mathematics

MDPI

Article

Signaling Security Games with Attack Planner Deception

Santing He [1,*], Mingchu Li [2] and Runfa Zhang [3]

[1] School of Mathematical Sciences, Dalian University of Technology, Dalian 116000, China
[2] School of Software Technology, Dalian University of Technology, Dalian 116000, China; mingchul@dlut.edu.cn
[3] School of Automation and Software Engineering, Shanxi University, Taiyuan 030013, China; zrf@sxu.edu.cn
* Correspondence: hesanting@mail.dlut.edu.cn

Abstract: This paper studies a class of attack behavior in which adversaries assume the role of initiators, orchestrating and implementing attacks by hiring executors. We examine the dynamics of strategic attacks, modeling the initiator as an attack planner and constructing the interaction with the defender within a defender–attack planner framework. The individuals tasked with executing the attacks are identified as attackers. To ensure the attackers' adherence to the planner's directives, we concurrently consider the interests of each attacker by formulating a multi-objective problem. Furthermore, acknowledging the information asymmetry where defenders have incomplete knowledge of the planners' payments and the attackers' profiles, and recognizing the planner's potential to exploit this for strategic deception, we develop a defender–attack planner model with deception based on signaling games. Subsequently, through the analysis of the interaction between the defender and planner, we refine the model into a tri-level programming problem. To address this, we introduce an effective decomposition algorithm leveraging genetic algorithms. Ultimately, our numerical experiments substantiate that the attack planner's deceptive strategy indeed yield greater benefits.

Keywords: signaling game; attack planner deception; multi-attackers; multi-objective optimization; tri-level programming; decomposed algorithm; genetic algorithm

MSC: 91A80

Citation: He, S.; Li, M.; Zhang, R. Signaling Security Games with Attack Planner Deception. *Mathematics* **2024**, *12*, 2532. https://doi.org/10.3390/math12162532

Academic Editor: Antanas Cenys

Received: 8 July 2024
Revised: 9 August 2024
Accepted: 14 August 2024
Published: 16 August 2024

1. Introduction

Game theory has found extensive applications in various fields, especially in the Stackelberg games (SGs), which are simple yet powerful models for sequential interaction between two strategic players: a leader and a follower. Specifically, the leader commits to a strategy first, and the follower responds after observing the leader's commitment. The game was first proposed in 1934, when researchers introduced it to analyze market competition between a large-leader firm and a small-follower firm [1]. Since then, it has been applied to a diverse array of problems, including principal–agent contract design [2], pricing and planning in transportation systems [3], influence maximization [4] and exam design [5], often focusing on the leader's perspective and with the aim of identifying the optimal strategy for the leader to commit to. Over recent years, with the successful application of SGs in security resource allocation, an increasing number of experts have utilized them to address security problems, known as the Stackelberg Security Games (SSGs) [6]; thus, SSGs have emerged as a significant area of research for SGs. To date, SSGs have been applied to address a multitude of practical issues, such as threat screening games [7], green security games [8–10] and crime prevention strategies [11]. In SSGs, the defender, as the leader, implements an optimal mixed (or randomized) defense strategy. This mixed strategy represents a probability distribution across all the possible defense strategies. The attackers, as the followers, observe this and adjust their strategy to optimize their outcome [12]. This has been seen in, for instance, the Los Angeles Sheriff's Department's application to the

subway system [13] and the US Coast Guard's initiatives on ports and waterways in Boston and New York City [14], among others. In these applications, a defender aims to protect targets from strategic adversaries by deploying limited resources. The central solution concept in SSGs is the strong Stackelberg equilibrium (SSE) [15,16], which determines the defender's optimal strategy. However, SSE is predicated on the assumption that both parties have complete knowledge of each other's information. In numerous real-world scenarios, this assumption often fails to hold true. The defender frequently lacks complete information about the follower. Currently, these issues have been extensively researched, resulting in a multitude of proposed solutions. For example, when the defender is uncertain about the attacker's payoff but is aware of its range, strategies are explored that are robust to small interval uncertainties [17–19]. In scenarios where the defender is only aware of the distribution of the attacker's payoff, the Bayesian Stackelberg equilibrium is used to maximize the defender's utility [20–22]. Furthermore, as defenders gather information through interactions with attackers, numerous studies have investigated learning from the attackers' response strategies to refine leadership tactics [17,23–26]. The essence of these approaches is the defender's attempt to acquire relevant information about the attacker, assuming the information obtained is accurate. However, when the attacker becomes aware of the defender's reliance on information, there is a tendency for the attacker to use tactics that obfuscate the defender, intending to provide them with misleading information, which is a form of deceitful behavior.

Currently, the attackers' deception has attracted significant attention, prompting a variety of studies to focus on this phenomenon. Existing research mainly investigates scenarios where attackers provide false information about a single aspect, such as their payoff [27–29] or the degree of rationality [30–32]. However, in numerous real-world scenarios, defenders confront a more intricate challenge where attackers have multiple pieces of undisclosed information. Recognizing this gap, broadening the scope of analysis to include situations with followers possessing multiple private information sets is essential. By considering a scenario in which the attacker has two types of private information, we can establish a more comprehensive model. In the existing body of research, the attacker commonly uses strategies that involve either selecting a target to attack [33] or allocating resources for an attack on the targets [34,35]. However, real-world scenarios frequently exhibit a more complex dynamic where the attacker assumes the role of a "leader" who orchestrates an attack and recruits individuals to execute it. In this context, the individual who devises the attack strategy is referred to as the "attack planner", while the person commissioned to execute the attack is termed the "attacker". Unlike resources, which can be allocated at will, individuals have independent thoughts and agency. Therefore, it is essential to consider their interests to ensure their adherence to the arranged strategy. Furthermore, the defender is often unaware of the specifics regarding the hired individuals, which we establish as the attack planner's private information. This can be exploited by the attack planner to mislead defenders by misrepresenting the number of hired individuals.

In summary, this paper introduces a model that delineates the interaction between the attack planner and the defender, then extends this model to account for the attack planner's deception and the compliance of hired individuals through a multi-objective problem. Subsequently, we propose a tri-level programming framework that aligns with the interactive process. The first level involves signaling, wherein the attack planner seeks to obscure the defender's comprehension and maximize personal benefits. The second level pertains defense, wherein the defender strategically allocates resources to minimize potential losses. The third level constitutes the attack phase, wherein the attack planner coordinates the execution of the attack, aiming to maximize profit. Simultaneously, to ensure the attackers' adherence with the plan, the attack strategy formulated by the attack planner must safeguard the attackers' interests. This model deviates from the conventional tri-level min–max–min programming models [36,37], rendering traditional solving algorithms inapplicable. Consequently, we introduce an efficient, customized algorithm

designed to accommodate the unique structure and requirements of the model, ensuring effective resolution and thorough analysis.

Our contributions: In order to deal with the opponent's deception effectively, this paper analyzes a kind of deception behavior in attack in detail.

- We model the attack initiator as the attack planner—the individual they hire to carry out the attack is the attacker—and we propose a defender–attack planner (D-AP) model to describe the interaction between them;
- We account for the interests of each attacker to ensure compliance, framing this as a multi-objective optimization issue;
- This paper also addresses the defender's uncertainty to the attack planner's payment and the particulars of the hired attackers, constructing a defender–attack planner game model with deception (D-APD).
- We articulate D-APD as a tri-level programming problem and propose a customized decomposition algorithm based on a genetic algorithm (DA-GA) to address it.

The remainder of this paper consists of the following. In Section 2, we summarize the previous related work. In Section 3, we present detailed descriptions of the D-AP and D-APD models. In Section 4, we formulate a tri-level programming problem and propose a solution algorithm. In Section 5, we a conduct further analysis of the model via experimental validation. Finally, Section 6 presents the conclusions of this paper and outlines directions for future research.

2. Related Work

This section provides an overview of the existing literature that is pertinent to our research, focusing on two key aspects: deception behavior within games and the role of the planner.

2.1. Deception in Games

As deception can obfuscate an adversary's understanding and disrupt their decision-making process, certain research endeavors have explored methods to mitigate or neutralize criminal offensives through deceptive tactics. Within the homeland security domain, studies have formulated models of deceptive strategies wherein the defender employs a range of signals to mislead the attacker [38–40]. Ref. [33] examines security games involving the defender's use of disguised resources. Ref. [41] explores the concept of strategic secrecy in the defender's approach to infrastructure protection. In addition, when the attacker is unsure of the amount of the defender's resources, the defender may deceive the attacker by disguising or hiding a portion of the resources, potentially reporting fewer resources than are actually available [33]. In contrast to [33], Ref. [42] studied the phenomenon of bluffing in signaling games, where the defender may claim to possess more resources than are actually available, serving as a deterrent to the attacker and thus preventing the attack. In [43], a problem with multiple attacker types is studied, in which the defender initially commits to a mixed strategy and signaling scheme, identifies the attacker's type and then acts, and the attacker subsequently responds. In [44], a two-stage game model is presented, in which the first stage is a basic defender–attacker game, and in the second stage, the defender sends a deceptive signal regarding actual resource allocation. Ref. [40] explores the issue of deception in a single-objective multi-period game where multiple types of defenders exist. In each game, they initially select a behavior and a signaling strategy. Another work has discussed the problem of stochastic games with multiple defenders cheating [45]. In [34], researchers used the hypergame framework to study the deception of defenders in a game with multiple attackers. Attackers with multiple types possess distinct payoff functions and resource quantities, so the hypergame framework is used to model each attacker individually. Ref. [46] examines the optimal conditions and methods for defenders to effectively deceive various attackers. Other studies investigated the deception of a defender in a game with ambiguous benefits and multiple attackers [47]. Ref. [48] examines how defenders counter advanced sustainability threats through deceptive behavior.

However, precisely because the deceptive behavior confuses the adversary, more and more attackers will choose to deceive, leading to a deviation in defense strategy and an infliction of more severe damage. In [27], with distinct types characterized by varying payoffs, the attacker deceives the defender by imitating other attacker types. In [28], the attacker deceives the defender by directly presenting a false type. Nguyen et al. studied the deception strategies of attackers in repeated games [29]. Ref. [49] addresses similar problems. Unlike Ref. [29], this study examines the issues within the SSG framework, whereas Ref. [29] focuses on a stochastic one. Ref. [30] studies deception in games with multiple attackers, where multiple attackers pay the same but have different levels of rationality, and perfectly rational attackers adopt a guise of bounded rationality to deceive defenders. However, these studies do not consider the scenario where attackers possess multiple pieces of private information.

In addition, because asymmetric information is a key friction in many economic interactions, deception often appears in the principal–agent problem [2,50,51], which is a prevalent application of the SGs. The principal delegates the management of the output process to the agent. A contract is signed in advance, specifying the terms of an incentive payment. The agent exerts costly effort for managing the output. Subsequently, given the contract offered by the principal, the agent returns an optimal effort response that optimally balances their cost of effort with the proposed compensation. Ultimately, the principal selects the optimal contract to motivate the agent's effort [52]. Then, a principal may offer a range of contracts to an agent with uncertain characteristics and ask the agent to choose the one that matches their true type. Naturally, the principal's menu must consider the agent's potential misrepresentation of their type, meaning that the agent may mislead the principal [53]. However, such a deception is predominantly observed in economic contexts and significantly differs from the deception encountered in the security domain we examine.

2.2. About Planner

Several studies have examined the beneficial impacts of community participation in wildlife preservation [54–58], but these studies lack a mathematical model of the interaction between defenders and attackers; some researchers formulated it using the Stackelberg game framework [59]. Acknowledging the possibility of community members colluding with the attacker, the patrol, acting as the defender, employs community informants to gather intelligence on the attacker's activities. However, no existing studies have yet considered a scenario where attackers may also hire some people to execute the attack, which significantly diverges from the defender's recruitment of community members, including the consideration of the interests of the hired attackers. It is in addressing this gap that this paper contributes to the literature.

3. Model

In this section, we initially construct a foundational defender–attack planner game model. Subsequently, considering the attack planner's private information, we construct a defender–attack planner with a deception model based on signaling games. Table 1 describes the mathematical notation used in this paper.

Table 1. Legend of common symbols.

Symbol	Description
Sets:	
SD	The set of defender's pure strategy
SA	The set of attack planner's pure strategy
Θ	The set of attack planner's payment types
Φ	The set of attacker types
S^1	The set of signals about payment type

Table 1. *Cont.*

Symbol	Description
S^2	The set of signal about attacker type
TY	The set of attack planner's types
S	The set of signals
Δ_{o1}	The set of mixed signaling strategy about planner's payoff type
Δ_{o2}	The set of mixed signaling strategy about planner's attacker type
Parameters:	
N	The number of targets
bs	Attacker's base salary vector
c	Defender's cost vector
B	Defender's budget constraint
n	The number of strong and weak attacker
v^d	Defender's valuation of targets
v^a	Attack planner's valuation of targets
sd	Defender's pure strategy
sa	Attack planner's pure strategy
q_j	The probability that the target j is attacked successfully
u^i	i-th attacker's base salary
ac_{ij}	The i-th attacker's cost when they attacks target j
α	The ratio of the commission when attack is successful
p^{pt}	The prior probability distribution about payment types
v^θ	The target valuation of Attack planner with payment type θ
p^θ	The prior probability distribution about attacker type
μ^{pt}	The posterior probability about the payment types
μ^{an}	The posterior probability about the attacker type
dc	Attack planner's deception cost
p^{ty}	The prior probability distribution of attack planner's type
$\mu(ty\|s)$	The probability that the attack planner's type is ty after receiving the signal s
L^D, U^{AP}, U^i	The utility of defender, planner and the i-th attacker without deception
L^{to}, U^{to}, U_i^{ty}	The utility of defender, planner and the i-th attacker with deception
Decision variables:	
π_o	The signaling strategy
π_d	Defender's strategy scheme
π_a	Attack planner's attack scheme

3.1. Defender–Attack Planner Game Model

In this section, we develop a model that captures the strategic interaction between a defender and an attack planner within the framework of a Stackelberg game, known as the Defender–Attack Planner (D-AP) model.

3.1.1. Model Description

A D-AP game model consists of a defender and an attack planner (hereafter referred to as the planner), where the defender needs to protect a set of N targets, and the planner hires some attackers to attack a subset of targets. The hired attackers are categorized into two types—strong and weak—based on their offensive capabilities (the model is also extensible to scenarios involving multiple attacker types). Their basic salary is different, which is denoted as $bs = (bs_1, bs_2)$; bs_1 and bs_2 correspond to the base salaries of the strong and weak attackers, respectively. The defender also possesses two defense levels: the high-level defense is capable of repelling all attackers, while the low-level defense is effective only against weak attackers. The defender incurs a fixed cost upon protecting a target; $c = (c_1, c_2, c_3)$ represents the cost vector, where $c_1 = 0$ signifies the absence of defense and c_2 and c_3 correspond to the costs of low-level and high-level defense, respectively. The defender has a budget constraint B and the planner also faces budgetary constraints,

which restricts the number of attackers they can hire; this is denoted as $n = (n_1, n_2)$, where planner-hired attackers are n, including n_1 strong attackers and n_2 weak attackers. $at = (at_1, \ldots, at_n)$ denotes each attacker's type (strong or weak), where $at_i \in \{1, 2\}$ and $at_i = 1(2)$ represent that the i-th attacker is weak (strong). Each target holds a distinct importance to the defender and the planner, that is, they have different valuations of the targets. The defender's valuation of the targets is $v^d = (v_1^d, \ldots, v_N^d)$ and the planner's valuation is $v^a = (v_1^a, \ldots, v_N^a)$. When a target j is attacked successfully, the defender incurs a penalty P_j^D and the planner receives a reward R_j^A. On the contrary, if an attack on a target j is unsuccessful, the defender will gain a reward R_j^D and the planner will incur a penalty P_j^A, and they satisfy $R_j^D > P_j^D$, $R_j^A > P_j^A$. This paper assumes that the utility of both sides is only related to the targets that were successfully attacked; in this case, $R_j^D = P_j^A = 0$, $P_j^D = -v_j^d$ and $R_j^A = v_j^a$.

3.1.2. Strategy Expression

The defender's pure strategy is to decide what level of protection to apply to each target within budget constraints which is denoted by $sd = (sd_1, \ldots, sd_N)$, where $sd_j \in \{0, 1, 2\}$ and $\sum_{j=1}^{N} c_{sd_j+1} \leq B$, $sd_j = 0$ indicate that the target j is not protected; $sd_j = 1$ means low-level defense against the target $j$and $sd_j = 2$ denotes high-level defense against the target j. $SD = \{sd | sd_j \in \{0, 1, 2\}, \sum_{j=1}^{N} c_{sd_j+1} \leq B\}$ represents the set of all the defender's pure strategies. The planner's pure strategy involves determining the allocation of the hired attackers to execute an attack on a designated set of targets, represented by $sa = (sa_{ij})_{i\in[n], j\in[N]}$, where $sa_{ij} \in \{0, 1\}$ and $sa_{ij} = 1(0)$ mean that the i-th attacker is (not) arranged to attack the target j. Following a successful attack on a target, the attacker responsible is granted a commission proportional to the target's value to the planner. Although it may appear that one attacker is assigned per target and vice versa, the diverse values of the targets result in a scenario of contention among multiple attackers, each seeking to secure a higher commission. Acknowledging this possibility, we assume that different attackers are permitted to attack the same target. However, the commission received from a successful attack will be evenly divided among the participating attackers, and the probability of a successful attack on a target remains unaffected by an increased number of attackers. This assumption facilitates adherence to the planner's strategy among attackers. $SA = \{sa_{n\times N} | sa_{ij} \in \{0, 1\}, \sum_{j=1}^{N} sa_{ij} = 1, i = 1, \ldots, n\}$ denotes the planner's strategy space, where $\sum_{j=1}^{N} sa_{ij} = 1$ means that each attacker can only attack one target.

3.1.3. Utility Function

Given the attack strategy sa, we let $sa^1 = (sa_{ij}^1)_{i\in[n], j\in[N]}$, where $sa_{ij}^1 = 0$ if $sa_{ij} = 0$ and $sa_{ij}^1 = at_i$ if $sa_{ij} = 1$; then, $sa_j^2 = max\{sa_{ij}^1, i = 1, \ldots, n\}$ signifies the magnitude of the attack that target j is projected to encounter, denoting $sa^2 = (sa_1^2, \ldots, sa_N^2)$. Specifically, should the planner allocate only weak attackers to a target, then the target faces only weak attacks; conversely, the target endures strong attacks. Then, for a strategy profile (sd, sa), the probability that the target j is successfully attacked is

$$q_j(sd, sa) = \begin{cases} 1, \ sa_j^2 > sd_j \\ 0, \ otherwise, \end{cases}$$

this is because the high-level defense is capable of repelling all attackers, while the low-level defense is effective only against weak attackers. In addition, we define $sa_j^3 = \sum_{i=1}^{n} sa_{ij}$ as the number of attackers attacking the target j, and let $sa^3 = (sa_1^3, \ldots, sa_N^3)$.

The total loss of the defender, which is denoted as L^D, is the sum of the value of the targets successfully attacked and the cost of the defensive strategy; then, for a strategy profile (sd, sa), we have

$$L^D(sd, sa) = \sum_{j=1}^{N} (q_j(sd, sa)v_j^d + c_{sd_j+1}). \tag{1}$$

The utility of the planner, U^{AP}, is the aggregate value derived from successfully attacked targets, minus the cost of hiring all attackers, then

$$U^{AP}(sd, sa) = \sum_{j=1}^{N} q_j(sd, sa)v_j^a - \sum_{i=1}^{n} U^i(sd, sa), \tag{2}$$

where U^i is the benefit of the i-th employed attacker, i.e., the cost of hiring the i-th attacker. For each hired attacker, irrespective of the attack's success, they receive a base salary, incur a certain attack cost, and upon a successful attack, they receive an additional commission proportional to the target's value. Thus

$$U^i(sd, sa) = u_i + \sum_{j=1}^{N} sa_{ij}q_j(sd, sa) \cdot \frac{\alpha v_j^a}{sa_j^3} - \sum_{j=1}^{N} sa_{ij}ac_{ij}, \tag{3}$$

where $u_i = bs_{at_i}$ is the i-th attacker's base salary, $\alpha \in (0, 1)$ denotes the proportionality constant of the commission awarded to the attacker post-successful attack and ac_{ij} denotes the i-th attacker's cost when they attacks target j. Attackers, despite having identical capabilities, may demonstrate diverse target preferences, influenced by a multitude of factors, such as the distance between the attacker and the target. This study incorporates these individual differences among attackers by representing these differences through variable attack costs ac.

3.2. Defender–Attack Planner Game Model with Deception

Building upon the previously outlined model, we develop an advanced Defender–Attack Planner game model with Deception (D-APD). This model takes into account the planner's strategic deception in two critical dimensions: the payoff type and the attacker type. The deception in payoff type could lead the defender to misestimate the importance of the targets to the planner. Similarly, deception in attacker type could involve obscuring the true attack capabilities, which can confuse the defender's assessment of the attack's treat level. By integrating these elements, the D-APD model encapsulates the intricate interplay of strategy and information, and thus provides a framework for analyzing sophisticated security threats.

3.2.1. The Attack Planner with Multiple Types

In reality, defenders are typically aware only of the range of possible scenarios and the likelihood of each occurring, which is encapsulated in the prior probability. First, denoting the set of all possible payment types as $\Theta = \{\theta^1, \ldots, \theta^m\}$, the corresponding prior probability distribution is $p^{pt} : \Theta \to [0, 1]$. A planner with payment type θ values each target as $v^\theta = (v_1^\theta, \ldots, v_N^\theta)$. Then, we denote the set of all possible hired attacker situations in which the planner with payment type θ as $\Phi^\theta = \{\varphi_1^\theta, \ldots, \varphi_{m^\theta}^\theta\}$; that is, the planner with payment type θ has m^θ possible attacker types, where $\varphi_k^\theta = (\varphi_{k1}^\theta, \varphi_{k2}^\theta)$ means that the planner with payment type θ hired φ_{k1}^θ strong attackers and φ_{k2}^θ weak attackers. The corresponding prior probability distribution is $p^\theta : \Phi^\Theta \to [0, 1]$. $\Phi = \bigcup_{\theta \in \Theta} \Phi^\theta$ represents the set of all attacker types. In practice, although the attacker type for each payment type planner is different, they can be viewed as the same by taking a union. For example, if the set of attacker types for a planner with payment type θ_1 is $\Phi^{\theta_1} = \{\varphi_1, \varphi_2\}$, the set of attacker types for a planner with payment type θ_2 is $\Phi^{\theta_2} = \{\varphi_2, \varphi_3\}$, then we can set $\Phi^{\theta_1} = \Phi^{\theta_2} = \{\varphi_1, \varphi_2, \varphi_3\}$ and $p^{\theta_1}(\varphi_3) = 0, p^{\theta_2}(\varphi_1) = 0$. Therefore, we can directly let $\Phi = \{\varphi^1, \ldots, \varphi^M\}$, which represents the set of all possible attacker types (M in total). $\varphi^i = (\varphi_1^i, \varphi_2^i)$ means hiring a φ_1^i strong attacker and a φ_2^i weak attacker, and we denote the

prior probability of the planner of payment type θ over all attacker types as $p^\theta : \Phi \to [0,1]$. Obviously, each attacker type φ corresponds to a group of attackers, and the base salary of this group of attackers is recorded as $u^\varphi = (u_1^\varphi, \ldots, u_{||\varphi||_1}^\varphi)$; the attack cost matrix is recorded as $ac^\varphi = (ac_{ij}^\varphi)_{i\in[||\varphi||_1],j\in[N]}$. Then, under the strategy profile (sd, sa), the utility function of the planner with payment type θ and attacker type φ is

$$U^{\theta,\varphi}(sd,sa) = \sum_{j=1}^{N} q_j(sd,sa)v_j^\theta - \sum_{i=1}^{||\varphi||_1} U^{\theta,\varphi,i}(sd,sa), \tag{4}$$

where $U^{\theta,\varphi,i}$ is each employed attacker's benefit, that is

$$U^{\theta,\varphi,i}(sd,sa) = u_i^\varphi + \sum_{j=1}^{N} sa_{ij}q_j(sd,sa)\cdot\frac{\alpha v_j^\theta}{sa_j^3} - \sum_{j=1}^{N} sa_{ij}ac_{ij}^\varphi. \tag{5}$$

3.2.2. Attack Planner's Deception

For private information, the planner is usually more inclined to mislead the defenders by deceiving rather than reporting the truth. The practice of sending preemptive signals has become the prevalent method of deception. These signals are crafted to influence the defender's perception of the threat and subsequent resource allocation. However, it is important to note that there is a cost associated with this deceptive behavior, termed the deception cost, denoted as $dc = (dc_1, dc_2)$, dc_1 and dc_2 is the planner's deception cost about payment and attacker type, respectively. First, the planner will achieve deception by simulating the behavior of a planner of other payment types, so the set of signals sent about the payment type is $S^1 = \Theta$ and the corresponding set of mixed-signal sending strategies is $\Delta_{o1} = \{o1 = (o1_{s^1})_{s^1\in S^1} | o1_{s^1} \in [0,1], ||o1||_1 = 1\}$, where $o1_{s^1}$ is the probability of sending the signal s^1. In addition, for the attacker type, the planner will deceive the defender by arranging the attacker to disguise or hide (such as plainclothes police), so that the defender will have a wrong understanding of the real number of hired attackers. Therefore, the number of attackers reported by the planner cannot be more than the number of real hired attackers. Then the set of signals sent by the planner with the payment type θ and the attacker type φ about the attacker type is $S^{2,\theta,\varphi} = \{s^2 \in \mathbb{N}^+ | s^2 \le \varphi_1 + \varphi_2\}$ (the planner only reports the total number of attackers, not the specific number of strong and weak attackers). For example, suppose a planner's set of attacker types is $\{(1,2),(2,4),(3,1)\}$, then the set of signals they send is $\{1,2,3,4,5,6\}$. But in fact, we can find that when the planner sends signal 1 or 2, the effect on the posterior probability is the same as when they send signal 3; this is because when sending signals 1, 2 and 3, the possible types of planner are $(1,2),(2,4),(3,1)$. Therefore, we reasonably let $S^2 = \{||\varphi||_1 | \forall \varphi \in \Phi\}$ represent the set of signals about the planner's attacker type; then, $S^{2,\varphi} = \{s^2 \in S^2 | s^2 \le \varphi_1 + \varphi_2\}$ is the set of signals about the attacker type for the attacker type φ planner, and their collection of mixed signaling strategies is $\Delta_{o2}^\varphi = \{o2 = (o2_{s^2})_{s^2\in S^{2,\varphi}} | o2_{s^2} \in [0,1], ||o2||_1 = 1\}$, where $o2_{s^2}$ is the probability that the planner sends a signal s^2 about their attacker type. Then $\Delta_{o2} = \bigcup_{\varphi\in\Phi}\Delta_{o2}^\varphi$ is the collection of all the mixed strategies about the attacker type.

3.2.3. Game Process

The game begins with the planner committing to a signaling scheme $\pi_o = (\pi_{o1}, \pi_{o2})$, where $\pi_{o1} : \Theta \times \Phi \to \Delta_{o1}$ is the planner's signaling scheme about their payment type, that is, $\pi_{o1}(\theta,\varphi)$ represents the signaling scheme of the planner with payment type θ and the attacker type φ; $\pi_{o1}(s^1|\theta,\varphi)$ is the probability that they send signal s^1, $\pi_{o1}(s^1|\theta) = \sum_{\varphi\in\Phi}\pi_{o1}(s^1|\theta,\varphi)$ is the probability that the planner with payment type θ sends signal s^1. Similarly, $\pi_{o2} : \Theta \times \Phi \to \Delta_{o2}$ is the planner's signaling scheme regarding the attacker type.

Then, after receiving the signal (s^1, s^2), the defender updates their belief on the planner's type according to Bayes' rule, which is the posterior probability. The posterior probability about the payment type is

$$\mu^{pt}(\theta|s^1) = \frac{p^{pt}(\theta)\pi_{o1}(s^1|\theta)}{\sum_{\theta'\in\Theta} p^{pt}(\theta')\pi_{o1}(s^1|\theta')} = \frac{p^{pt}(\theta)\sum_{\varphi\in\Phi}\pi_{o1}(s^1|\theta,\varphi)}{\sum_{\theta'\in\Theta} p^{pt}(\theta')\sum_{\varphi\in\Phi}\pi_{o1}(s^1|\theta',\varphi)}, \tag{6}$$

and the posterior probability about the attacker type of planner with payment type θ is

$$\mu^{an}(\varphi|\theta,s^2) = \frac{p^{pt}(\theta)p^{\theta}(\varphi)\pi_{o2}(s^2|\theta,\varphi)}{\sum_{\varphi'\in\Phi} p^{pt}(\theta)p^{\theta}(\varphi')\pi_{o2}(s^2|\theta,\varphi')} = \frac{p^{\theta}(\varphi)\pi_{o2}(s^2|\theta,\varphi)}{\sum_{\varphi'\in\Phi} p^{\theta}(\varphi')\pi_{o2}(s^2|\theta,\varphi')}. \tag{7}$$

Then the posterior probability that the planner's payment type is θ and attacker's type is φ is $\mu(\theta,\varphi|s^1,s^2) = \mu^{pt}(\theta|s^1)\cdot\mu^{an}(\varphi|\theta,s^2)$. Subsequently, the defender chooses a defense strategy based on their posterior belief to protect targets; we denote their strategic scheme as $\pi_d : S^1 \times S^2 \to SD$, that is, $\pi_d(s^1,s^2) = (\pi_d(j|s^1,s^2))_{j\in[N]}$ represents the strategy that the defender takes after receiving the signal s^1, s^2 and $\pi_d(j|s^1,s^2) \in \{0,1,2\}$ is the defense level of the defender at target j at this time.

Finally, the planner arranges for the attackers to carry out the attack and the strategic scheme is $\pi_a : \Theta\times\Phi\times S^1\times S^2 \to SA$; $\pi_a(\theta,\varphi,s^1,s^2) = (\pi_a(ij|\theta,\varphi,s^1,s^2))_{i\in[||\varphi||_1],j\in[N]}$ is the attacker arrangement strategy adopted by the planner with payment type θ and attacker type φ when sending signals s^1, s^2 and $\pi_a(ij|\theta,\varphi,s^1,s^2) \in \{0,1\}$ represents the case where the i-th attacker is scheduled to attack target j at this time. The game process is shown in Figure 1.

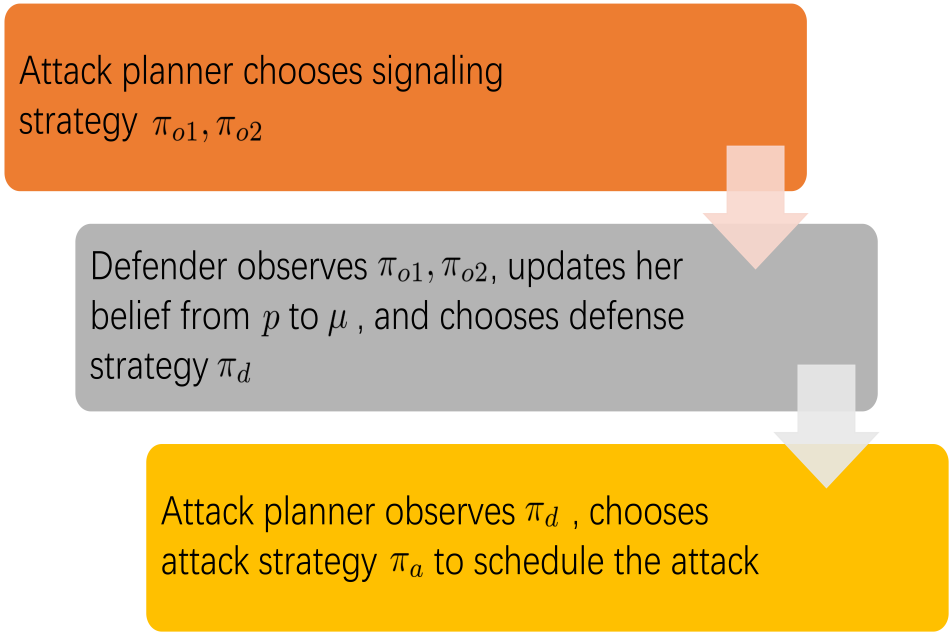


Figure 1. Game process.

3.2.4. Utility of Both Sides

After receiving the signal s^1, s^2, the defender's loss function is

$$L^{s^1,s^2}(\pi_o,\pi_d,\pi_a) = \sum_{\theta\in\Theta}\sum_{\varphi\in\Phi:||\varphi||_1\geq s^2} \mu(\theta,\varphi|s^1,s^2)\cdot L^D(\pi_d(s^1,s^2),\pi_a(\theta,\varphi,s^1,s^2)). \tag{8}$$

The utility function of the planner with payment type θ and the attacker type φ when sending signal s^1, s^2 is

$$U^{\theta,\varphi}_{s^1,s^2}(\pi_d,\pi_a) = U^{\theta,\varphi}(\pi_d(s^1,s^2),\pi_a(\theta,\varphi,s^1,s^2)) - ind(s^1,\theta)\cdot dc_1 - ind(s^2,||\varphi||_1)\cdot dc_2, \tag{9}$$

where $ind(a,b) = \begin{cases} 0, & a = b \\ 1, & a \neq b \end{cases}$ is indicator function. Thus, the total expected utility of the planner of type θ and attacker type φ is

$$U_{AP}^{\theta,\varphi}(\pi_o, \pi_d, \pi_a) = \sum_{s^1 \in S^1} \sum_{s^2 \in S^{2,\varphi}} \pi_{o1}(s^1|\theta,\varphi) \cdot \pi_{o2}(s^2|\theta,\varphi) \cdot U_{s^1,s^2}^{\theta,\varphi}(\pi_d(s^1,s^2), \pi_a(\theta,\varphi,s^1,s^2)), \tag{10}$$

which corresponds to the gains of the i-th attacker, which are shown as

$$U_i^{\theta,\varphi}(\pi_o, \pi_d, \pi_a) = \sum_{s^1 \in S^1} \sum_{s^2 \in S^{2,\varphi}} \pi_{o1}(s^1|\theta,\varphi) \cdot \pi_{o2}(s^2|\theta,\varphi) \cdot U^{\theta,\varphi,i}(\pi_d(s^1,s^2), \pi_a(\theta,\varphi,s^1,s^2)). \tag{11}$$

3.2.5. The Equilibrium

For each planner with the payment type θ and the attacker type φ, given the signaling strategy π_o and defender strategy π_d, let $f_k(\pi_a) = U_k^{\theta,\varphi}(\pi_o, \pi_d, \pi_a)$, $k = 1, \ldots, ||\varphi||_1$; then, by solving the Pareto optimal solutions of the multi-objective problem

$$max_{\pi_a} f = (f_1(\pi_a), \ldots, f_k(\pi_a), \ldots, f_{||\varphi||_1}(\pi_a)),$$

we can obtain a Pareto optimal set, denoted as $ParSA^{\theta,\varphi}$. In Definition 1, we give the specific definition of the model equilibrium solution.

Definition 1. *The strategy profile $(\pi_o^*, \pi_d^*, \pi_a^*)$ is called an equilibrium solution of model if and only if*

- $\pi_o^* =< \pi_{o1}^*(\theta,\varphi), \pi_{o2}^*(\theta,\varphi) >= argmax_{\pi_{o1},\pi_{o2}} U_{AP}^{\theta,\varphi}(\pi_o, \pi_d^*, \pi_a^*),\ \forall\theta,\varphi$
- $\pi_d^*(s^1,s^2) = argmin_{\pi_d} L^{s^1,s^2}(\pi_o^*, \pi_d, \pi_a^*),\ \forall s^1, s^2$
- $\pi_a^*(\theta,\varphi) = argmax_{\pi_a \in ParSA^{\theta,\varphi}} U_{AP}^{\theta,\varphi}(\pi_o^*, \pi_d^*, \pi_a),\ \forall\theta,\varphi.$

3.3. Motivating Example

To explain the model more intuitively, we give an example. Assume there are three targets and the set is $T = \{t_1, t_2, t_3\}$, and the planner has two payment types $\Theta = \{\theta_1, \theta_2\}$ and two attacker types $\Phi = \{\varphi_1, \varphi_2\}$. The specific values are shown in Table 2.

Table 2. Motivation example.

Variable	Value
N	3
m	2
p_1	[0.5 0.5]
M	2
p_2	[0.5 0.5]
α	0.01
v^d	[5000 3000 2000]
Θ	[2000 5000 2000;5000 5000 3000]
bs	[50 30]
Φ	[1 0;1 1]
u_1	50
u_2	[30 50]
ac_1	[2 1 3]
ac_2	[1.5 2 1;1 3 2]
dc	[3 3]
c	[0 20 30]
B	40

When the planner chooses to reveal the true type, we can compute the SSE of this game. For the planner with payment type θ_1 and attacker type φ_1, the defender strategy is $(0, 2, 0)$, i.e., they choose to implement a high-level defense against target t_2, which makes sense because target t_2 is of the highest value to the planner with payment type θ_1; the corresponding attack strategy is 't_3'. For a planner with payment type θ_1 and attacker type φ_2 where the defender chooses not to defend, the corresponding attack strategy is (t_3, t_2), i.e., arrange for a strong attacker to attack target t_3 and a weak attacker to attack target 't_2'. For a planner with payment type θ_2 and attacker type φ_1 where the defender chooses not to defend, the corresponding attack strategy is t_2. For a planner with payment type θ_2 and attacker type φ_2 where the defender implements a high level of defense against the target t_1, the corresponding attack strategy is (t_3, t_2). Under the strategy profile, the loss suffered by the defender was 3750, and the gain gained by the planner was 5633. The game tree is shown in Figure 2.

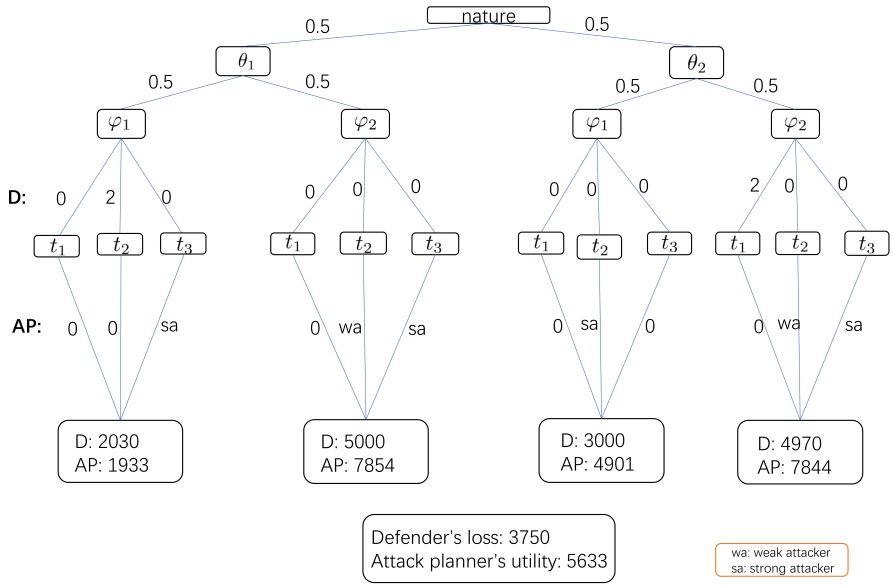


Figure 2. Game tree without deception.

When the planner first commits to a signaling strategy, the planner with payment type θ_1 and the attacker with type φ_1 will mislead the defender into thinking that their payment type is θ_2 by sending signals, causing the defender to mistake target t_1 as the most valuable to them, and will thus choose to implement a high-level defense against target t_1. In this case, the planner will choose to attack the target t_2 so that it benefits and the defender loses. Similarly, the planner with payment type θ_2 and an attacker of type φ_2 misleads the defender into thinking that their attacker type is φ_1 by sending signals, making them gain and causing the defender to suffer losses. With the signaling strategy, the defender ultimately suffered a loss of 4022.5, while the planner made a gain of 6623.5. The game tree is given in Figure 3. It is observed that the outcomes for both parties are more pronounced when a signaling strategy is employed. This instance illustrates that the planner's deception can lead to increased benefits for themselves, simultaneously resulting in more significant detriments for the defender.

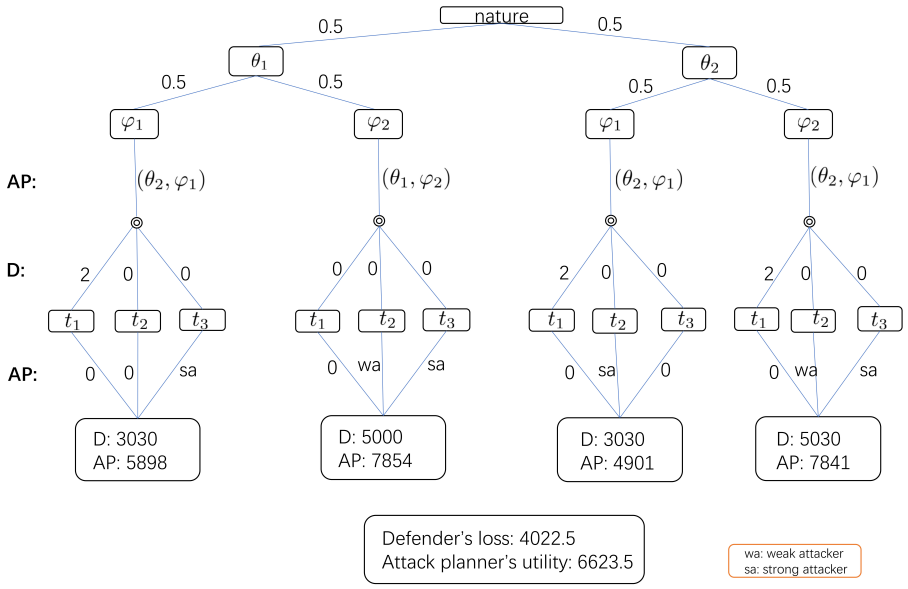


Figure 3. Game tree with deception.

4. Equilibrium Computation

In the introduced model, it is acknowledged that the planner's strategy involves the transmission of two distinct signals, each necessitating individual calculation. It substantially increases the complexity of the resolution procedure. To address this issue, we propose a model simplification. Then, a tri-level programming framework is formulated based on the game process, and a decomposition algorithm is subsequently applied to solve it.

4.1. Modified Model

We propose a model simplification by considering all possible combinations of the two signals. This approach not only reduces the computational burden but also maintains the strategic essence of the signaling process, ensuring that the model remains a faithful representation. Specifically, the planner has $m \cdot M$ possible combinations of types, so the set of types can be reformulated into $TY = \{ty^{\theta,\varphi} = (\theta, \varphi) | \theta \in \Theta, \varphi \in \Phi\}$. Then, the set of all types with payment type θ is represented as $TY^{\theta} = \{ty^{\theta,\varphi} = (\theta, \varphi) | \varphi \in \Phi\}$ and the set of all types with attacker type φ is represented as $TY^{\varphi} = \{ty^{\theta,\varphi} = (\theta, \varphi) | \theta \in \Theta\}$. The corresponding prior probability is $p^{ty} : TY \to [0, 1]$, where $p^{ty}(ty^{\theta,\varphi}) = p^{pt}(\theta) \cdot p^{\theta}(\varphi)$. Rewrite the set of signals sent by the planner as $S = \{s = (s^1, s^2) | s^1 \in S^1, s^2 \in S^2\}$. Denote the set of signals for each type $ty^{\theta,\varphi}$ as $S^{ty^{\theta,\varphi}} = \{s = (s^1, s^2) | s^1 \in S^1, s^2 \in S^{2,\varphi}\}$, and the corresponding mixed signal strategy set is $\Delta_o^{ty^{\theta,\varphi}} = \{o = (o_s)_{s \in S^{ty^{\theta,\varphi}}} | o_s \in [0, 1], ||o_s||_1 = 1\} = \{o = (o_s)_{s \in S} | o_s \in [0, 1], ||o_s||_1 = 1, o_s = 0 \forall s_2 > ||\varphi||_1\}$; set $\Delta_o = \bigcup_{ty \in TY} \Delta_o^{ty^{\theta,\varphi}}$. With a slight abuse of the notation, denote the modified signaling strategy as $\pi_o : TY \to \Delta_o$, and $\pi_o(s|ty^{\theta,\varphi})$ is the probability that the planner with type $ty^{\theta,\varphi}$ sends the signal s. Then, accordingly, the defender's strategy scheme is $\pi_d : S \to SD$, where $\pi_d(s) = (\pi_d(1|s), \ldots, \pi_d(N|s))$ is the defender's strategy after receiving the signal s, and $\pi_d(j|s) \in \{0, 1, 2\}$ represents the defense level in the j-th target. The planner's attack scheme is $\pi_a : TY \times S \to SA$, where $\pi_a(ty^{\theta,\varphi}, s) = (\pi_a(ij|ty^{\theta,\varphi}, s))_{i \in [||\varphi||_1], j \in [N]}$ is the attack strategy after the planner with type $ty^{\theta,\varphi}$ sends the signal s and $\pi_a(ij|ty^{\theta,\varphi}, s) \in \{0, 1\}$. After receiving the signal s, the defender's belief that the planner type is $ty^{\theta,\varphi}$ is redenoted as $\mu(ty^{\theta,\varphi}|s) = \frac{p^{ty}(ty^{\theta,\varphi}) \cdot \pi_o(s|ty^{\theta,\varphi})}{\sum_{ty \in TY} p^{ty}(ty) \cdot \pi_o(s|ty)}$. Then, combined with Equations (2), (4) and (10), the total utility function of the planner is

$$U^{to}(\pi_o, \pi_d, \pi_a) = \sum_{ty^{\theta,\varphi} \in TY} p^{ty}(ty^{\theta,\varphi}) \sum_{s \in S^{ty^{\theta,\varphi}}} \pi_o(s|ty^{\theta,\varphi}) \cdot$$
$$(\sum_{j=1}^{N} q_j(\pi_d(s), \pi_a(ty^{\theta,\varphi}, s)) v_j^{\theta} - \sum_{i=1}^{||\varphi||_1} U_i^{ty^{\theta,\varphi}} - ind(s^1, \theta) \cdot dc_1 - ind(s^2, \varphi) \cdot dc_2), \quad (12)$$

where $U_i^{ty^{\theta,\varphi}}$ is the income of the i-th attacker hired by the planner with type $ty^{\theta,\varphi}$; that is

$$U_i^{ty^{\theta,\varphi}} = u_i^{\varphi} + \sum_{j=1}^{N} \pi_a(ij|ty^{\theta,\varphi}, s) q_j(\pi_d(s), \pi_a(ty^{\theta,\varphi}, s)) \cdot \frac{\alpha v_j^{\theta}}{\pi_a(ty^{\theta,\varphi}, s)_j^3} - \sum_{j=1}^{N} \pi_a(ij|ty^{\theta,\varphi}, s) ac_{ij}^{\varphi}. \quad (13)$$

Combined with Equations (1) and (8), the total loss function of the defender is

$$L^{to}(\pi_o, \pi_d, \pi_a) = \sum_{s \in S} \sum_{ty^{\theta,\varphi} \in TY} \pi_o(s|ty^{\theta,\varphi}) \cdot$$
$$\sum_{ty^{\theta,\varphi} \in TY: ||\varphi||_1 \geq s^2} \mu(ty^{\theta,\varphi}|s) \sum_{j=1}^{N} (q_j(\pi_d(s), \pi_a(ty^{\theta,\varphi}, s)) v_j^d + c_{\pi_d(j|s)+1}). \quad (14)$$

4.2. Tri-Level Programming

In this section, based on the game process, we delineate the D-APD model as a tri-level programming framework that encapsulates the strategic interactions between an attack planner and a defender, as depicted in Figure 4. The following discussion provides an in-depth exposition of this tri-level programming framework.

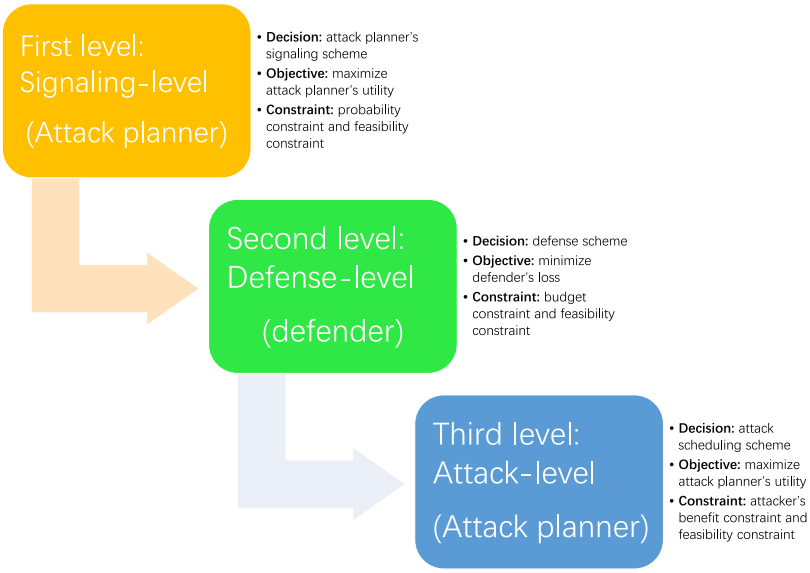


Figure 4. The tri-level programming framework.

4.2.1. Signaling-Level Problem

At the beginning of the game, the planner chooses a signaling strategy to confuse the defender and thus make their own gains greater. Therefore, the first level is the signaling-level (SL), where the planner chooses a signal strategy designed to mislead the defender's decision-making process to achieve greater benefits. Formally, the SL problem optimizes the planner's utility and finds the corresponding optimal signaling strategy and is denoted by $P1$:

$$P1: \ max_{\pi_o} \ U^{to}(\pi_o, \pi_d^*, \pi_a^*) \tag{1a}$$
$$s.t. \ \ \pi_o(s|ty^{\theta,\varphi}) \in [0,1], \ \forall s \in S, ty^{\theta,\varphi} \in TY \tag{1b}$$
$$\sum_{s \in s} \pi_o(s|ty^{\theta,\varphi}) = 1, \ \forall ty^{\theta,\varphi} \in TY \tag{1c}$$
$$\pi_o(s|ty^{\theta,\varphi}) = 0 for s^2 > ||\varphi||_1, \ \forall ty^{\theta,\varphi} \in TY \tag{1d}$$

In $P1$, the objective function (1a) represents the planner's objective to maximize their utility. The decision is to determine the probability of sending each signal. Constraint (1b) and (1c) ensure the validity of the decision variable. Constraint (1d) illustrates the feasibility of the decision variable, that is, the planner can only send the signal with a smaller number of attackers than the actual number of hired attackers.

4.2.2. Defense-Level Problem

Upon receiving the signal from the planner, the defender updates their beliefs in light of the received signals, and accordingly, chooses a defense strategy, aiming to minimize their loss. Thus, the second level, designated as the defense-level (DL), is designed to identify a defense strategy for the defender that minimizes the potential loss. This level is mathematically formalized as $P2$:

$$P2: \ min_{\pi_d} \ L^{to}(\pi_o^*, \pi_d, \pi_a^*) \tag{2a}$$
$$s.t. \ \ \sum_{j=1}^{N} c_{\pi_d(j|s)+1} \leq B \tag{2b}$$
$$\pi_d(j|s) \in \{0,1,2\}, \ \forall j \in [N], s \in S \tag{2c}$$

In $P2$, the objective function (2a) is to minimize the defender's loss. Constraint (2b) indicates that the defense cost cannot exceed its budget. Constraint (2c) sets a ternary restriction on defense decisions.

4.2.3. Attack-Level Problem

After observing the defense strategy, the planner arranges for the attackers to attack a subset of targets and the game ends. Consequently, the third and final level, known as the attack-level (AL), involves the planner deploying the attackers to execute an attack strategy that maximizes the utility. Specifically, in the AL problem, the decision is to decide the specific target for each attacker. It is essential to consider the individual benefits of each attacker simultaneously in formulating the attack strategy, ensuring that the strategy aligns with the Pareto optimal solutions within the multi-objective framework defined by the attackers' benefits. This problem is formalized as $P3$:

$$P3: \ max_{\pi_o} \ U^{to}(\pi_o^*, \pi_d^*, \pi_a) \tag{3a}$$
$$s.t. \ \ \pi_a(ty^{\theta,\varphi}, s) \in ParSA^{ty^{\theta,\varphi}}, \ \forall s \in S \tag{3b}$$
$$\sum_{j=1}^{N} \pi_a(ij|ty^{\theta,\varphi}, s) = 1, \ \forall i \in [||\varphi||_1], s \in S, ty^{\theta,\varphi} \in TY \tag{3c}$$
$$\pi_a(ij|ty^{\theta,\varphi}, s) \in \{0,1\}, \ \forall i \in [||\varphi||_1], j \in [N], s \in S, ty^{\theta,\varphi} \in TY \tag{3d}$$

In $P3$, the objective function (3a) is to maximize the planner's payoffs. Constraint (3c) states that each attacker can only attack one target. Constraint (3d) sets a binary restriction on attack decisions. Constraint (3b) guarantees a profit for each attacker. Formally, for the planner with type $ty^{\theta,\varphi}$, let $g_k(\pi_a) = U_k^{ty^{\theta,\varphi}}(\pi_o, \pi_d, \pi_a), \ k = 1, \ldots, ||\varphi||_1$; then, $ParSA^{ty^{\theta,\varphi}}$ is the Pareto optimal set for the following multi-objective problem $P4$.

$$P4: \quad max_{\pi_a} \quad g = (g_1(\pi_a), \ldots, g_k(\pi_a), \ldots, g_{||\varphi||_1}(\pi_a)) \tag{4a}$$

$$s.t. \quad \sum_{j=1}^{N} \pi_a(ij|ty^{\theta,\varphi}, s) = 1, \ \forall i \in [||\varphi||_1], s \in S \tag{4b}$$

$$\pi_a(ij|ty^{\theta,\varphi}, s) \in \{0,1\}, \ \forall i \in [||\varphi||_1], j \in [N], s \in S \tag{4c}$$

4.3. Algorithm Description

Solving tri-level programming problems is inherently complex, with even the most straightforward scenarios being NP-hard [60]. The extant algorithms for addressing these problems are primarily divided into exact and heuristic methods. Exact algorithms encompass implicit enumeration [61,62], reformation techniques [63] and decomposition algorithms [37,64]. However, implicit enumeration is limited to very small-scale problems. The reformation technique is tailored to problems with continuous lower-level decision variables, which does not align with the discrete variables present in our model. Consequently, the decomposition algorithm has garnered widespread application in research due to its effectiveness. The crux of employing this algorithm lies in its successful decomposition of the problem. Heuristic algorithms are also favored for their swift computation times [65]. But it may yield suboptimal solutions. So, the column and constraint generation algorithm (C&CG) has been introduced as an extension of decomposition algorithms [66,67]. While previous C&CG algorithms are tailored to solve min–max–min or max–min–max problems, they are not directly applicable to the model presented in this paper. In response, we extend the decomposition algorithm and devise a customized decomposition algorithm based on a genetic algorithm (DA-GA) for the tri-level programming problem outlined herein. Our proposed algorithm operates within a master-subproblem framework. The master problem (MP) is tasked with determining the signal strategy, while the subproblem (SP) is designed to extract a defense strategy for a given signal strategy. We will now delve into the construction and computation of both the master problem and subproblem, as well as the intricate process of our customized DA-GA. Given that the resolution of the master problem and subproblem is contingent upon the solution of the lowest attack level problem, we initially present a detailed methodology for addressing the attack level problem in Section 4.3.1, prior to discussing the master problem and subproblem in depth.

4.3.1. Attack-Level Algorithm

The implementation steps of the AL problem's algorithm are demonstrated in Algorithm 1. In fact, the essence of resolving the attack-level problem $P3$ lies in addressing the multi-objective problem $P4$. Given the vast number of strategy combinations available to planners of various types, a direct computational approach would incur significant computational expense. However, since the problem-solving process for planners of different types is analogous and their decisions are mutually independent, we can substantially mitigate computational costs by addressing each type of planner separately. Furthermore, the dimensionality of the attack scheduling strategy space escalates exponentially as the number of hired attackers increases, making it impractical to identify all Pareto optimal solutions by exhaustively examining each potential attack strategy. Leveraging the effectiveness of the NSGA-II algorithm in tackling multi-objective problems, this paper introduces a genetic algorithm (GA) tailored to problem $P4$. Taking the planner with type $ty^{\theta,\varphi}$ as an example, when they send signal s, the specific steps are as follows:

0. Code: The planner's attack scheduling strategy is a $0-1$ matrix with size $||\varphi||_1 \times N$, so we first need to encode it. Specifically, we encode each attack scheduling strategy as an $||\varphi||_1$-dimensional row vector with elements in set $[N]$. For example, a strategy $\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ can be encoded as $(1,5,3)$, which represents that the 1-th attacks the 1-th target, the 2-th attacks the 5-th target and the 3-th attacks the 3-th target. In

fact, the number of columns in each row of the policy matrix with the value of 1 is taken as the encoding of the policy.

1. Determine an initial population P consisting of n individuals that meet the constraints;
2. Calculate the fitness $f = (f_1, f_2)$ of each individual in P;
   (1) For f_1:
   - Calculate the function value for each individual on each objective;
   - For each individual p, calculate the number of individuals n_p that dominate it and the set of individuals S_p that it dominates;
   - Set the f_1 of the individuals with $n_p = 0$ to 1, and the set of these individuals to $\mathcal{F}_1$, then subtract the n_p values of the individuals in S_p by 1; then, set the f_1 of the individuals with $n_p = 0$ at this time to 2, the set of these individuals to $\mathcal{F}_2$, and so on until the f_1 of each individual is calculated.

   (2) For f_2, solve it for each individual in $\mathcal{F}_l$ in (1) separately:
   - For each individual i in $\mathcal{F}_l$, let i's crowding distance $L[i]_d = 0$;
   - Calculate the m-th function value for each individual and sort them in ascending order according to these values;
   - In order to give individuals on the edge a selection advantage, let $L[1]_d = L[end]_d = inf$; that is, after sorting, let the crowding distance of the first and last individuals be infinity. Then for the other individuals $i = 2, \ldots, |\mathcal{F}_l| - 1$

     $$L[i]_d = L[i]_d + (L[i+1]_m - L[i-1]_m)/(f_m^{max} - f_m^{min})$$

     where $L[i+1]_m$ and $L[i-1]_m$ represent the m-th objective function value for the $(i+1)$-th and $(i-1)$-th individuals, and f_m^{max} and f_m^{min} represent the maximum and minimum values of the m-th objective function of the individuals in $\mathcal{F}_l$, respectively;
   - The above three steps are repeatedly calculated for each objective function, and the crowding distance obtained when each individual corresponds to different objective functions is added up to obtain the final crowding distance, which is denoted as f_2.
3. By selecting, crossing and mutating for P, a new population Q is generated;
   (1) Selection operator: The selection is based on fitness using the binary tournament method. Randomly select two individuals for comparison, select the better individual to enter the next generation and repeat this operation until we select enough n individuals where an individual p is better than an individual q, if, and only if, the f_1 of p is less than q or the f_1 of p is equal to q and the f_2 of p is greater than q. That is, compare the rank first; the individual with a lower rank is better, individuals with an equal rank have a comparative crowding distance, and individuals with larger crowding distances are better;
   (2) Crossover operator: For two individuals in the population, a location is randomly selected and the genes at that location are exchanged with a certain probability;
   (3) Mutation operator: In order to ensure the diversity of the population and avoid falling into the local optimal situation, the individuals in the population are randomly selected in a position and the genes at that position are changed.
4. According to step 2, calculate the fitness of each individual in $P \bigcup Q$;
5. Selection* operator: Select the n individuals in $P \bigcup Q$ as the offspring; if there is an l, that makes $\sum_{i=1}^{l} |\mathcal{F}_i| = n$ true, so $P = \bigcup_{i=1}^{l} \mathcal{F}_i$ is the offspring population. Otherwise, when l satisfies the conditions $\sum_{i=1}^{l-1} |\mathcal{F}_i| < n$ and $\sum_{i=1}^{l} |\mathcal{F}_i| > n$, the individuals in $\mathcal{F}_l$ are sorted in descending order of crowding distance (f_2), and then the first $(n - \sum_{i=1}^{l-1} |\mathcal{F}_i|)$ individuals are selected to form offspring P together with $\bigcup_{i=1}^{l-1} \mathcal{F}_i$;

6. Return to step 3 and loop until the termination condition is satisfied, and finally obtain a Pareto optimal solution set.

Algorithm 1: GA for AL problem

Input: model parameters;
Output: the attack scheme π_a^*, attack planner's total utility U^{to};
for $ty^{\theta,\varphi} \in TY$; **do**
  for $s \in S$ **do**
    Generate an initial population at random P;
    Calculate the fitness $f = (f_1, f_2)$ of each individual in P;
    iteration = 1;
    while *iteration* $\leq$ *gen* **do**
      Select parents using the binary tournament method based on fitness for P;
      Perform crossover operator with probability p_c;
      Perform mutation operator with probability p_m;
      Denote generated new population as Q;
      Calculate the fitness of each individual in $P \bigcup Q$;
      Perform select* operator to select n individuals in $P \bigcup Q$ as the offspring and denote it as P;
      iteration = iteration + 1;
    end
    Take the individuals in P whose f_1 value is 1, and archive the set of them as $ParSA_s^{ty^{\theta,\varphi}}$, that is, the Pareto optimal set;
    Calculate attack planner's utility $U_s^{ty^{\theta,\varphi}}$ for each individual in $ParSA_s^{ty^{\theta,\varphi}}$;
    $\pi_a^*(ty^{\theta,\varphi}, s) = argmax_{sa \in ParSA_s^{ty^{\theta,\varphi}}} U_s^{ty^{\theta,\varphi}}(sa)$;
  end
end
$U^{to} = \sum_{ty^{\theta,\varphi} \in TY} p^{ty}(ty^{\theta,\varphi}) \sum_{s \in S} \pi_o^*(s|ty^{\theta,\varphi}) \cdot U_s^{ty^{\theta,\varphi}}$.

4.3.2. Master Problem

First, we construct and solve a master problem by considering a subset of the defense strategy schemes to obtain a signaling strategy. Specifically, we give a set of defense strategies $\hat{\Pi}_d = \{\hat{\pi}_d^1, \ldots, \hat{\pi}_d^h, \ldots, \hat{\pi}_d^H\}$, where h represents the iteration index. For each $\hat{\pi}_d^h$, we define a attack scheme π_a^h. Then the master problem can be formulated as follows:

$$P5: \quad max_{\pi_o, \pi_a^h} \quad U^{to}(\pi_o, \hat{\pi}_d^h, \pi_a^h) \tag{5a}$$

$$s.t. \quad \pi_o(s|ty^{\theta,\varphi} \in [0,1], \ \forall s \in S, ty^{\theta,\varphi} \in TY) \tag{5b}$$

$$\sum_{s \in S} \pi_o(s|ty^{\theta,\varphi}) = 1, \ \forall ty^{\theta,\varphi} \in TY \tag{5c}$$

$$\pi_o(s|ty^{\theta,\varphi}) = 0 \ for \ s^2 > ||\varphi||_1, \ \forall ty^{\theta,\varphi} \in TY \tag{5d}$$

$$\pi_a^h(ty^{\theta,\varphi}, s) \in ParSA^{ty^{\theta,\varphi}}, \ \forall s \in S, h \in [H] \tag{5e}$$

$$\sum_{j=1}^{N} \pi_a^h(ij|ty^{\theta,\varphi}, s) = 1, \ \forall i \in ||\varphi||_1, s \in S, ty^{\theta,\varphi} \in TY, h \in [H] \tag{5f}$$

$$\pi_a^h(ij|ty^{\theta,\varphi}, s) \in \{0,1\}, \ \forall i \in ||\varphi||_1, j \in [N], s \in S, ty^{\theta,\varphi} \in TY, h \in [H] \tag{5g}$$

The planner's signaling strategy is crafted specifically to mislead the defender, and as such, it exerts a direct influence on the defense strategy within the middle-level problem. Then problem $P5$ effectively translates into a linear programming problem, in which, the

dimensions of columns and constraints are finite and the constraints (5e) can be effectively addressed using Algorithm 1. So the master problem can be solved directly in polynomial time by using a professional linear programming solver.

4.3.3. Subproblem

The subproblem is centered on determining the defender's strategy in response to a predefined signaling strategy π_o^*. It is important to recognize that the subproblem is concerned solely with deriving the defender's strategy. The defender's eventual loss, the corresponding attack strategy and the planner's utility are all contingent upon the defender's potentially incorrect perception of the planner's intentions. These outcomes, which are based on this misaligned cognition, require further analysis and are addressed in the attack-level problem. Then, the subproblem is constructed as the following bi-level problem:

$$
\begin{aligned}
P6: \quad & min_{\pi_d} \quad L^{to}(\pi_o^*, \pi_d, \pi_a) && \text{(6a)}\\
& max_{\pi_a} \quad U^{to}(\pi_o^*, \pi_d, \pi_a) && \text{(6b)}\\
& s.t. \quad \sum_{j=1}^{N} c_{\pi_d(j|s)+1} \leq B && \text{(6c)}\\
& \qquad \pi_a(ty^{\theta,\varphi}, s) \in ParSA^{ty^{\theta,\varphi}}, \ \forall s \in S && \text{(6d)}\\
& \qquad \sum_{j=1}^{N} \pi_a(ij|ty^{\theta,\varphi}, s) = 1, \ \forall i \in [||\varphi||_1], s \in S, ty^{\theta,\varphi} \in TY && \text{(6e)}\\
& \qquad \pi_d(j|s) \in \{0,1,2\}, \ \forall j \in [N], s \in S && \text{(6f)}\\
& \qquad \pi_a(ij|ty^{\theta,\varphi}, s) \in \{0,1\}, \ \forall i \in [||\varphi||_1], j \in [N], s \in S && \text{(6g)}
\end{aligned}
$$

This problem's complexity arises from the vast number of possible defense strategy combinations for each signal, and the dimension of the defense strategy space grows exponentially with an increasing number of targets. To address this, we employ a genetic algorithm to find solutions for each signal that the defender might receive. Each defense strategy is represented as an N-dimensional row vector with elements from the set $\{0,1,2\}$, and these vectors are used directly as chromosomes without the need for additional encoding. The defender's objective is to minimize their loss; that is, an individual with a lower loss function value would conventionally be considered more advantageous. Thus, we instead define the fitness of each individual as $f = \sum_{j=1}^{N} v_j^d - L$. Here, L represents the objective function value that corresponds to the defender's strategy under misperception. It is essential to note that during the algorithm's iterative process, we must evaluate whether the individuals in each new generation adhere to the budget constraints. This is a critical step to ensure that the solutions remain feasible within the given parameters. With the signaling scheme π_o^* as a given, the detailed steps of the algorithm are provided in Algorithm 2.

4.3.4. Customized Decomposition Algorithm Based on Genetic Algorithm

To solve the tri-level programming problem in this paper, we propose a customized decomposition algorithm based on a genetic algorithm. Since the goal of the algorithm is to maximize the utility of the planner, first make $\underline{P} = \infty$ starting with $\hat{\Pi}_d = \emptyset$, representing that the defender does not defend against any target, and then make the utility greater by comparing the utility with $\underline{P}$ after each iteration. At each iteration, solve the MP for the defense scheme $\hat{\Pi}_d$ and obtain a signal scheme $\hat{\pi}_o$. Then for this $\hat{\pi}_o$, solve SP using Algorithm 2; obtain the defense scheme $\hat{\pi}_d^h$ and let $\hat{\Pi}_d = \hat{\Pi}_d \bigcup \hat{\pi}_d^h$. Subsequently, solve the AL problem using Algorithm 1 and obtain the objective function value obj_{AL} and corresponding attack scheme $\hat{\pi}_a$. Compare obj_{AL} and $\underline{P}$; if obj_{AL} is larger, update $\pi_o^*, \pi_d^*, \pi_a^*, \underline{P}$ to $\hat{\pi}_o, \hat{\pi}_d^h, \hat{\pi}_a, obj_{AL}$, respectively, otherwise do not update. Then, add a variable and a corresponding set of constraints to MP, and continue solving until the iteration stops. It is

important to note that the $\pi_o^*, \pi_d^*, \pi_a^*$ is the final strategy scheme, and U is given by $\underline{P}$, but L needs to solve $L^{to}(\pi_o^*, \pi_d^*, \pi_a^*)$ for the strategy profile $(\pi_o^*, \pi_d^*, \pi_a^*)$, and cannot be directly given by the objective function value of SP. This is because the SP is carried out under the wrong cognition formed by the defender after receiving the deception signal of the planner; the attack scheme is not the real strategy scheme taken by the planner. The specific steps are shown in Algorithm 3.

Algorithm 2: GA for SP

Input: model parameters;
Output: the defense scheme π_d^*;
for $s \in S$ **do**
  iteration = 1;
  Generate a set of individuals satisfying the budget constraint is randomly as the initial population P;
  while *iteration* $\leq$ *gen* **do**
    For each individual p in P, calculate the corresponding attack scheme π_a^* using Algorithm 1;
    Calculate each individual p's fitness $f = \sum_{j=1}^{N} v_j^d - L(\pi_o^*, p, \pi_a^*)$;
    Select parents using the roulette wheel mechanism from P;
    Perform crossover operator with probability p_c;
    Perform mutation operator with probability p_m;
    For individuals in the resulting population who don't meet the budget constraints, the defense level at the less valuable targets is reduced until the budget constraint is met;
    Denote the generated offspring as P;
    iteration = iteration + 1;
  end
  Find the individual with the highest fitness in final P; that is $\pi_d^*(s)$.
end

Algorithm 3: Customized DA-GA

Input: model parameters;
Output: signal scheme π_o^*, defense scheme π_d^*, attack scheme π_a^*, defender's loss L, attack planner's utility U;
Initialize: $\underline{P} = -\infty, \hat{\Pi}_d = \emptyset, h = 1$;
while $h \leq H$ **do**
  Solve MP($\hat{\Pi}_d$), obtain the signal scheme $\hat{\pi}_o$;
  For $\hat{\pi}_o$, solve SP($\hat{\pi}_o$) using **Algorithm 2**, obtain defense scheme $\hat{\pi}_d^h$;
  $\hat{\Pi}_d \leftarrow \hat{\Pi}_d \bigcup \hat{\pi}_d^h$;
  For $\hat{\pi}_o$ and $\hat{\pi}_d^h$, solve AL problem using **Algorithm 1**, obtain the optimal value obj_{AL} and corresponding attack scheme $\hat{\pi}_a$;
  if $obj_{AL} > \underline{P}$ **then**
    $\pi_o^* = \hat{\pi}_o, \pi_d^* = \hat{\pi}_d^h, \pi_a^* = \hat{\pi}_a, \underline{P} = obj_{AL}$;
  end
  Create a extra variable π_a^h;
  Add the new variable and constraints $(5e) - (5g)$ to MP(P5);
  h = h + 1;
end
$U = \underline{P}$, and for $\pi_o^*, \pi_d^*, \pi_a^*$, calculate $L = L^{to}(\pi_o^*, \pi_d^*, \pi_a^*)$.

5. Experiment and Analysis

In this section, we analyze the model through some numerical experiments. Suppose that the value of each target to both sides is divided into three levels according to their importance to both sides, and the specific values are 10,000, 15,000 and 25,000. v^d and v^Θ are randomly selected among all combinations of the three levels. Let $bs = (350, 200)$ and $\alpha = 0.01$. Each element in the attacker's cost matrix ac takes a random value from $[0, 50]$.

5.1. Algorithm Performance

In this section, we evaluate the computational performance of the customized DA-GA. Figure 5 illustrates that the algorithm is finitely convergent and influenced by the budget constraints and the number of targets. Specifically, the number of iterations increases as the budget increases, as evidenced by comparing (a) with (c) and (b) with (d) in Figure 5. This trend is attributed to the expansion of the strategy space with an increased budget, complicating the search for the optimal strategy and consequently increasing the iterations needed for convergence. Similarly, an increase in the number of targets also leads to a higher number of iterations, as demonstrated by the comparison between (a) and (b), and (c) and (d) in Figure 5.

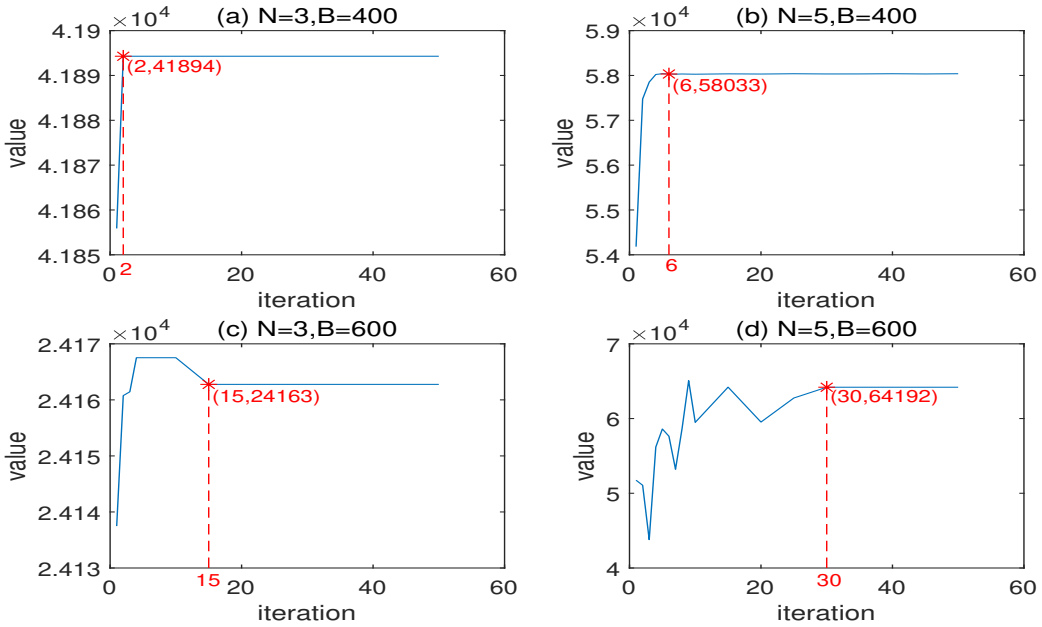


Figure 5. Convergence of DA-GA.

Then, we have compared the runtime of the DA-GA algorithm introduced in this paper against the Enumeration Algorithm (EA). The comparative results are illustrated in Figure 6. It is observed that while the EA holds a marginal advantage in scenarios with a small number of targets, the DA-GA algorithm consistently outperforms EA when the number of targets exceeds four. The performance gap becomes particularly pronounced as the number of targets increases. For instance, when the target count reaches six, the EA's execution time exceeds 500 min, demonstrating the computational inefficiency of EA at higher target volumes. In contrast, the DA-GA algorithm exhibits a significant performance advantage under these conditions. This comparative analysis underscores the efficiency of the DA-GA in solving complex problems involving multiple targets.

Furthermore, to validate the accuracy of the algorithm, we have compared its computational results with the exact algorithm, as depicted in the Table 3. The findings indicate that the error margin of the DA-GA algorithm is confined within a narrow range.

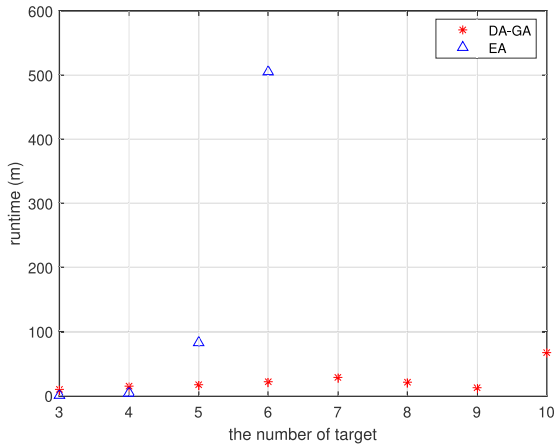


Figure 6. Algorithm runtime.

Table 3. Effectiveness of DA-GA.

	B = 400		**B = 600**	
N	**EA**	**DA-GA**	**EA**	**DA-GA**
3	4.1894×10^4	4.1894×10^4	2.4163×10^4	2.4163×10^4
4	5.5653×10^4	5.5653×10^4	4.6396×10^4	4.6399×10^4
5	5.8032×10^4	5.8039×10^4	6.4190×10^4	6.4192×10^4

5.2. The Impact of Defense Budgets

The impact of defense budgets on the planner's utility is analyzed and presented in Figure 7. With an increase in the defense budget, the planner's utility is observed to decrease in a stepwise pattern. This inverse relationship is logical, given that a larger defense budget enables broader target protection, which in turn lowers the probability of a successful attack and thus reduces the planner's utility. Moreover, within a certain range of the defense budget, minor budget variations lead to negligible changes in the planner's utility. This phenomenon occurs as minor budget adjustments do not substantially change the defender's strategy. For example, a defense budget of 150 or 200 units allows the defender to provide only low-level protection for a single target. This analysis highlights the critical role of budget allocation in defense strategy.

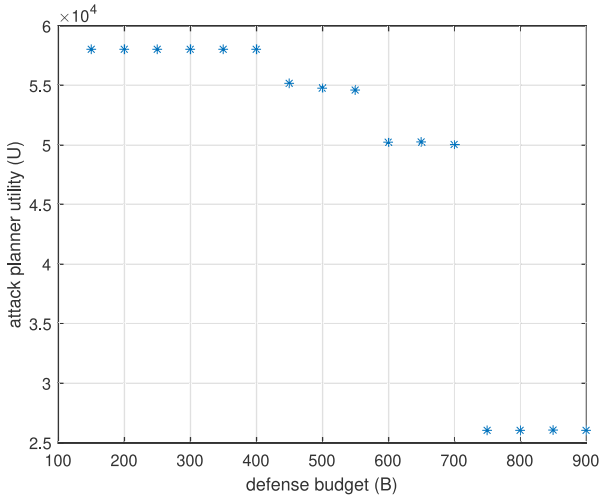
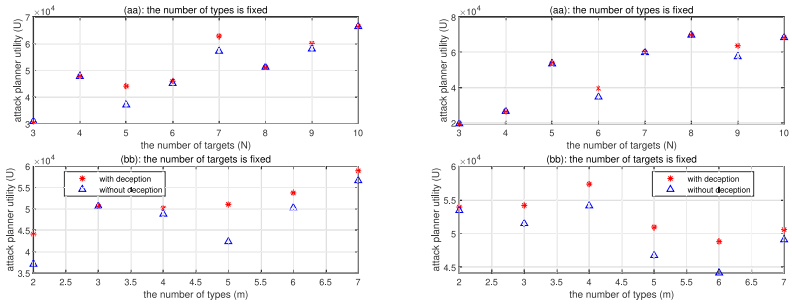


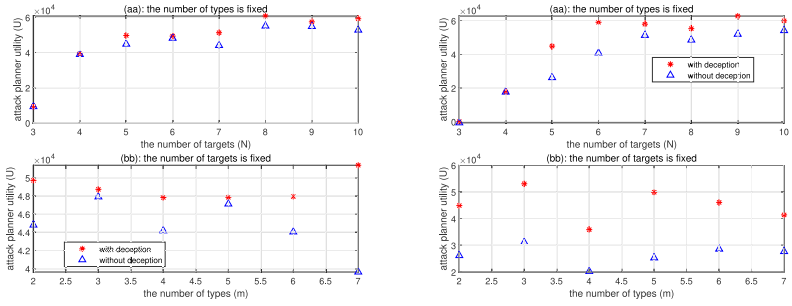
Figure 7. The impact of defense budgets.

5.3. The Impact of Signaling Strategy

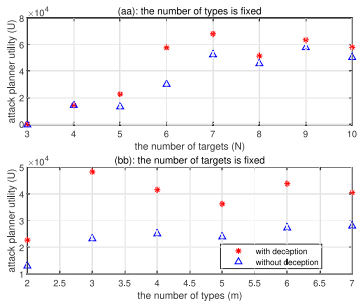
Following the results from Section 5.2, in this section, we calculate the utility of the attack planner when the defense budget is 300, 450, 600, 750 and 900, respectively, with and without How to solve without the signaling strategy is in the Appendix A. the signaling strategy, and subsequently analyze the impact of the signaling strategy on the utility. For each subfigure (aa) in Figure 8a–e, we set $m = 2, M = 3$ and for each subfigure (bb) in Figure 8a–e, we set $N = 5$. The findings indicate that the utility of the planner employing a signaling strategy exceeds that of the planner not employing such a strategy. This suggests a propensity for the planner to deceive the defender in order to reap greater benefits. Additionally, each (aa) in Figure 8 shows that an increase in the number of targets causes the planner's utility to increase. This increase is logical, given that a constant defense budget implies that the defender cannot effectively protect all targets, facilitating the planner's selection of high-value targets for attack to maximize their utility. Furthermore, each (bb) in Figure 8 exhibits a more pronounced difference in the planner's utility between scenarios with and without signaling strategies compared to corresponding (aa). This suggests that signaling strategies are more responsive to the diversity of payment types than to the number of targets. With an increased number of payment types, the defender's uncertainty about the planner's intentions escalates, enabling the planner to exploit this uncertainty through signaling and to enhance the signal strategy's effectiveness.



(**a**) B = 300 (**b**) B = 450



(**c**) B = 600 (**d**) B = 750



(**e**) B = 900

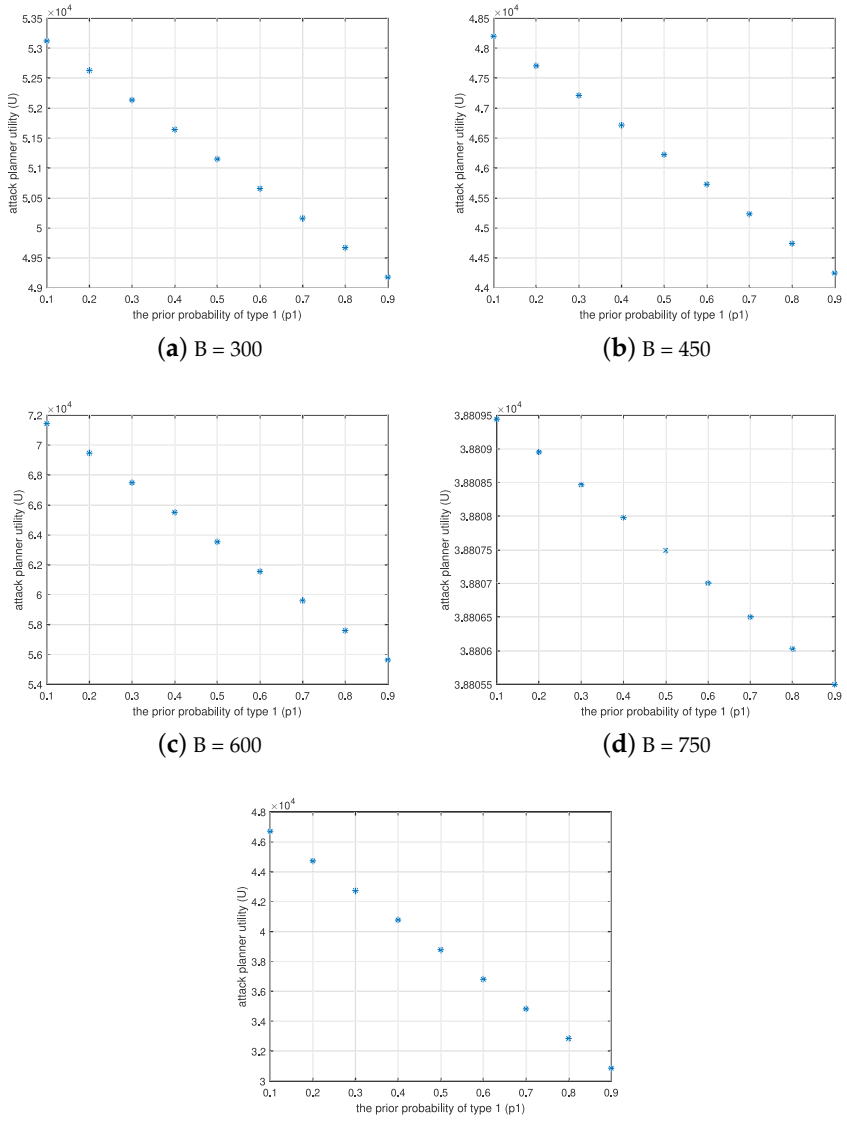
Figure 8. The impact of signaling strategy.

5.4. The Impact of Prior Probability

Furthering our investigation, we examine the impact of prior probability on the planner's utility. To facilitate the analysis, we focus on a scenario involving solely two types of planners. We propose the following theorem:

Theorem 1. *When there are two types of attack planner, there is a linear relationship between their utility and the prior probability of each type (the proof of this theorem is in the Appendix B).*

We proceed to validate Theorem 1 through experimentation. The number of targets was fixed at five. Utilizing the outcomes from Section 5.2, we analyzed how the planner's utility varied with changes in the prior probability across defense budgets of 300, 450, 600, 750 and 900, respectively. The findings are presented in Figure 9. The results demonstrate a linear decrease in the planner's utility with an increase in prior probability, consistent with the predictions of Theorem 1. This trend is attributed to the reduction in the defender's uncertainty about the planner's type due to an elevated prior probability. Consequently, the defender protects the targets with increased precision, resulting in a corresponding decrease in the planner's utility.



(**a**) B = 300 (**b**) B = 450

(**c**) B = 600 (**d**) B = 750

(**e**) B = 900

Figure 9. The impact of prior probability.

5.5. The Impact of Deception Cost

Our analysis further explores the influence of deception costs on signaling strategies, with a summary of the findings presented in Table 4. The data indicate a positive correlation between the cost of deception and the planner's tendency to report their types truthfully. As the deception cost increases, the probability of the planner reporting their types truthfully also rises. Furthermore, the analysis indicates that the relative magnitude of deception costs associated with payment and attacker types significantly influences the planner's signaling strategy. When the deception cost is lower for the payment type than for the attacker type, the planner is more inclined to deceive the defender regarding the payment type. Conversely, the planner is more likely to mislead the defender about the attacker type.

Table 4. The impact of deception costs on signaling strategy.

	Type	**1**	**2**	**3**	**4**	**5**
dc = [0,50]	1	(1,1)	(2,1)	(1,1)	(2,1)	(2,1)
	2	(1,2)	(4,2)	(5,2)	(5,2)	(1,2)
	3	(1,3)	(5,3)	(3,3)	(3,3)	(1,3)
dc = [10,50]	1	(3,1)	(5,1)	(2,1)	(5,1)	(5,1)
	2	(1,2)	(1,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(4,3)	(4,3)	(5,3)
dc = [20,50]	1	(1,1)	(5,1)	(5,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(3,3)	(3,3)	(4,3)	(5,3)
dc = [30,50]	1	(1,1)	(1,1)	(3,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [40,50]	1	(1,1)	(5,1)	(3,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [50,50]	1	(1,1)	(5,1)	(3,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [60,50]	1	(1,1)	(2,3)	(3,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [70,50]	1	(1,1)	(2,3)	(3,2)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [80,50]	1	(1,1)	(2,2)	(3,1)	(4,2)	(5,3)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [90,50]	1	(1,1)	(2,3)	(3,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)
dc = [100,50]	1	(1,1)	(2,3)	(3,1)	(4,1)	(5,1)
	2	(1,2)	(2,2)	(3,2)	(4,2)	(5,2)
	3	(1,3)	(2,3)	(3,3)	(4,3)	(5,3)

6. Conclusions

This paper models the interaction between an attack planner and a defender, in which the attack planner hires some attackers and arranges them to attack the target. To ensure the attackers' compliance with the arrangement, we safeguard their interests by incorporating a multi-objective problem into our model. In addition, considering that defenders may not fully grasp various details about the attack planner, which could be exploited using

the attack planner to induce misjudgment and ineffective defense strategies, we develop a D-APD signaling game model to make up for the deficiencies in prior research on this type of attack. To address the model, we formulate it as a tri-level programming framework according to the game process, and propose a customized DA-GA for its computation. Then, we validate the algorithm's effectiveness through experimental validation. Moreover, by assessing the benefits of an honest planner, we determine that the planner is more likely to opt for deception to secure greater benefits. Hence, there is an imperative to investigate effective defenses against these attacks. Finally, through controlled variable experiments, we establish that the defense budget can significantly curtail the attack planner's incomes, which further verify the significance of budgetary considerations in defense strategy.

We proceed under the assumption that attackers will adhere to the directives of the planner. However, in reality, hired attackers may defy the planner's attack arrangements due to inadequate compensation or financial incentives from the defenders, among other factors, introducing a layer of complexity. Consequently, our future research will delve into this issue. Furthermore, despite our detailed examination of a specific attack behavior, the critical aspect lies in our response to it. Thus, the implementation of effective defenses by defenders in response to such complex attacks remains a central focus of our future research, and is the paramount objective of our study on attack behaviors.

Author Contributions: Conceptualization, S.H. and M.L.; methodology, S.H.; validation, S.H. and R.Z.; formal analysis, S.H.; investigation, S.H., R.Z. and M.L.; writing—original draft preparation, S.H.; writing—review and editing, S.H.; supervision, M.L. and R.Z.; funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Nature Science Foundation of China under grant number: T2350710232.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

When there is no signal strategy, the model is a Stackelberg game; therefore, the strategies and utility of both the attack planner and the defender are solved using the following problem P7:

$$
\begin{aligned}
P7: \quad min_{sd} \quad & \sum_{j=1}^{N}(q_j(sd, sa^*)v_j^d + c_{sd_j+1}) && \text{(A1)}\\
s.t. \quad & \sum_{j=1}^{N} q_j(sd, sa^*)v_j^\theta - \sum_{i=1}^{||\varphi||_1}(u_i + \sum_{j=1}^{N} sa_{ij}^* q_j(sd, sa^*) \cdot \frac{\alpha v_j^\theta}{sa_j^{*3}} - \sum_{j=1}^{N} sa_{ij}^* ac_{ij}) \geq \\
& \sum_{j=1}^{N} q_j(sd, sa)v_j^\theta - \sum_{i=1}^{||\varphi||_1}(u_i + \sum_{j=1}^{N} sa_{ij} q_j(sd, sa) \cdot \frac{\alpha v_j^\theta}{sa_j^{3}} - \sum_{j=1}^{N} sa_{ij} ac_{ij}), \\
& \forall sa \in ParSA^{\theta,\varphi} && \text{(A2)}\\
& \sum_{j=1}^{N} c_{sd_j+1} \leq B && \text{(A3)}\\
& sd_j \in \{0, 1, 2\},\ \forall j \in [N] && \text{(A4)}
\end{aligned}
$$

This is a discrete planning problem, and in fact, it is equivalent to the SP in which the signal strategy is an identity matrix, i.e., the attack planner sends a truthful signal about its own type. So, we can solve it through solve SP(E) (E represents an identity matrix), using Algorithm 2.

Appendix B

Proof of Theorem 1. Suppose that there are two types of attack planner, that is ty_1, ty_2; the prior probability of ty_1 is p_1, so the probability of ty_2 is $1 - p_1$. Then, the attack planner's total utility is

$$\begin{aligned}
U = p_1 \cdot \sum_{s \in S^{ty_1}} \pi_o(s|ty_1) \cdot (\sum_{j=1}^{N} q_j(\pi_d(s), \pi_a(ty_1, s)) v_j^{a1} - \sum_{i=1}^{||n1||_1} U_i^{ty_1} - ind(s^1, ty_1^1) \cdot dc_1 \\
- ind(s^2, ty_1^2) \cdot dc_2) + (1 - p_1) \cdot \sum_{s \in S^{ty_2}} \pi_o(s|ty_2) \cdot (\sum_{j=1}^{N} q_j(\pi_d(s), \pi_a(ty_2, s)) v_j^{a2} - \sum_{i=1}^{||n2||_1} U_i^{ty_2} \\
- ind(s^1, ty_2^1) \cdot dc_1 - ind(s^2, ty_2^2) \cdot dc_2) \\
= p_1 \cdot (\sum_{s \in S^{ty_1}} \pi_o(s|ty_1) \cdot (\sum_{j=1}^{N} q_j(\pi_d(s), \pi_a(ty_1, s)) v_j^{a1} - \sum_{i=1}^{||n1||_1} U_i^{ty_1} - ind(s^1, ty_1^1) \cdot dc_1 \\
- ind(s^2, ty_1^2) \cdot dc_2) - \sum_{s \in S^{ty_2}} \pi_o(s|ty_2) \cdot (\sum_{j=1}^{N} q_j(\pi_d(s), \pi_a(ty_2, s)) v_j^{a2} - \sum_{i=1}^{||n2||_1} U_i^{ty_2} - ind(s^1, ty_2^1) \cdot dc_1 \\
- ind(s^2, ty_2^2) \cdot dc_2)) + \sum_{s \in S^{ty_2}} \pi_o(s|ty_2) \cdot (\sum_{j=1}^{N} q_j(\pi_d(s), \pi_a(ty_2, s)) v_j^{a2} - \sum_{i=1}^{||n2||_1} U_i^{ty_2} - ind(s^1, ty_2^1) \cdot dc_1 \\
- ind(s^2, ty_2^2) \cdot dc_2).
\end{aligned}$$

From this equation, we can obtain that U is linearly related to p_1. □

References

1. van Stackelberg, H. *Marktform und Gleichgewicht*; Springer: Berlin, Germany, 1934.
2. Page, F.H. Optimal contract mechanisms for principal-agent problems with moral hazard and adverse selection. *Econ. Theory* **1991**, *1*, 323–338. [CrossRef]
3. Gan, J.; An, B.; Wang, H.; Sun, X.; Shi, Z. Optimal pricing for improving efficiency of taxi systems. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013.
4. Carnes, T.; Nagarajan, C.; Wild, S.M.; van Zuylen, A. Maximizing influence in a competitive social network: A follower's perspective. In Proceedings of the Ninth International Conference on Electronic Commerce, Minneapolis, MN, USA, 19–22 August 2007; pp. 351–360.
5. Li, Y.; Conitzer, V. Game-theoretic question selection for tests. *J. Artif. Intell. Res.* **2017**, *59*, 437–462. [CrossRef]
6. Tambe, M. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*; Cambridge University Press: Cambridge, UK, 2011.
7. Brown, M.; Sinha, A.; Schlenker, A.; Tambe, M. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
8. Fang, F.; Nguyen, T.H. Green security games: Apply game theory to addressing green security challenges. *ACM SIGecom Exch.* **2016**, *15*, 78–83. [CrossRef]
9. Fang, F.; Stone, P.; Tambe, M. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. *IJCAI* **2015**, *15*, 2589–2595.
10. Fang, F.; Nguyen, T.; Pickles, R.; Lam, W.; Clements, G.; An, B.; Singh, A.; Tambe, M.; Lemieux, A. Deploying paws: Field optimization of the protection assistant for wildlife security. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30, pp. 3966–3973.
11. Zhang, C.; Gholami, S.; Kar, D.; Sinha, A.; Jain, M.; Goyal, R.; Tambe, M. Keeping pace with criminals: An extended study of designing patrol allocation against adaptive opportunistic criminals. *Games* **2016**, *7*, 15. [CrossRef]
12. Kar, D.; Nguyen, T.H.; Fang, F.; Brown, M.; Sinha, A.; Tambe, M.; Jiang, A.X. Trends and applications in Stackelberg security games. In *Handbook of Dynamic Game Theory*; Springer: Cham, Switzerland, 2017; pp. 1–47.
13. Delle Fave, F.M.; Jiang, A.X.; Yin, Z.; Zhang, C.; Tambe, M.; Kraus, S.; Sullivan, J.P. Game-theoretic patrolling with dynamic execution uncertainty and a case study on a real transit system. *J. Artif. Intell. Res.* **2014**, *50*, 321–367. [CrossRef]
14. An, B.; Ordóñez, F.; Tambe, M.; Shieh, E.; Yang, R.; Baldwin, C.; DiRenzo, J., III; Moretti, K.; Maule, B.; Meyer, G. A deployed quantal response-based patrol planning system for the US Coast Guard. *Interfaces* **2013**, *43*, 400–420. [CrossRef]
15. Leitmann, G. On generalized Stackelberg strategies. *J. Optim. Theory Appl.* **1978**, *26*, 637–643. [CrossRef]

16. Von Stengel, B.; Zamir, S. *Leadership with Commitment to Mixed Strategies*; Technical Report; London School of Economics: London, UK, 2004.
17. Letchford, J.; Conitzer, V.; Munagala, K. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory: Proceedings of the Second International Symposium, SAGT 2009, Paphos, Cyprus, 18–20 October 2009*; Proceedings 2; Springer: Berlin/Heidelberg, Germany, 2009; pp. 250–262.
18. Kiekintveld, C.; Islam, T.; Kreinovich, V. Security games with interval uncertainty. In Proceedings of the Twelfth International Conference on Autonomous Agents and Multiagent Systems AAMAS'2013, Saint Paul, MN, USA, 6–10 May 2013.
19. Nguyen, T.; Yadav, A.; An, B.; Tambe, M.; Boutilier, C. Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
20. Conitzer, V.; Sandholm, T. Computing the optimal strategy to commit to. In Proceedings of the 7th ACM conference on Electronic Commerce, Ann Arbor, MI, USA, 11–15 June 2006; pp. 82–90.
21. Paruchuri, P.; Pearce, J.P.; Marecki, J.; Tambe, M.; Ordonez, F.; Kraus, S. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, Estoril, Portugal, 12–16 May 2008; Volume 2, pp. 895–902.
22. Jain, M.; Pita, J.; Tambe, M.; Ordónez, F.; Paruchuri, P.; Kraus, S. Bayesian Stackelberg games and their application for security at Los Angeles International Airport. *ACM SIGecom Exch.* **2008**, *7*, 1–3. [CrossRef]
23. Blum, A.; Haghtalab, N.; Procaccia, A.D. Learning optimal commitment to overcome insecurity. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1826–1834.
24. Haghtalab, N.; Fang, F.; Nguyen, T.H.; Sinha, A.; Procaccia, A.D.; Tambe, M. Three strategies to success: Learning adversary models in security games. In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 9–15 July 2016.
25. Roth, A.; Ullman, J.; Wu, Z.S. Watch and learn: Optimizing from revealed preferences feedback. In Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing, Cambridge, MA, USA, 19–21 June 2016; pp. 949–962.
26. Peng, B.; Shen, W.; Tang, P.; Zuo, S. Learning optimal strategies to commit to. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 2149–2156.
27. Gan, J.; Xu, H.; Guo, Q.; Tran-Thanh, L.; Rabinovich, Z.; Wooldridge, M. Imitative follower deception in stackelberg games. In Proceedings of the 2019 ACM Conference on Economics and Computation, Phoenix, AZ, USA, 24–28 June 2019; pp. 639–657.
28. Gan, J.; Guo, Q.; Tran-Thanh, L.; An, B.; Wooldridge, M. Manipulating a learning defender and ways to counteract. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8274–8283.
29. Nguyen, T.H.; Wang, Y.; Sinha, A.; Wellman, M.P. Deception in finitely repeated security games. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 2133–2140.
30. Nguyen, T.H.; Sinha, A.; He, H. Partial adversarial behavior deception in security games. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Virtual, 7–15 January 2021.
31. Nguyen, T.H.; Sinha, A. Sequential Manipulative Attacks in Security Games. In Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems, London, UK, 3–7 May 2021.
32. Nguyen, T.H.; Sinha, A. The Art of Manipulation: Threat of Multi-Step Manipulative Attacks in Security Games. *arXiv* **2022**, arXiv:2202.13424.
33. Guo, Q.; An, B.; Bosanský, B.; Kiekintveld, C. Comparing Strategic Secrecy and Stackelberg Commitment in Security Games. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 3691–3699.
34. Cheng, Z.; Chen, G.; Hong, Y. Single-leader-multiple-followers Stackelberg security game with hypergame framework. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 954–969. [CrossRef]
35. Korzhyk, D.; Conitzer, V.; Parr, R. Security games with multiple attacker resources. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
36. Li, Q.; Li, M.; Tian, Y.; Gan, J. A risk-averse tri-level stochastic model for locating and recovering facilities against attacks in an uncertain environment. *Reliab. Eng. Syst. Saf.* **2023**, *229*, 108855. [CrossRef]
37. Ghorbani-Renani, N.; González, A.D.; Barker, K.; Morshedlou, N. Protection-interdiction-restoration: Tri-level optimization for enhancing interdependent network resilience. *Reliab. Eng. Syst. Saf.* **2020**, *199*, 106907. [CrossRef]
38. Zhuang, J.; Bier, V.M. Reasons for secrecy and deception in homeland-security resource allocation. *Risk Anal. Int. J.* **2010**, *30*, 1737–1743. [CrossRef] [PubMed]
39. Zhuang, J.; Bier, V.M. Secrecy and deception at equilibrium, with applications to anti-terrorism resource allocation. *Def. Peace Econ.* **2011**, *22*, 43–61. [CrossRef]
40. Zhuang, J.; Bier, V.M.; Alagoz, O. Modeling secrecy and deception in a multiple-period attacker–defender signaling game. *Eur. J. Oper. Res.* **2010**, *203*, 409–418. [CrossRef]
41. Zhang, C.; Ramirez-Marquez, J.E.; Wang, J. Critical infrastructure protection using secrecy—A discrete simultaneous game. *Eur. J. Oper. Res.* **2015**, *242*, 212–221. [CrossRef]
42. Chen, L.; Li, M.; Liu, J. Modeling bluffing behavior in signaling security games. *Int. Trans. Oper. Res.* **2022**, *29*, 1825–1841. [CrossRef]

43. Xu, H.; Freeman, R.; Conitzer, V.; Dughmi, S.; Tambe, M. Signaling in Bayesian Stackelberg Games. In Proceedings of the 15th International Conference on Au- tonomous Agents and Multiagent Systems AAMAS, Singapore, 9–13 May 2016; pp. 150–158.
44. Xu, H.; Rabinovich, Z.; Dughmi, S.; Tambe, M. Exploring information asymmetry in two-stage security games. *Proc. AAAI Conf. Artif. Intell.* **2015**, *29*, 1057–1063. [CrossRef]
45. Li, X.; Yi, S.; Sycara, K. Multi-agent deception in attack-defense stochastic game. In *Distributed Autonomous Robotic Systems: Proceedings of the 15th International Symposium, Online, 1–4 June 2022*; Springer: Cham, Switzerland, 2022; pp. 242–255.
46. Nguyen, T.; Xu, H. When can the defender effectively deceive attackers in security games? In Proceedings of the AAAI Conference on Artificial Intelligence, Philadephia, PA, USA, 27 February–2 March 2022; Volume 36, pp. 9405–9412.
47. Esmaeeli, S.; Hassanpour, H.; Bigdeli, H. Deception in multi-attacker security game with nonfuzzy and fuzzy payoffs. *Iran. J. Numer. Anal. Optim.* **2022**, *12*, 542–566.
48. Wan, Z.; Cho, J.H.; Zhu, M.; Anwar, A.H.; Kamhoua, C.; Singh, M.P. Resisting multiple advanced persistent threats via hypergame-theoretic defensive deception. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 3816–3830. [CrossRef]
49. Nguyen, T.H.; Butler, A.; Xu, H. Tackling Imitative Attacker Deception in Repeated Bayesian Stackelberg Security Games. In Proceedings of the ECAI, Santiago de Compostela, Spain, 29 August–8 September 2020; pp. 187–194.
50. Long, N.V.; Sorger, G. A dynamic principal-agent problem as a feedback Stackelberg differential game. *Cent. Eur. J. Oper. Res.* **2010**, *18*, 491–509. [CrossRef]
51. Caldentey, R.; Haugh, M. A Cournot-Stackelberg model of supply contracts with financial hedging and identical retailers. *Found. Trends Technol. Inf. Oper. Manag.* **2017**, *11*, 124–143. [CrossRef]
52. Lin, Y.; Ren, Z.; Touzi, N.; Yang, J. Random horizon principal-agent problems. *SIAM J. Control. Optim.* **2022**, *60*, 355–384. [CrossRef]
53. Behnk, S. On the Reduction of Deception in Principal-Agent Relationships. Ph.D. Thesis, Universitat Jaume I, Castello de la Plana, Spain, 2015.
54. Duffy, R.; St John, F.A.; Büscher, B.; Brockington, D. The militarization of anti-poaching: Undermining long term goals? *Environ. Conserv.* **2015**, *42*, 345–348. [CrossRef]
55. Gill, C.; Weisburd, D.; Telep, C.W.; Vitter, Z.; Bennett, T. Community-oriented policing to reduce crime, disorder and fear and increase satisfaction and legitimacy among citizens: A systematic review. *J. Exp. Criminol.* **2014**, *10*, 399–428. [CrossRef]
56. Linkie, M.; Martyr, D.J.; Harihar, A.; Risdianto, D.; Nugraha, R.T.; Maryati; Leader-Williams, N.; Wong, W.M. Editor's Choice: Safeguarding Sumatran tigers: Evaluating effectiveness of law enforcement patrols and local informant networks. *J. Appl. Ecol.* **2015**, *52*, 851–860. [CrossRef]
57. Moreto, W.D. Introducing intelligence-led conservation: Bridging crime and conservation science. *Crime Sci.* **2015**, *4*, 15. [CrossRef]
58. Smith, M.; Humphreys, J. The Poaching Paradox: Why South Africa's 'Rhino Wars' Shine a Harsh Spotlight on Security and Conservation. In *Environmental Crime and Social Conflict*; Routledge: London, UK, 2016; pp. 197–220.
59. Huang, T.; Shen, W.; Zeng, D.; Gu, T.; Singh, R.; Fang, F. Green security game with community engagement. *arXiv* **2020**, arXiv:2002.09126.
60. Bard, J.F. Some properties of the bilevel programming problem. *J. Optim. Theory Appl.* **1991**, *68*, 371–378. [CrossRef]
61. Parajuli, A.; Kuzgunkaya, O.; Vidyarthi, N. Responsive contingency planning of capacitated supply networks under disruption risks. *Transp. Res. Part Logist. Transp. Rev.* **2017**, *102*, 13–37. [CrossRef]
62. Parajuli, A.; Kuzgunkaya, O.; Vidyarthi, N. The impact of congestion on protection decisions in supply networks under disruptions. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *145*, 102166. [CrossRef]
63. Lei, X.; Shen, S.; Song, Y. Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Comput. Oper. Res.* **2018**, *90*, 97–109. [CrossRef]
64. Ghorbani-Renani, N.; González, A.D.; Barker, K. A decomposition approach for solving tri-level defender-attacker-defender problems. *Comput. Ind. Eng.* **2021**, *153*, 107085. [CrossRef]
65. Fakhry, R.; Hassini, E.; Ezzeldin, M.; El-Dakhakhni, W. Tri-level mixed-binary linear programming: Solution approaches and application in defending critical infrastructure. *Eur. J. Oper. Res.* **2022**, *298*, 1114–1131. [CrossRef]
66. Ding, T.; Yao, L.; Li, F. A multi-uncertainty-set based two-stage robust optimization to defender–attacker–defender model for power system protection. *Reliab. Eng. Syst. Saf.* **2018**, *169*, 179–186. [CrossRef]
67. Fang, Y.P.; Zio, E. An adaptive robust framework for the optimization of the resilience of interdependent infrastructures under natural hazards. *Eur. J. Oper. Res.* **2019**, *276*, 1119–1136. [CrossRef]