}, \{1,2\}, \ldots, \{1,2,\ldots,n\}\}$ we have

$$h_i(x_i) = h_j(x_j) = \left(\sum_{i' \in T'} \frac{g_{i'}(0)}{g_{i'}(0) - g_{i'}(1)} - k \right) \lambda_{T'} \text{ for all } i, j \in T'. \tag{10}$$

Suppose that $T^* = \{1, 2, \ldots, t^*\}$ corresponds to an optimal solution. Then,

$$\left(\sum_{i' \in T'} \frac{g_{i'}(0)}{g_{i'}(0) - g_{i'}(1)} - k \right) \lambda_{T'} \leq \left(\sum_{i' \in T^*} \frac{g_{i'}(0)}{g_{i'}(0) - g_{i'}(1)} - k \right) \lambda_{T^*} \tag{11}$$

for all $T' \in \{\{1\}, \{1,2\}, \ldots, \{1,2,\ldots,n\}\}$. This holds for the following reason. If $T' = \{1, 2, \ldots, t'\}$ with $t' < t^*$ then (11) holds, because the bodyguards allocated to target t^* can be allocated over the first $t^* - 1$ targets. Note, this is only possible because $x_{t^*} > 0$. If $t' > t^*$ then (11) holds, because $x_j < 0$ for all $j = t^*+1, t^*+2, \ldots, t'$, implying that fictitious bodyguards are introduced and allocated over the first t^* targets. Hence, from equation (11) we learn that

$$\max_{T' \in \{\{1\},\{1,2\},\ldots,\{1,2,\ldots,n\}\}} \left\{ \left(\sum_{i \in T'} \frac{g_i(0)}{g_i(0) - g_i(1)} - k \right) \lambda_{T'} \right\} = \sum_{i \in T^*} \left(\frac{g_i(0)}{g_i(0) - g_i(1)} - k \right) \lambda_{T^*}.$$

In conclusion, to identity r^*, we need to compare the n candidate optimal solutions of optimization problem (9) with $\max_{i \in N}\{g_i(1)\}$. The maximum of these values coincides with r^*, i.e.,

$$r^* = \max \left\{ \max_{i \in N} g_i(1), \max_{T' \in \{\{1\},\{1,2\},\ldots,\{1,2,\ldots,n\}\}} \left\{ \left(\sum_{i \in T'} \frac{g_i(0)}{g_i(0) - g_i(1)} - k \right) \lambda_{T'} \right\} \right\}.$$

Lidbetter and Lin (2020) study a booby trap game in which one player can booby trap k out of a total of $n > k$ boxes and the other player opens 1 box to either get the reward in the box if the box is not booby trapped, or get nothing if it is booby trapped. This booby trap game is a special case of our bodyguard allocation game $\mathscr{G}$ if $g_i(1) = 0$ for all $i \in N$.

3.5.2 Exponential damage function

Suppose function g_i has an exponential form, i.e., $g_i(z) = \gamma_i \cdot \alpha_i^z$ with $z \in \mathbb{N}_{\geq 0}$, $\gamma_i, \in \mathbb{R}_{>0}$, and $\alpha_i \in (0, 1)$ for all $i \in N$. One way to interpret this damage function is that each additional bodyguard adds a defense layer for the target. The attacker penetrates each defense layer of target $i \in N$ with probability α_i, independently of everything else, and succeeds in the attack only if the attacker penetrates all defense layers. Moreover, γ_i could be interpreted as the importance/societal value of a target (e.g., this value could be extremely high for the prime minister). In the special case that all targets are equally important, so that $g_i(z) = \alpha_i^z$ for $i \in N$, the game $\mathscr{G}$ is mathematically equivalent to a hide-search game studied in Subelman (1981) between a hider and a searcher. In the hide-search game studied in Subelman (1981), the hider chooses to hide in one of n locations, while the searcher decides how to allocate k searches among these n locations. The searcher wants to maximize the probability of finding the target within these k searchers, while the hider wants to minimize it. Each search in location i will independently find the target—if the target is hidden there—with probability $1 - \alpha_i$, for $i \in N$. Therefore, if the searcher searches location i for z times, then the probability of not finding the target is α_i^z, for $i \in N$. The searcher decides how to distribute the k searches among the n locations in this hide-search game, just as in our game $\mathscr{G}$ the defender decides how to allocate k bodyguards among the n targets. Subelman (1981) develops an algorithm for this special case, which involves maximizing a characteristic function and using its solution to compute the optimal strategy for each player. The algorithm developed in this section is more powerful because it works as long as the damage functions are non-increasing and convex in the number of bodyguards assigned.

3.5.3 Homogeneous targets

Suppose that all targets have the same damage function, so $g_i = g$ for all $i \in N$. For this case, it follows immediately from optimization problem (9) that it is optimal to allocate bodyguards evenly—probabilistically if needed—among the targets. In other words, first allocate $\lfloor k/n \rfloor$ bodyguards to each target, and then choose $k - n\lfloor k/n \rfloor$ targets uniformly randomly and allocate one additional bodyguard to each of these targets. The value of the game subsequently reads

$$r^* = g\left(\left\lfloor \frac{k}{n} \right\rfloor\right) - \left(\frac{k}{n} - \left\lfloor \frac{k}{n} \right\rfloor\right)\left(g\left(\left\lfloor \frac{k}{n} \right\rfloor\right) - g\left(\left\lfloor \frac{k}{n} \right\rfloor + 1\right)\right). \quad (12)$$

4 The bodyguard scheduling game

The bodyguard allocation game $\mathscr{G}$ assumes that each bodyguard can be allocated to exactly one target. In this section, we study a bodyguard *scheduling* game, denoted by $\mathscr{G}_S$, which extends $\mathscr{G}$ by allowing a bodyguard to be assigned more targets subject to appropriate schedule constraints.

In the bodyguard scheduling game $\mathscr{G}_S$, we associated to each target $i \in N$ a start time $t_i^s \in \mathbb{R}_{\geq 0}$ and an end time $t_i^e \in \mathbb{R}_{\geq 0}$.[3] Moreover, for each target $i \in N$ we draw a node and use a directional arc between each pair of nodes $i, j \in N$ for which the start time of target j is later than the end time of i, i.e., $t_i^e \leq t_j^s$. For each directional arc, we also add a flow capacity $q_{ij} \in \mathbb{N}_{\geq 0}$ to indicate that at most q_{ij} bodyguards can be assigned to protect target j after they have completed their assignment for target i.[4] Such a restriction could, for instance, represent a specific travel regulation set by a security agenda. Similar to the bodyguard allocation game we write $z_i \in \mathbb{N}_{\geq 0}$ for the number of bodyguards assigned to target $i \in N$. We use $w_{ij} \in \mathbb{N}_{\geq 0}$ to indicate the number of bodyguards who report to target $j \in N$ directly after their assignment at target $i \in N$. In addition, we write $w_{0j} \in \mathbb{N}_{\geq 0}$ for the number of bodyguards whose first assignment is to protect target j, where node 0 can be interpreted

[3]For instance, we use 7.5 to represent 7:30AM.

[4]In the remainder of this paper, we will only show those directional arcs for which $q_{ij} > 0$.

as the source node (e.g., a security headquarter). We denote a pure strategy for the defender as a *feasible flow* in the network. Formally, a pure strategy for the defender $(z_1, z_2, \dots, z_n)$ is feasible if there exists flows $(w_{0j})_{j\in N}$ and $(w_{ij})_{i,j\in N}$ that satisfy the following flow/schedule constraints:

$$\begin{aligned}
&\sum_{j\in N} w_{0j} = k \\
&w_{0j} + \sum_{i\in N} w_{ij} = z_j \qquad \forall j \in N \\
&\sum_{j\in N} w_{ij} \le z_i \qquad \forall i \in N \\
&w_{ij} \in \{0, 1, \dots, q_{ij}\} \qquad \forall i, j \in N,
\end{aligned} \tag{13}$$

The first constraint ensures that we use exactly k bodyguards. The next constraint equates the number of bodyguards reporting to each target (left-hand side) to the number of bodyguards assigned to the target (right-hand side). The next inequality ensures that the number of bodyguards leaving from a target (left-hand side) cannot exceed the number of bodyguards assigned to it (right-hand side). Finally, the variables $(w_{ij})_{i,j\in N}$ and $(z_i)_{i\in N}$ must be non-negative integers and should not exceed their respective upper bounds. We would like to mention that the constraint matrix, resulting from constraints (13), is totally unimodular. Hence, there is no need to enforce w_{ij} to be integer: one can relax it and use linear programming to check pure strategy $(z_1, z_2, \dots, z_n)$ on feasibility.

A pure strategy for the attacker in $\mathscr{G}_S$ is an integer in N, which corresponds to the target the attacker chooses to attack. We next demonstrate $\mathscr{G}_S$ with a simple example.

Example 5. *Consider a setting with* $n = 3$ *targets and* $t_1^s = 9, t_1^e = 12$*,* $t_2^s = 14, t_2^e = 18$ *and* $t_3^s = 8, t_3^e = 15$*. Suppose there is* $k = 1$ *bodyguard,* $b_1 = b_2 = b_3 = 1$ *and* $g_1(0) = 0.8, g_1(1) = 0.4$*,* $g_2(0) = 0.6, g_2(1) = 0.3$*,* $g_3(0) = 0.5$ *and* $g_3(1) = 0.3$*. Moreover,* $q_{12} = 1$ *and* $q_{ij} = 0$ *for all other combinations of* $i, j \in N$*. A visual representation is depicted in Figure 4.*

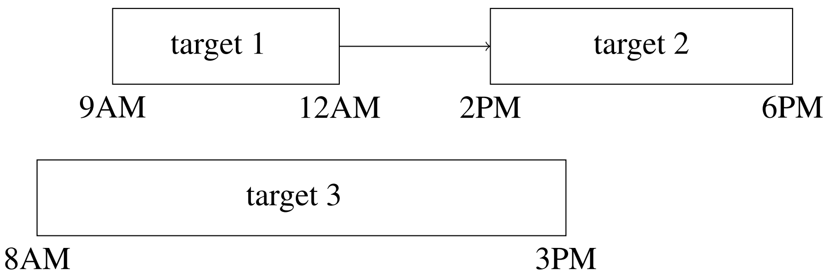

Figure 4: Visualisation of the setting with $n = 3$ targets and schedule constraints. The arc indicates that the bodyguard assigned to target 1 can travel to target 2 afterwards.

All feasible pure strategies of the defender are given by $(1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 0)$*. Note that strategy* $(1, 1, 0)$ *is the only "new" strategy, compared to a setting without travel possibilities, and it dominates two strategies, namely* $(1, 0, 0)$ *and* $(0, 1, 0)$*. The associated damage for each relevant combination of strategies is presented below, with the rows representing the pure attacker strategies and the columns representing the relevant pure defender strategies.*

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} (0,0,1) & (1,1,0) \end{array} \\ \begin{bmatrix} 0.8 & 0.4 \\ 0.6 & 0.3 \\ 0.3 & 0.5 \end{bmatrix} \end{array}$$

By inspection, we learn that strategy 2 of the attacker is dominated by strategy 1 (i.e., $0.8 > 0.6$ *and* $0.4 > 0.3$*). For the 2x2 matrix, it is optimal for the defender to mix strategies and protect target 3 with one bodyguard with probability* $1/6$ *and protect target 1 and target 2—with the same bodyguard—with probability* $5/6$*. For the attacker, it is optimal to attack target 1 with probability* $1/3$ *and attack target 3 with probability* $2/3$*. Consequently, the value of the game is* $7/15$*.*

One way to solve $\mathscr{G}_S$ is to enumerate all pure strategies for each player and compute each player's optimal mixed strategy by linear programming. To enumerate all pure strategies for the defender, one needs to consider $(z_1, \ldots, z_n) \in \Pi_{i=1}^n \{0, \ldots, b_i\}$ and determine whether it is a feasible pure strategy that meet all constraints in (13). This algorithm, however, quickly becomes computationally intractable as n and k increase. We therefore present an alternative algorithm in the next section.

4.1 Analysis of the bodyguard scheduling game

The first step of the algorithm is to solve the linear program of (9) by including the schedule constraints of (13) but relax the integrality constraints. Formally, we solve linear program:

$$
\begin{aligned}
&\min v \\
&\text{s.t. } v \geq (g_i(j+1) - g_i(j)) \cdot (x_i - j) + g_i(j) \ \forall j = 0, 1, \ldots, b_i - 1\,, \qquad \forall i \in N \\
&\quad v \geq g_i(b_i) \qquad \forall i \in N \\
&\quad \sum_{j \in N} w_{0j} = k \\
&\quad w_{0j} + \sum_{i \in N} w_{ij} = x_j \qquad \forall j \in N \\
&\quad \sum_{j \in N} w_{ij} \leq x_i \qquad \forall i \in N \\
&\quad w_{ij} \leq q_{ij} \qquad \forall i, j \in N \\
&\quad v \geq 0 \\
&\quad x_i \geq 0 \qquad \forall i \in N,
\end{aligned}
\tag{14}
$$

The solution to the linear program of (14) corresponds to the optimal expected number of bodyguards assigned to each target, assuming that the solution can be achieved by a consistent mixed strategy. In other words, if we can find a consistent mixed strategy that produces the expected number of bodyguards assigned to each target indicated by the solution to the linear program in (14), then that consistent mixed strategy is optimal for $\mathscr{G}_S$.

To search for such a consistent mixed strategy, we first identify candidate pure strategies. Based on the solution $(x_i)_{i \in N}$ to the linear program (14), we restrict our attention to those pure strategies that assign either $\lfloor x_i \rfloor$ or $\lceil x_i \rceil$ bodyguards to target $i \in N$. Consequently, the number of candidate pure strategies equals 2^n. The next step is to eliminate those pure strategies that do not meet the schedule/flow constraints in (13). The final step is to use linear programming again to solve the two-person zero-sum matrix game $\mathscr{G}_S^*$, in which the attacker can choose any of the n targets to attack but the defender can use only the feasible pure strategies just identified that assigns either $\lfloor x_i \rfloor$ or $\lceil x_i \rceil$ bodyguards to target $i \in N$. Because in $\mathscr{G}_S^*$ the defender's pure strategy set is a subset of that in $\mathscr{G}_S$, the value of $\mathscr{G}_S^*$ is an upper bound for the value of $\mathscr{G}_S$. In addition, the optimal value of the linear program in (14) is a lower bound for the value of $\mathscr{G}_S$ because the expected number of bodyguards assigned to each target may not be achieved by any defender's mixed strategy. Consequently, if the value of $\mathscr{G}_S^*$ coincides with the optimal value of the linear program in (14), then that common value is also the value of the game $\mathscr{G}_S$, and the defender's optimal mixed strategy for $\mathscr{G}_S^*$ is also optimal for $\mathscr{G}_S$. If the value of $\mathscr{G}_S^*$ is strictly higher than the value of the linear program in (14), then we find an upper bound for the value of $\mathscr{G}_S$ and the defender can achieve this upper bound by playing an optimal mixed strategy in $\mathscr{G}_S^*$. In order to assess this algorithm, we randomly generated 50,000 instances of $\mathscr{G}_S$ (see Appendix 6.1 for details). It turns out that among these 50,000 instances, the value of $\mathscr{G}_S^*$ is equal to the value of $\mathscr{G}_S$. Therefore, we make the following conjecture.

Conjecture 1. *The value of $\mathscr{G}_S^*$ is equal to the value of $\mathscr{G}_S$.*

We now demonstrate how the algorithm can be applied to the game of Example 5.

Example 6. *Reconsider the setting of Example 5. For step 1, we solve the standard linear programming problem in (14), leading to $x_1 = \frac{5}{6}, x_2 = \frac{5}{6}, x_3 = \frac{1}{6}$, $w_{12} = \frac{5}{6}, w_{01} = \frac{5}{6}, w_{02} = 0, w_{03} = \frac{1}{6}$, $w_{12} = w_{13} = w_{21} = w_{23} = w_{31} = w_{32} = 0$ with objective value $v = \frac{7}{15}$. Because the number of targets equals $n = 3$, we need to generate 2^3 pure strategies. That is, we consider strategies $(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1)$ and $(1,1,1)$. For step 3, we check each of them on feasibility. This leads to leaving our strategies $(1,0,1)$ and $(1,1,1)$. For the remaining 6 strategies, we execute step 4 and solve a new two-person zero-sum game, in which the defender's pure strategy set consists of $(0,0,0), (1,0,0), (0,1,0), (0,0,1), (1,1,0)$ and $(0,1,1)$. This leads to the same solution as we found in Example 5 already.*

In Figure 5, we summarize the steps of the algorithm to generate an upperbound of the game.

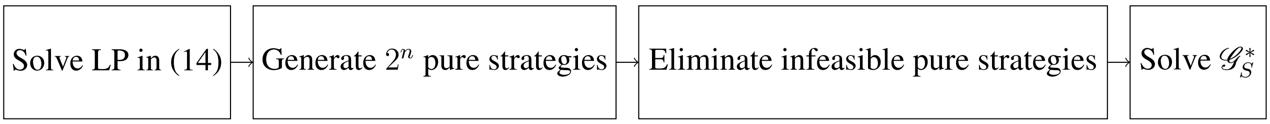

Figure 5: Steps of algorithm to generate an upper bound of the game

The significance of the the first step of the algorithm is that it removes a huge number of pure strategies that the defender does not need to consider. For example, if there are $n = 10$ targets and $k = 30$ bodyguards, then we need to screen only $2^{10} = 1024$ pure strategies on feasibility. Without this step, we need to screen up to $(30 + 1)^{10} \approx 8.19 \times 10^{14}$ pure strategies on feasibility.

We next discuss an example illustrating how the algorithm can be used in a realistic setting for a security agency during a morning shift.

Example 7. *Consider a setting with $n = 7$ targets, $k = 10$ bodyguards, $q_{12} = q_{15} = q_{23} = q_{45} = q_{65} = q_{67} = 10$ and $q_{ij} = 0$ otherwise. Moreover, we have $g_i(z) = \gamma_i \cdot \exp\{-\alpha_i \cdot z_i\}$ for all $i \in N$ with $(\gamma_i)_{i \in N} = (0.95, 0.90, 0.85, 0.80, 0.75, 0.70, 0.65)$ and $(\alpha)_{i \in N} = (0.5, 0.2, 0.3, 0.6, 0.4, 0.1, 0.2)$. Next, $b_i = 10$ for all $i \in N$. A visual representation of the setting is presented in Figure 6.*

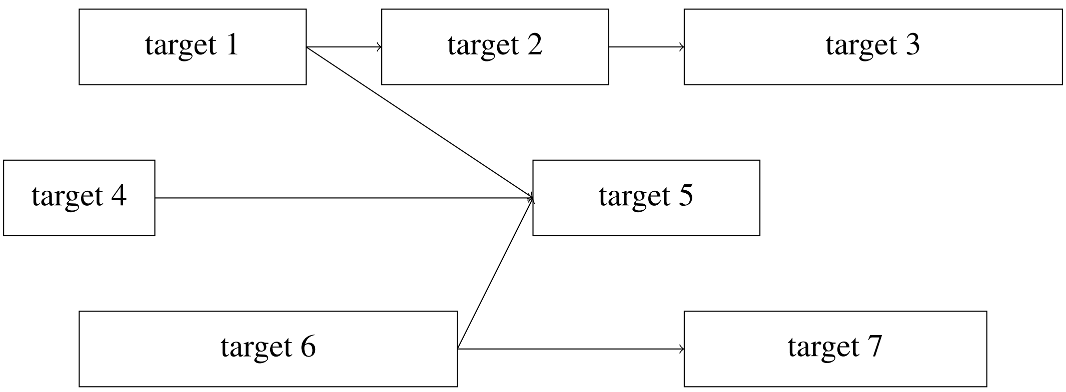

Figure 6: Visualisation of the setting with $n = 7$ targets.

The solution of the linear programming problem of (14) is presented in Table 2 and Table 3.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	v
3.8162	3.8162	2.3774	1.0918	1.4940	5.0920	2.1881	0.4209

Table 2: Solution $(x_i)_{i \in N}$ and v of LP.

w_{ij}	1	2	3	4	5	6	7
0	3.8162	0	0	1.0918	0	5.0920	0
1	0	3.8162	0	0	0	0	0
2	0	0	2.3774	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	1.4940	0	2.1881
7	0	0	0	0	0	0	0

Table 3: Solution $(w_{0j})_{j \in N}$ and $(w_{ij})_{i,j \in N}$ of LP.

Because the number of targets is equal to $n = 7$ targets, we need to generate $2^7 = 128$ pure strategies with $z_1 \in \{3,4\}$, $z_2 \in \{3,4\}$, $z_3 \in \{2,3\}$, $z_4 \in \{1,2\}$, $z_5 \in \{1,2\}$, $z_6 \in \{5,6\}$, and $z_7 \in \{2,3\}$. Forty of these pure strategies turn out to violate the schedule/flow constraints. Subsequently, we formulate the matrix game $\mathscr{G}_S^*$ by allowing the defender to use the remaining $128 - 40 = 88$ pure strategies. Solving $\mathscr{G}_S^*$—a matrix game of size 7×88—via linear programming, we obtain an optimal mixed strategy for the defender, as shown in Table 4.

probability	z_1	z_2	z_3	z_4	z_5	z_6	z_7
0.09198	3	3	2	1	2	6	3
0.09185	3	3	3	2	1	5	3
0.41411	3	4	3	1	1	5	2
0.40206	3	4	3	1	2	5	3

Table 4: The defender's optimal mixed strategy for $\mathscr{G}_S^*$ in Example 7.

The value of $\mathscr{G}_S^*$ equals 0.4209, which matches an optimal solution of the linear program of (14). Therefore, the defender's optimal mixed strategy for $\mathscr{G}_S^*$ in Table 4 is also optimal for $\mathscr{G}_S$.

One strength of the algorithm is that it is scalable in the number of bodyguards k, because its runtime is more or less constant in the number of bodyguards. The algorithm, however, is not scalable in the number of targets n. In the next section, we discuss specific structures of the schedule constraints that help reduce n, implying that we can solve $\mathscr{G}_S$ faster.

4.2 Special structures

In this section, we discuss three specific structures of the schedule constraints that can reduce the runtime to solve $\mathscr{G}_S$. For each of the three structures, we assume that $q_{ij} \in \{0, k\}$ for all $i, j \in N$.

4.2.1 Horizontal clusters

A *horizontal cluster* is a group of targets close in locations with no overlaps in time—but away from the other targets not in the cluster—such that a bodyguard assigned to the first target in the cluster can only be reassigned to the other targets in the same cluster. An example of two horizontal clusters is represented in Figure 7.

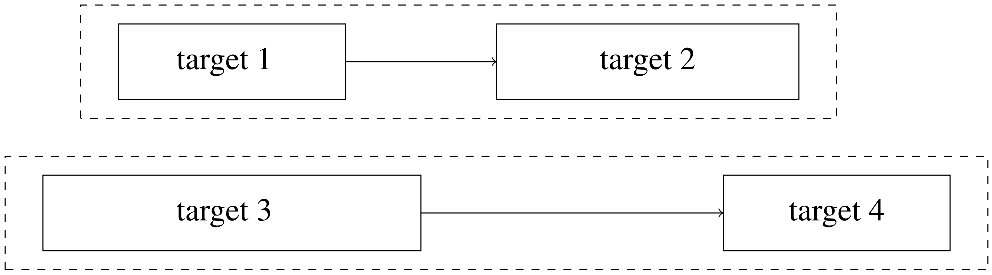

Figure 7: Visualisation of the setting with $n = 4$ targets and two clusters ($\{1, 2\}$ and $\{3, 4\}$).

The upper cluster could, for example, represent a candidate who is running for an election and is holding two political rallies in a remote town, one in the morning and one in the afternoon. For this setting, it is natural to assign the same team of bodyguards to both of these political rallies.

If all the targets can be partitioned into several horizontal clusters in a bodyguard scheduling game, then the bodyguard *scheduling* game reduces to a bodyguard *allocation* game, because we can merge all targets in a cluster into one single target whose damage function corresponds to the maximum of the damage functions of all targets within the cluster. If only a subset of targets can be put into horizontal clusters, then we can still use one damage function to represent each horizontal cluster, which effectively reduces the number of *targets* in a bodyguard scheduling game.

4.2.2 Vertical clusters

A *vertical cluster* is a group of targets that take place around the same time, with all the other targets taking place either before or after them, so that all k bodyguards are available and shared by the targets in the vertical cluster. Figure 8 displays a bodyguard scheduling game with two vertical clusters.

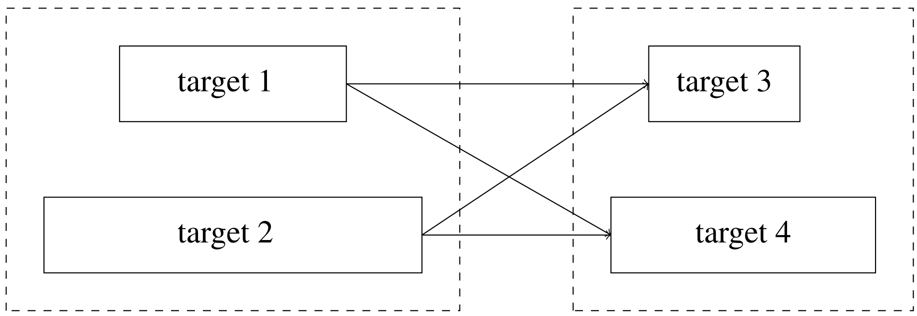

Figure 8: Visualization of the setting with $n = 4$ targets and two groups ($\{1, 2\}$ and $\{3, 4\}$).

The first vertical cluster could, for instance, represent two members of the Royal family each visiting one city in the early morning, while the second cluster could represent two cabinet ministers, each visiting a university in the late afternoon. Because there is plenty of time between the morning and afternoon activities, there is also enough time for bodyguards to travel between them.

If all the targets can be partitioned into several vertical clusters in a bodyguard scheduling game, then solving the bodyguard scheduling game reduces to solving a bodyguard allocation game for every vertical cluster. If only a subset of targets can be put into vertical clusters, then we can still solve each of these vertical clusters via a bodyguard allocation game separately, which effectively reduces the number of *targets* of the original bodyguard scheduling game.

4.2.3 Diverging cluster

A *diverging cluster* is a group of targets that form a tree in the schedule network. Figure 9 displays an example of a diverging cluster that consists of targets 3, 4 and 5.

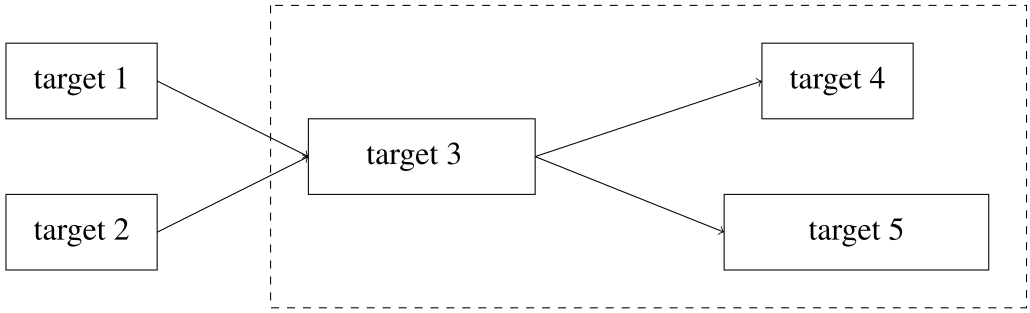

Figure 9: Visualization of the setting with $n = 5$ targets and one tree ($\{3, 4, 5\}$).

The diverging cluster as shown in Figure 9 could, for instance, represent three cabinet members each having meetings close in location, but at different times. One of the cabinet members has a meeting shortly after lunch, followed by two meetings—in parallel—for the other two members.

If a group of targets can be identified as a diverging cluster in a bodyguard scheduling game, then there is no need to execute steps 2, 3 and 4 of the algorithm of the bodyguard scheduling game for all targets. We demonstrate this by the example in Figure 9. First, according to step 1 of the algorithm, we solve the linear program in (14) for all targets.

Thereafter, we execute the remaining steps for only target 1, 2, and 3. Instead of checking 2^5 strategies on feasibility, we thus only check 2^3 of them. We ignore targets 4 and 5 in the remaining steps, because we can construct a consistent mixed strategy for target 4 and 5, based on the solution of the linear program in (14).

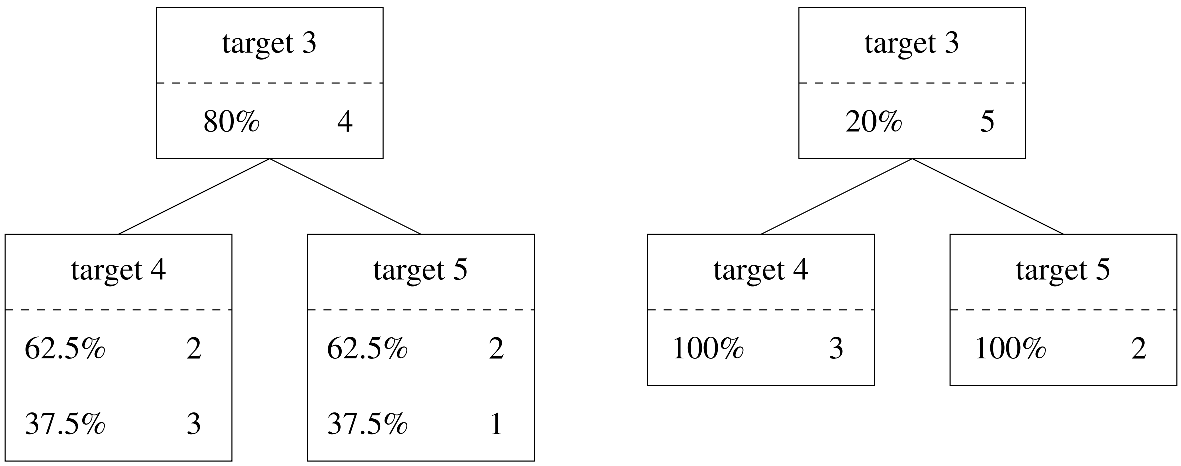

Figure 10: Visualization of how to assign bodyguards to target 4 and 5.

To illustrate this idea, suppose that the LP generates solution $x_3 = 4.2$, $x_4 = 2.5$, and $x_5 = 1.7$. For any consistent mixed strategy, we know that in 80% of the time we assign 4 bodyguards to target 3 and in 20% of the time we assign 5 bodyguards to target 3. If we assign 4 bodyguards to target 3, we assign 2 of them with probability 62.5% to target 4 and we assign 3 of them with probability 37.5% to target 4. If we assign 5 bodyguards to target 3, we always assign 3 bodyguards to target 4. In summary, we assign in $62.5\% \cdot 80\% = 50\%$ of the time 2 bodyguards and in $37.5\% \cdot 80\% + 20\%$ = 50% of the time 3 bodyguards, which matches with x_4. Consequently, we also know that in $62.5\% \cdot 80\% = 50\%$ of the time we assign $(4 - 2) = 2$ bodyguards to target 4 and in $37.5\% \cdot 80\% =$

30% of the time, we assign $(4-3) = 1$ bodyguard to target 4. Finally, in 20% of the time, we assign $(5-3) = 2$ bodyguards to target 4. In summary, we assign in $62.5\% \cdot 80\% + 20\% = 70\%$ of the time 2 bodyguards to target 4 and in $37.5\% \cdot 80\%$ = 30% of the time 1 bodyguard to target 5, which matches with x_5. Hence, we have constructed a consistent mixed strategy for target 4, and 5. We also visualized this procedure in Figure 10.

If target 4 would have two additional children, in the form of target 6 and 7 (see Figure 11), then we construct a consistent mixed strategy by first considering target 4, 6 and 7 as one consolidated node—and follow the procedure described above, with target 3 as the parent node and target 5 and the consolidated node as the children nodes. Subsequently, we repeat our procedure, but this time for the tree with target 4 as parent node and target 6 and target 7 as children nodes.

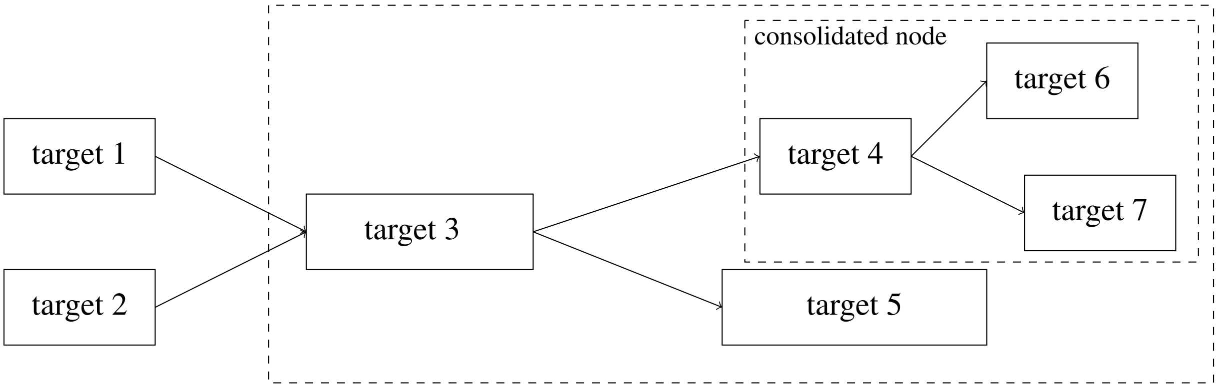

Figure 11: Visualization of the setting with $n = 7$ targets and a tree ($\{3, 4, 5, 6, 7\}$).

5 Conclusion

We have investigated a resource allocation problem where a limited supply of bodyguards are to be allocated for protecting individuals under threat. We model the problem as a two-person zero-sum game between a defender who allocates the bodyguards and an attacker who chooses one target to attack. Because the number of feasible bodyguard allocations grows quickly as either the number of protected targets or the number of bodyguards increase, solving the game by brute force with a linear program becomes computationally intractable for problems of practical size. By assuming that the marginal effectiveness of each additional bodyguard assigned to a target is non-increasing, we show that we can solve the game with a different linear program whose size is linear in the numbers of both targets and bodyguards. In doing so, we exploited that the best way to implement a defender's mixed strategy—with an expected number $x \in \mathbb{R}_+$ of bodyguards allocated to a target—is to allocate either $\lfloor x \rfloor$ bodyguards or $\lceil x \rceil$ bodyguards with appropriate probabilities. Next, we extended the allocation game to a scheduling game, which allows a bodyguard to report to multiple targets if their schedules allow. We developed an algorithm to compute a bound for the value of this game and presented a mixed strategy that achieved this bound in all numerical experiments we have conducted. Finally, we discussed several special structures of our bodyguard game that can be solved efficiently. These special structures may arise in large bodyguard scheduling problems in practice.

The results in this paper are established under some assumptions. First, we assume that the damage function is convex and non-increasing for each additional bodyguard assigned to protect a target. This assumption appears reasonable in many situations and is supported by the study of Nachtegael (2024) who interviewed various Dutch bodyguards. However, there may be other situations where this assumption does not apply, and how to leverage the specific property of the damage function to solve such a game efficiently would be a practical research problem. In our paper, we also made the assumption that bodyguards are interchangeable. That is, every bodyguard is equally effective when assigned to protect different targets. In reality, however, there are situations where the protection effectiveness depends on which bodyguards are assigned to protect a certain target – not just how many bodyguards are assigned to the task. Treating such a variation requires an entirely different formulation, which considers variability among bodyguards' qualifications and targets' vulnerabilities. As a final future research direction, one could either prove Conjecture 1 or find a counterexample.

Acknowledgement

This work was supported by NWO grant VI.Veni.201E.017.

References

Adler, N., Hakkert, A. S., Kornbluth, J., Raviv, T., and Sher, M. (2014). Location-allocation models for traffic police patrol vehicles on an interurban network. *Annals of operations research*, 221(1):9–31.

Azaiez, M. N. and Bier, V. M. (2007). Optimal resource allocation for security in reliability systems. *European Journal of Operational Research*, 181(2):773–786.

Baron, O., Berman, O., and Gavious, A. (2018). A game between a terrorist and a passive defender. *Production and Operations Management*, 27(3):433–457.

Bier, V., Oliveros, S., and Samuelson, L. (2007). Choosing what to protect: Strategic defensive allocation against an unknown attacker. *Journal of Public Economic Theory*, 9(4):563–587.

Bier, V. M., Haphuriwat, N., Menoyo, J., Zimmerman, R., and Culpen, A. M. (2008). Optimal resource allocation for defense of targets based on differing measures of attractiveness. *Risk Analysis: An International Journal*, 28(3):763–770.

Camacho-Collados, M. and Liberatore, F. (2015). A decision support system for predictive police patrolling. *Decision support systems*, 75:25–37.

Chaiken, J. M. and Dormont, P. (1978). A patrol car allocation model: Capabilities and algorithms. *Management science*, 24(12):1291–1300.

Curtin, K. M., Hayslett-McCall, K., and Qiu, F. (2010). Determining optimal police patrol areas with maximal covering and backup covering location models. *Networks and spatial economics*, 10(1):125–145.

Dahan, M., Sela, L., and Amin, S. (2022). Network inspection for detecting strategic attacks. *Operations Research*, 70(2):1008–1024.

Dunnett, S., Leigh, J., and Jackson, L. (2019). Optimising police dispatch for incident response in real time. *Journal of the Operational Research Society*, 70(2):269–279.

Green, L. (1984). A multiple dispatch queueing model of police patrol operations. *Management science*, 30(6):653–664.

Green, L. and Kolesar, P. (1984). The feasibility of one-officer patrol in new york city. *Management science*, 30(8):964–981.

Green, L. V. and Kolesar, P. J. (2004). Anniversary article: Improving emergency responsiveness with management science. *Management Science*, 50(8):1001–1014.

Guan, P., He, M., Zhuang, J., and Hora, S. C. (2017). Modeling a multitarget attacker–defender game with budget constraints. *Decision Analysis*, 14(2):87–107.

Gupta, S., Starr, M. K., Zanjirani Farahani, R., and Ghodsi, M. M. (2020). Prevention of terrorism–an assessment of prior pom work and future potentials. *Production and Operations Management*, 29(7):1789–1815.

Hausken, K. (2008). Strategic defense and attack for series and parallel reliability systems. *European Journal of Operational Research*, 186(2):856–881.

Hausken, K. (2024). 50 years of operations research in defense–a european perspective. *European Journal of Operational Research*.

Hunt, K. and Zhuang, J. (2024). A review of attacker-defender games: Current state and paths forward. *European Journal of Operational Research*, 313(2):401–417.

Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Ordónez, F., and Tambe, M. (2012). Software assistants for randomized patrol planning for the LAX airport police and the federal air marshals service. *Interfaces: Operations Research Management Science*, 52(1):133.

Kaag, S. (2023). Voorjaarsnota 2023. *Tweede Kamer, Nederland.*

Kolesar, P. J. and Green, L. V. (1998). Insights on service system design from a normal approximation to erlang's delay formula. *Production and Operations Management*, 7(3):282–293.

Kolesar, P. J., Rider, K. L., Crabill, T. B., and Walker, W. E. (1975). A queuing-linear programming approach to scheduling police patrol cars. *Operations Research*, 23(6):1045–1062.

Lidbetter, T. and Lin, K. Y. (2020). A search game on a hypergraph with booby traps. *Theoretical Computer Science*, 821:57–70.

Musegaas, M., Schlicher, L., and Blok, H. (2022). Stackelberg production-protection games: Defending crop production against intentional attacks. *European Journal of Operational Research*, 297(1):102–119.

Nachtegael, M. (2024). Risk-based guard allocation for threatened individuals. *Master thesis, Operations Management and Logistics, Eindhoven University of Technology*, pages 1–50.

Pita, J., Jain, M., Ordónez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., and Kraus, S. (2009). Using game theory for Los Angeles airport security. *AI magazine*, 30(1):43–43.

Ross, S. M. and Lin, K. Y. (2001). Applying variance reduction ideas in queuing simulations. *Probability in the Engineering and Informational Sciences*, 15(4):481–494.

Schaack, C. and Larson, R. C. (1989). An n server cutoff priority queue where arriving customers request a random number of servers. *Management Science*, 35(5):614–634.

Schlicher, L. and Lurkin, V. (2024). Fighting pickpocketing using a choice-based resource allocation model. *European Journal of Operational Research*, 315(2):580–596.

Shan, X. and Zhuang, J. (2013). Hybrid defensive resource allocations in the face of partially strategic attackers in a sequential defender–attacker game. *European Journal of Operational Research*, 228(1):262–272.

Start, R. (2023). Stelsel beveiliging bedreigde personen dreigt te bezwijken. *NOS*.

Subelman, E. J. (1981). A hide-search game. *Journal of Applied Probability*, 18(3):628–640.

Wu, C.-K., Chen, Y.-M., Wu, D., and Chi, C.-L. (2020). A game theory approach for assessment of risk and deployment of police patrols in response to criminal activity in san francisco. *Risk analysis*, 40(3):534–549.

Zhuang, J. and Bier, V. M. (2007). Balancing terrorism and natural disasters—defensive strategy with endogenous attacker effort. *Operations Research*, 55(5):976–991.

Zouridis, S. (2023). Bewaken en beveiligen lessen uit drie beveiligingssituaties. *Onderzoeksraad voor Veiligheid.*

6 Appendix

6.1 Experiment for the upper bound for bodyguard scheduling games

In this experiment, we generated 50,000 random bodyguard scheduling games. For each of them, we calculated the value of $\mathscr{G}_S$ and $\mathscr{G}_S^*$. It turns out that $\mathscr{G}_S$ and $\mathscr{G}_S^*$ coincide for all instances.

The 50,000 random bodyguard scheduling games are generated as follows. For each number of targets $n \in \{5, 6, 7, 8, 9\}$ and each number of bodyguards $k \in \{10, 11, ..., 19\}$, we generated 1,000 random bodyguard games. In doing so, we generated/set:

- $t_i^s \sim$ Uniform$[0, 10]$ and $t_i^e \sim t_i^s +$ Uniform$[0, 10]$ for all $i \in N$
- travel times $t_{ij} \sim$ Uniform$[0, 2]$ for all $i, j \in N$. Note, these travel times are not part of our bodyguard scheduling game, but we used them to identify q_{ij} for all $i, j \in N$.
- $q_{ij} = k$ if $t_i^e + t_{ij} \leq t_j^s$ and $q_{ij} = 0$ otherwise.
- $g_i(z) = \gamma_i \cdot \alpha_i^z$ with $\gamma_i \sim$ Uniform$[0, 1]$, $\alpha_i \in$ Uniform$[0, 1]$ and $b_i = k$ for all $i \in N$.

Function g is commonly used in the homeland security literature to model diminishing returns (see, e.g., Bier et al. (2007)) and also has an operational interpretation as already discussed in Section 3.5.2. For instance, one way to interpret this damage function is that each additional bodyguard adds a defense layer for the target. The attacker penetrates each defense layer of target $i \in N$ with probability α_i, independently of everything else, and succeeds in the attack only if the attacker penetrates all defense layers. In that regard, α_i could be interpreted as the effectiveness of each additional bodyguard for a given target $i \in N$. Moreover, value γ_i could be interpreted as the importance/societal value of target $i \in N$.