GAME THEORY APPLICATIONS IN SECURITY

By

ABDOLMAJID YOLMEH

A dissertation submitted to the
School of Graduate Studies
Rutgers, The State University of New Jersey
In partial fulfillment of the requirements
For the degree of
Doctor of Philosophy
Graduate Program in Industrial and Systems Engineering
Written under the direction of
Melike Baykal-Gürsoy
And approved by

____

____

____

____

____

New Brunswick, New Jersey
January, 2021

ABSTRACT OF THE DISSERTATION

Game Theory Applications in Security

by ABDOLMAJID YOLMEH

Dissertation Director:

Melike Baykal-Gürsoy

In this dissertation, we study the applications of game theory in determining protection strategies for various infrastructures. The game models are played between a defender (she) and an adversary (he). The defender seeks to minimize the damage to the infrastructure network, while the adversary aims to maximize it. This dissertation is divided into two parts. In the first part, we consider the resource allocation game models, and in the second part, we study patrolling and search games.

In the area of resource allocation games, we address some of the existing limitations in literature. One such limitation is that most of these models assume that the parameters of the game are either deterministic or follow a known distribution. Whereas in reality, some parameters of the game may be uncertain with no known distribution or distributional information about them may be unreliable. To this end, we study one-shot security games under uncertainty about target valuations. We propose a model in which both players use a robust approach to contend with the uncertainty of target valuations. We show that the Nash equilibrium for this model is of threshold type and develop closed-form solutions to characterize the equilibrium point. We then apply our model to a real case of assigning funds for security to 10 urban areas in the United States.

Another limitation is the lack of models that address hierarchical decision making. Protecting infrastructures and their users against intentional attacks involves making both strategic and operational decisions in an organization's hierarchy. Although usually analyzed separately, these decisions influence each other. To address this issue, we develop a two-stage game model. In the first stage, the players make investment decisions and in the second stage, they decide which sites to defend/attack. We distinguish between two types of games that arise in the second stage: Maximal Damage game and Infiltration/Harassment game. We prove that the solution to this game under budget constraints is unique. In fact, when the second stage game is of Infiltration/Harassment type, the invest-defend game has a unique closed-form solution that is very intuitive. The results reveal that an increase in defense investments on a target site decreases the probability of both defending and attacking that target. However, an increase in attack investments increases the probability of both defending and attacking that target. Similarly, an increase in the defender's (attacker's) investment efficiency leads to a decrease (increase) in investments of both the defender and the attacker. We also apply the proposed model to a real case. The results from real data demonstrate that the attacker's penalty from a failed attack is an important factor in determining the defender's optimal distribution of investments and defense probabilities. The defender's second stage defense decisions complement the first stage investment decisions. That is, among target sites that receive little or zero investment, the most important one is covered with a relatively high defense probability in the second stage. Moreover, as the attacker's budget increases, the defense investments shift from less important sites to the more important ones.

We also investigate the overarching protection options in the resource allocation models. An overarching protection refers to an option that protects multiple targets at the same time, e.g., emergency response, border security and intelligence. Most of the defensive resource allocation models with overarching protections assume that there is only one overarching protection option that protects all targets. However, this may not be realistic, for example, emergency response investment may cover only a certain region. To address this issue, we develop a new resource allocation model to accommodate generalized overarching protections against intentional attacks. The model also considers multiple natural disaster types. We show that our proposed model is a convex optimization problem and therefore can be solved to optimality in polynomial time. Furthermore, the overall country-level resource allocation problem can be decomposed into smaller city-level subproblems, thus resulting in a more efficient algorithm. The numerical experiments demonstrate the performance of the proposed approach.

Patrolling and search games are usually played on a graph where players make decisions over a time horizon. In patrolling games, the defender controls a set of patrollers and directs them to follow a walk on the graph to minimize the damage of attacks of the adversary, while the adversary selects a target and a time to attack. In order to successfully destroy a target site, the adversary needs some preparation time without being interrupted by the patrollers. Most patrolling game models assume that the site values are either the same or that they do not change over time. However, this is not a realistic assumption. Particularly in the case of soft targets, these values may correspond to the occupancy level of a site, thus, as such may be different and may change over time. We propose new models with time-dependent node values and node-based attack times. We solve these models numerically using algorithms like column generation, and column and row generation. We apply these algorithms to a real case of an urban rail network in a major US city. The results show the efficiency of the proposed solution approach. They also demonstrate a diminishing returns for additional patrollers.

In search games, a Hider hides a set of objects in a set of potential hiding locations. The Searcher controls a set of search teams and directs them to follow a walk on the network and find the hidden objects such that an objective function is optimized.

Most search game models assume that the hiding places are identical and the players' objective is to optimize the search time. However, there are some cases in which the players may differentiate the hiding places from each other and the objective is to optimize a weighted search time. To address this, we introduce a new discrete search game with consideration given to the weights at different locations. We show that, under certain conditions, the game has a closed-form Nash equilibrium. For the general case, we develop an algorithm based on column and row generation. We show that the Searcher's subproblem is NP-hard and propose a branch and price algorithm to solve it. We also present a polynomial time algorithm for the Hider's subproblem. Numerical experiments investigate the performance of the approach and reveal insights on the properties of this game.

Dedication

To My Family.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Melike Baykal-Gürsoy. She has always been very patient and enthusiastic and I have found her advice and support invaluable. I feel very fortunate and privileged for being a student of such a great mentor. Without her guidance, this research project would not have been possible.

I would like to thank my dissertation committee members, Dr. David W. Coit, Dr. Thomas Lidbetter, Dr. Tuğrul Özel and Dr. Predrag Spasojevic. I would also like to thank all the Rutgers ISE faculty. My knowledge of Industrial Engineering has grown so much in the past years and I owe that to the professors of the many classes I have taken as a graduate student at Rutgers University. In particular, I would like to thank Dr. Mohsen A. Jafari, Dr. Myong K. Jeong, and Dr. Hoang Pham for all the knowledge I learned from them. I also want to thank my wonderful and supportive family, and all my friends.

This research is supported by:

- National Science Foundation (Grant Number: 1436288)
- National Institute of Justice (Grant Number: 2018-R2-CX-0011)

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Terrorist attacks are a serious concern for the national economy and the quality of life. Every year, thousands of people lose their lives or get injured or kidnapped due to these attacks. In 2015, a total of 11,774 terrorist attacks occurred worldwide, resulting in more than 28,300 deaths and more than 35,300 injuries. In addition, more than 12,100 people were kidnapped or taken hostage [22]. The psychological impact of the continued threat of terrorism is also considerable. Such incidents create fear, panic, anxiety and distress in the society.

Protecting critical infrastructures against terrorism is one of the top priorities in homeland security [104]. The physical protection of critical infrastructure can prevent the successful execution of high-impact terrorist attacks. In addition, the immediate response to a terrorist attack against critical infrastructures can prevent the cascading effects associated with such attacks.

These reasons along with the many high profile terrorist attacks that have happened during the past couple of decades, have highlighted the modeling and analyzing security of such infrastructures as a major research agenda. The consequences of attacks could be substantially reduced by evaluating the risk associated with each site within the infrastructure, mitigation planning, and designing protection strategies and response policies. Infrastructure security has recently been a subject of increased interest from researchers. Different approaches have been proposed to model strategic interactions in security problems, these methods include system analysis [115], mathematical modeling [51], probabilistic risk analysis [33, 39, 73, 100, 115, 116], and adversarial risk analysis [123]. However, since the terrorists can be strategic in their attacks, game-theoretic analysis of such attacks yields more realistic results. Thus, recent studies concentrated on developing game-theoretic models to capture terrorism risk and applying the results in enhancing security measures. One such model, ARMOR [112, 113, 114, 118] has been deployed at the Los Angeles International Airport (LAX) to enhance security of the airport.

This research focuses on game theory applications in finding the optimal protection strategies for various infrastructures against intentional attacks. This work can be divided into two parts: resource allocation models, and patrolling and search game models.

Resource allocation to protect against intentional attacks is generally expensive and deciding how to allocate resources in order to protect critical infrastructures is a difficult problem. Many factors affect such allocation policy, for example, equity plays a significant role in determining defense allocations in practice [129]. Moreover, creating a balance to protect against different types of threats (e.g. biological attacks versus bomb attacks, or between terrorism and nonterrorism prevention activities) is another factor. Some of these factors have already been addressed in the literature of static security games. However, there are still some limitations. For example, most infrastructure security games assume that the parameters of the game are either deterministic or follow a known distribution. Whereas in reality, some parameters of the game may be uncertain with no known distribution or distributional information about them may be unreliable. In this study, we develop robust distribution-free models of the incomplete-information infrastructure security game with and without private information. Moreover, hierarchical nature of decision making is often ignored in the literature. However, allocating resources to protect critical infrastructures involves decision making at different levels in an organization's hierarchy: strategic and operational decisions. The decisions influence each other and need to be studied simultaneously. In this research, we develop two-stage game models to address this issue. Moreover, most of the existing resource allocation models with overarching protection options assume that there is only one overarching protection option that protects all targets. However, in reality there may be many overarching protection options and each option may cover only a subset of targets. To address this issue, we develop a new resource allocation model with generalized overarching protection options. We also develop efficient decomposition algorithms to find the optimal resource allocation.

The patrolling and search games are usually played on a graph where the players make decisions over a time horizon. Designing patrols to protect open mass transit systems and other soft targets poses unique challenges that have not been addressed in the patrolling games literature so far. One of these challenges is the dynamic nature of crowd sizes inside these systems. Because the adversary's primary objective is to inflict human casualties, the node values depend on the number of people residing in those nodes. These numbers change over time and the terrorists tend to time their attacks according to these changes [68]. Other challenges include dealing with multiple attackers, accommodating human resource limitations, and developing efficient methods to design patrols for a general network. We address these challenges by developing new models with dynamically changing node values, node-based attack times, multiple patrollers, and multiple attackers. In order to efficiently solve these models, we develop advanced solution algorithms such as column generation, and column and row generation. In the search games, a Hider hides a set of objects in a set of potential hiding locations. The Searcher controls a set of search teams and directs them to follow a walk on the network and find the hidden objects such that an objective function is optimized. Most search game models assume that the hiding places are identical and the players' objective is to optimize the search time. However, there are some cases in which the players may differentiate the hiding places from each other and the objective is to optimize a weighted search time. To address this, we introduce a new discrete search game with consideration given to the weights at different locations.

1.1 Problem Statement and Motivation of Research

The main problem under consideration in this research is to determine the optimal protection strategies against intentional disruptions such as terrorist attacks. Because adversaries are also strategic in their decision making, game-theoretic analysis of these problems yields more realistic results. The game models considered in this research are played between a defender (she) and an adversary (he). The defender wants to minimize the damage to the infrastructure network, while the adversary wants to maximize it. The models can be categorized into two classes: resource allocation games, and patrolling and search games. In the resource allocation models, there is a set of N targets. Each target i has a value of C_i. The defender decides on which target to defend, while the adversary decides on which one to attack. If both players choose the same target i, then with probability δ_i, the attack will be detected and thwarted. This probability is called the detection probability. Component (i, j) of the following matrix shows the expected damage if the defender chooses target i and the adversary chooses target j. Note that, this matrix corresponds to the payoff matrix of the adversary, who tries to maximize the expected damage.

$$\begin{array}{c c} i \setminus j & \begin{array}{cccc} 1 & 2 & \cdots & N \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \vdots \\ N \end{array} & \begin{pmatrix} (1-\delta_1)C_1 & C_2 & \cdots & C_N \\ C_1 & (1-\delta_2)C_2 & \cdots & C_N \\ \vdots & \vdots & \ddots & \vdots \\ C_1 & C_2 & \cdots & (1-\delta_N)C_N \end{pmatrix} \end{array}.$$

Our aim is to characterize the Nash equilibrium (NE) in closed-form under various conditions such as the uncertainty of game parameters, and the existence of private information.

Another issue that we address in this dissertation is the hierarchical nature of decision making. Protecting infrastructures and their users against disruptions involves making both strategic and operational decisions in an organization's hierarchy (See Figure 1.1). The strategic decisions are long-term decisions with long-lasting effects. For example, investment decisions on "hardening" [17] of target sites to decrease success probability of attack is classified as a strategic decision. These include investment on new technologies to enhance security of a site. On the other hand, the operational decisions are short-term decisions that relate to the routine day-to-day operations such as patrolling, assigning first responders, and scheduling vehicle checkpoints. Note that, the word "strategic" can also be used to describe players. In this context a "strategic player" means a rational player whose objective is to maximize payoff. Therefore, in this dissertation "strategic decision" means long-term decision with long-lasting impacts and "strategic player" means a rational player whose objective is to maximize payoff. Most research only focus on either purely strategic decisions [63, 107] or purely operational decisions [16, 35, 36, 38]. However, these decisions influence each other. For instance, installing a CCTV camera in a certain area might render patrolling that area unnecessary. Or allocations of metal detectors

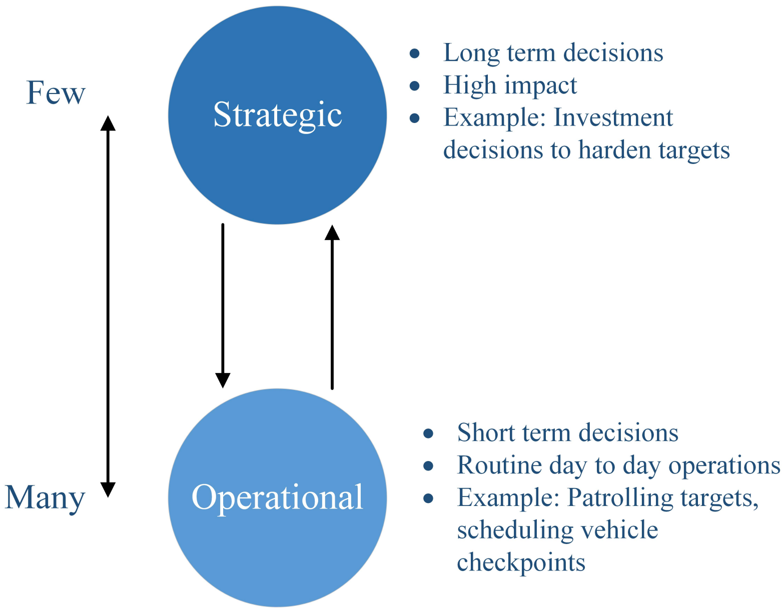

Few
Many
Strategic
Operational
• Long term decisions
• High impact
• Example: Investment decisions to harden targets
• Short term decisions
• Routine day to day operations
• Example: Patrolling targets, scheduling vehicle checkpoints

Figure 1.1: Strategic decisions vs operational decisions

and screening systems to target sites may affect optimal scheduling of patrol units among those targets. Moreover, investing in a new technology to enhance security of a certain target site may reduce its target attractiveness and affect the optimal probability of defending that target. Therefore, considering strategic and operational decisions in the same model would yield a more holistic analysis.

We study the effect of overarching protection options in resource allocation models considering both manmade and natural disasters. Overarching protection options refer to the alternatives that can protect multiple targets simultaneously. For example, investments in border security and intelligence efforts are expected to protect multiple targets from the threat of terrorism. The limitation of the existing literature in this area is that most of the existing models only consider a single overarching protection option that protects all targets. However, this may not be an accurate representation of reality. For example, investment in border security can be divided into different points of entry, each of which is expected to benefit areas that are closer to that particular point of entry. To this end, a new resource allocation model that accommodates multiple overarching protections that protect a subset of the targets, would lead to a more realistic analysis.

A patrolling game G investigated in this research is a zero-sum game played by a defender and an adversary on a connected graph $Q = (\mathcal{N}, \mathcal{E})$ with the set of nodes $\mathcal{N}$ and the set of edges $\mathcal{E}$ over the time horizon $\mathcal{T}$. The defender controls a set of security personnel (patrollers) $\mathcal{S}$ and directs them to follow a walk on the graph to minimize the damage of attacks from the adversary. While the adversary controls a set of attackers $\mathcal{A}$ and chooses a node and a time to attack for each attacker. In order to successfully destroy a target site, an attacker needs a certain number of time units on the target uninterrupted by any patroller. Majority of the papers in the literature of patrolling games assume that the adversary chooses a single target to attack, the target values are fixed over time, and some even assume that all targets are indistinguishable, i.e., they all have the same value. However, this is not the case in many realistic situations. For example, at a transportation facility, the number of people, occupancy level, at each location may be considered as the value of that location. Moreover, occupancy levels may change over time, it is expected that, during the rush hours, the occupancy levels would be higher than normal hours. Therefore, a patrolling game model with time-dependent node values, node-specific attack times, multiple patrollers and multiple attackers would lead to results that are more aligned with reality.

The search games considered in this research are played between a Searcher and a Hider. The Searcher controls a set of S search teams and the Hider controls a set of H objects to hide. The game is played on a complete graph $Q = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{0, 1, 2, \ldots, N\}$ is the set of nodes in the graph and $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the set of edges. Most of the search game models in the literature assume that the hiding places are identical and the players' objective is to optimize the search time. However, there are some cases in which the players may differentiate the hiding places from each other and the objective is to optimize a weighted search time. For example, in certain attacks (biological or chemical), casualty rate depends on factors such as population density, environment conditions etc. Therefore, different locations may have different casualty rates and the overall damage will be proportional to exposure time and casualty rate. Another example is the problem of detecting an eavesdropping agent over communication channels [37]. Different channels may have different transmission capacities and the rate of damage to the network will be proportional to the detection time and the capacity of the channel. Moreover, the hiding locations may be dispersed throughout a large area and the search may involve multiple search teams. To this end, a new search game that accommodates different weights at different locations, will lead to a more realistic analysis.

1.2 Research Contributions

In this research, new game-theoretic models are proposed to address some of the existing gaps in the areas of resource allocation games, and patrolling and search games. In the area of resource allocation games, the main contributions are: extending the existing models to handle hierarchical decision making; introducing generalized overarching protection options; addressing parameter uncertainties using a robust approach; and developing new models that are amenable to more efficient algorithms. In the area of patrolling and search games, our main contributions are: incorporating time-dependent node values, as well as multiple patrollers and multiple attack points; and introducing new and more efficient algorithms to solve the game-theoretic models.

In the next sections we will present our main contributions as given below.

1. We develop a robust approach to cope with parameter uncertainties in the security games and provide closed-form NE strategies in Chapter 2.

2. To address the hierarchical nature of decision making to protect against intentional attacks, in Chapter 3, we introduce a two-stage invest-defend game model and derive closed-form NE strategies under certain conditions. This model captures the combined effect of strategic investment decisions and operational attack/defense decisions.

3. In Chapter 4, we present a new resource allocation model for the protection of assets against both manmade and natural disasters with generalized overarching protection. This model is shown to lead to a convex optimization problem that is decomposable, thus can be efficiently solved.

4. In Chapters 5 and 6, we introduce new patrolling game models with time dependent node values, node based attack times, multiple patrollers, and multiple attack points; and develop efficient solution approaches, based on column generation, and column and row generation to solve realistic size problems.

5. We introduce a new search game model with different node weights, multiple search teams, multiple objects to hide, and dispersed hiding locations; and efficient solution approaches, based on column and row generation, to solve realistic size models in Chapter 7.

6. We present the conclusions of this research and discuss future research ideas in Chapter 8.

Part I

Resource Allocation Security Games

Chapter 2

Infrastructure Security Games Under Uncertainty: a Robust Approach

2.1 Introduction and Literature Review

Most studies in the literature of security games assume that the parameters of a game (such as occupancy levels, detection probabilities etc.) are known with certainty. However, this is not a realistic assumption because in reality we can only estimate some of these parameters based on historical data or expert judgments, both of which can be inaccurate. Although occupancy levels may be available to the defender through infrared or vision sensors, the attacker may only gather historical data. One possible approach to incorporate parameter uncertainty within a game is the Bayesian game model [52, 53, 54] that uses distributional information about the parameters of the game. However, such distributional information may not also be readily available to the players, or they may opt not to use potentially inaccurate distributional information. Moreover, the equilibrium strategy of the defender may be seriously affected by such pre-specified probability distributions. Consequently, some researchers consider robustness to address parameter uncertainty in game theoretic models. For example, Aghassi and Bertsimas [1] relax the assumptions of Harsanyi's Bayesian game model and present an alternative distribution-free equilibrium concept, *robust-optimization equilibrium*, for games with payoff uncertainty. In this approach, players try to optimize their worst case payoff functions simultaneously. The authors prove the existence of such equilibria for arbitrary robust finite games with bounded polyhedral payoff uncertainty sets. In the context of security applications, Nikoofal and Zhuang [107] develop a game theoretic model in which the defender uses a robust approach to tackle her uncertainty about the attacker's target valuation. In this model, they suggest a Stackelberg game model in which the defender acts as the leader and the attacker is the follower. This means that the attacker can observe the the defender's decision and acts accordingly, which might not always be the case. In some cases, the defender may opt not to reveal her decision, in such cases, simultaneous move games are more appropriate than Stackelberg games. Nikoofal and Zhuang [108] study the significance of the first mover's advantage and robustness of strategies under secrecy in the presence of private information. Shan and Zhuang [130] investigate the robustness of the proposed game theoretic model under the presence of strategic and non-strategic attackers. One difference between their model and ours is that in their model, one of the attackers is completely non-strategic, however, in our model, attackers are both strategic but have different objectives. Moreover, robustness in their paper refers to the sensitivity of the equilibrium to the defender's mistaken assumption about the attacker's type. However, in our study, robustness is introduced with respect to the parameter uncertainty. Kiekintveld *et al.* [75] present Stackelberg type security games and apply a robust optimization approach to optimize the worst case payoff to the defender. However, they do not address the attacker's private information in their model. Kardeş [74] proposes a robust optimization model for n-person stochastic games with finite states and actions, and uncertain payoffs. He develops an explicit mathematical programming formulation to compute equilibrium strategies for the case of polytopic uncertainty sets. The private information about player types is not included in the model. However, in reality, players may have some private information, such as their personal preferences or their attitude toward risk, that is not shared with other players. Qian *et al.* [121] study a Stackelberg game in which the adversary is risk averse, however, the defender is uncertain about the degree of the attacker's risk aversion and uses a robust approach to contend with this uncertainty. In this model, the adversary has complete knowledge about the defender's payoff, however, in our model both players are uncertain about the game parameters. Xu and Zhuang [140] introduce a game model in which the defender has private information about her own vulnerability. The adversary can invest in learning activities to gain intelligence about the defender's private information, while the defender decides on investment in counter-learning efforts. This paper is different from our study in the sense that in this paper, the defender has private information, while in our model, the adversary has private information. Moreover, they do not address parameter uncertainty in their model.

In this chapter, we develop robust models for infrastructure security games, both with and without private information, in which the players use a robust optimization approach to cope with payoff uncertainty. We present analytical results about the existence and uniqueness of the robust equilibrium for each game. We apply the proposed approach to real data on annual terrorism losses in the 10 most valuable urban areas of the United States. The results of the proposed model can be implemented to determine the optimal defensive resource allocation among these areas. The rest of the chapter is organized as follows. In section 2.2, the problem under consideration is described, three models are proposed to capture the security game under uncertainty. In section 2.3, the proposed approach is applied to real data.

2.2 Proposed Model

This section introduces a one-shot infrastructure security game. There are N sites in the infrastructure that are potential targets. There is a single defender and a single adversary, therefore, each player can choose only one site in the one-shot game. The adversary and the defender simultaneously choose their strategies over the potential sites. Payoff matrices for both the defender and the adversary are based on the occupancy level, $\tilde{C}_i$, of each site i in the infrastructure. $\tilde{C}_i$ is an uncertain parameter that has a compact and convex support $\left[\underline{C}_{i,}\overline{C}_i\right]$, and this range is known to both players. If the defender defends site i and the adversary attacks site $j, j \neq i$, a successful attack on site j will be launched. Therefore, payoff to the defender is $-\tilde{C}_j$ and the adversary receives a payoff of $\tilde{C}_j$. However, if both players choose the same site i, the attack will be detected with probability $\tilde{\delta}_i$, which is also uncertain and $\tilde{\delta}_i \in \left[\underline{\delta}_i, \overline{\delta}_i\right]$. Hence the defender's payoff becomes $-\left(1 - \tilde{\delta}_j\right)\tilde{C}_j$ and the adversary's becomes $\left(1 - \tilde{\delta}_j\right)\tilde{C}_j$. This means that, even when both rivals are at the same site, there is a probability that the defender may not detect the adversary. There are no assumptions about distributions of the uncertain parameters over their respective uncertainty intervals.

While the defender always attempts to minimize her expected damage, the objective of the adversary may vary depending on his type. There are two possible types of the adversary: maximum damage (MD) adversary and infiltration/harassment (INF) adversary. The MD adversary seeks to maximize his expected payoff, thus differentiates between the potential sites based on their occupancy levels. However, this is not the case for an INF adversary, for whom all sites are the same and the aim is to increase the probability of having a successful attack. In this section, three models are investigated. In the first model, the defender plays the security game with a MD adversary and knows the type of the adversary, in the second model the defender plays the security game against an INF adversary type, in the third model the defender is uncertain about the type of the adversary and only knows that with probability q, the adversary is a MD adversary and with probability $1-q$ the adversary is an INF adversary. Throughout the chapter, we assume that the sites are sorted in the order of decreasing $\underline{C}_i$s and $\underline{C}_i$s are distinct i.e. $\underline{C}_1 > \underline{C}_2 > \cdots > \underline{C}_N$. The first assumption is not restrictive by any means, it only requires rearrangement of site indexes so that the sites are sorted. As for the second assumption, our results will still hold even when $\underline{C}_i$s are not distinct, however, we are making this assumption in order to simplify the resulting formulas. In the following subsections, we describe and analyze our proposed models.

2.2.1 Model 1: Maximum Damage Game

In this model, the adversary wants to inflict the maximum damage. We assume that the defender knows the intention of the adversary i.e. there is no private information. In this case, the payoff to the adversary is:

$$u_A^1\left(\mathbf{x},\mathbf{y}\right) = \sum_{i=1}^{N}\left(1-\tilde{\delta}_i x_i\right)\tilde{C}_i y_i,$$

where x_i and y_i are the probability of choosing site i, by the defender and adversary, respectively. Therefore $\mathbf{x} = [x_1, x_2, ..., x_N]$ and $\mathbf{y} = [y_1, y_2, ..., y_N]$ are the defender's and the adversary's mixed strategies, respectively, and $y_i \geq 0, x_i \geq 0, \forall i = 1, 2, \ldots, N, \sum_i x_i = \sum_i y_i = 1$. In order to contend with the uncertainty of the game, both players use the robust approach, meaning that they seek to optimize their worst case expected payoff, where the worst case is taken with respect to the set of possible values for the uncertain parameters and the expectation is taken with respect to the mixed strategies of both players [1]. Hence, the adversary's best response to the defender's strategy $\mathbf{x}$ is:

$$\mathbf{y}^{*} = \arg\max_{\mathbf{y}} \min_{\substack{\tilde{\delta}_i\in\left[\underline{\delta}_i,\overline{\delta}_i\right] \\ \tilde{C}_i\in\left[\underline{C}_i,\overline{C}_i\right]}} \left(\sum_{i=1}^{N}\left(1-\tilde{\delta}_i x_i\right)\tilde{C}_i y_i\right).$$

Note that, the minimum of $\left(\sum\limits_{i=1}^{N}\left(1-\tilde{\delta}_i x_i\right)\tilde{C}_i y_i\right)$ in the above equation occurs when $\tilde{\delta}_i = \overline{\delta}_i$ and $\tilde{C}_i = \underline{C}_i$, thus giving the attacker's best response as $\mathbf{y}^{*} = \arg\max\limits_{y}\left(\sum\limits_{i=1}^{N}\left(1-\overline{\delta}_i x_i\right)\underline{C}_{i,}y_i\right)$. Using the same robust approach, the defender wants to minimize the maximum expected damage, therefore her best response to the adversary's mixed strategy $\mathbf{y}$ is:

$$\mathbf{x}^{*} = \arg\min_{\mathbf{x}} \max_{\substack{\tilde{\delta}_i\in\left[\underline{\delta}_i,\overline{\delta}_i\right] \\ \tilde{C}_i\in\left[\underline{C}_i\overline{C}_i\right]}} \left(\sum_{i=1}^{N}\left(1-\tilde{\delta}_i x_i\right)\tilde{C}_i y_i\right).$$

The maximum of $\left(\sum\limits_{i=1}^{N}\left(1-\tilde{\delta}_i x_i\right)\tilde{C}_i y_i\right)$ in the above equation happens at $\tilde{\delta}_i = \underline{\delta}_i$ and $\tilde{C}_i = \overline{C}_i$. Hence, the defender's best response is $\mathbf{x}^{*} = \arg\min\limits_{x}\left(\sum\limits_{i=1}^{N}\left(1-\underline{\delta}_i x_i\right)\overline{C}_i y_i\right)$. The following presents the payoff matrix to both players:

$$\begin{array}{c|cccc}
i\setminus j & 1 & 2 & \cdots & N \\
\hline
1 & -(1-\underline{\delta}_1)\overline{C}_1,(1-\overline{\delta}_1)\underline{C}_1 & -\overline{C}_2,\underline{C}_2 & \cdots & -\overline{C}_N,\underline{C}_N \\
2 & -\overline{C}_1,\underline{C}_1 & -(1-\underline{\delta}_2)\overline{C}_2,(1-\overline{\delta}_2)\underline{C}_2 & \cdots & -\overline{C}_N,\underline{C}_N \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
N & -\overline{C}_1,\underline{C}_1 & -\overline{C}_2,\underline{C}_2 & \cdots & -(1-\underline{\delta}_N)\overline{C}_N,(1-\overline{\delta}_N)\underline{C}_N
\end{array}.$$

In this matrix, at each position, the first number is the payoff to player 1 (defender) and the second number is the payoff to player 2 (adversary). Since the payoffs to the players do not add up to zero, or a fixed amount, this is a non-zero sum game. The following lemma gives the necessary and sufficient condition for the non-zero sum game to have a pure Nash Equilibrium (NE).

Lemma 2.1. *The maximum damage game has a pure Nash equilibrium if and only if* $\left(1-\overline{\delta}_1\right)\underline{C}_1 \geq \underline{C}_2$.

Proof. Suppose we have $\left(1-\overline{\delta}_1\right)\underline{C}_1 \geq \underline{C}_2$. It is easy to check that $\mathbf{x}=(1,0,0,...,0)$, $\mathbf{y}=(1,0,0,...,0)$ is a pure NE strategy pair. This establishes the sufficiency part. We prove the necessity part by contradiction, suppose that $\left(1-\overline{\delta}_1\right)\underline{C}_1 < \underline{C}_2$ and the game has a pure NE, this pure NE is definitely not $\mathbf{x}=(1,0,0,...,0)$, $\mathbf{y}=(1,0,0,...,0)$, because at this strategy profile the adversary can strictly increase his/her payoff by attacking site 2. Moreover it has to be on the diagonal of the matrix i.e. $x_i=y_i=1$ for some $i>1$ however, this implies that $\left(1-\overline{\delta}_i\right)\underline{C}_i \geq \underline{C}_1$ which contradicts our assumption of sorted $\underline{C}_i$s, thus proving the necessity part. □

Lemma 2.2 characterizes the conditions under which some strategies of the adversary are dominated by a linear combination of other strategies. This lemma helps us find a critical index to compute the NE.

Lemma 2.2. *If* $\sum\limits_{j=1}^{k}\frac{\underline{C}_j-\underline{C}_k}{\overline{\delta}_j\underline{C}_j}>1$, *then the adversary's strategies* $l\geq k$ *are strictly dominated by a mixed strategy that is composed of pure strategies* j *for* $1\leq j<k$, *i.e., there exist* $\lambda_i\geq 0$, $1\leq i\leq k-1$ *with* $\sum\limits_{i=1}^{k-1}\lambda_i=1$ *such that:*

$$\lambda_1\begin{bmatrix}\left(1-\overline{\delta}_1\right)\underline{C}_1\\ \underline{C}_1\\ \underline{C}_1\\ \vdots\\ \underline{C}_1\end{bmatrix}+\lambda_2\begin{bmatrix}\underline{C}_2\\ \left(1-\overline{\delta}_2\right)\underline{C}_2\\ \underline{C}_2\\ \vdots\\ \underline{C}_2\end{bmatrix}+\cdots+\lambda_{k-1}\begin{bmatrix}\underline{C}_{k-1}\\ \vdots\\ \left(1-\overline{\delta}_{k-1}\right)\underline{C}_{k-1}\\ \vdots\\ \underline{C}_{k-1}\end{bmatrix}>\begin{bmatrix}\underline{C}_l\\ \vdots\\ \vdots\\ \left(1-\overline{\delta}_l\right)\underline{C}_l\\ \vdots\\ \underline{C}_l\end{bmatrix}.$$

Proof. The inequality holds for rows $r\geq k$ because $\underline{C}_i$s are sorted, i.e., $\sum_{j=1}^{k-1}\lambda_j\underline{C}_j>\underline{C}_k$.

For rows $r<k$, consider the assumption $\sum\limits_{j=1}^{k-1}\frac{\underline{C}_j-\underline{C}_k}{\overline{\delta}_j\underline{C}_j}>1$. After some algebraic manipulations, this inequality can be rewritten as:

$$\frac{\left(1-\overline{\delta}_r\right)\underline{C}_r}{\overline{\delta}_r\underline{C}_r\sum\limits_{m=1}^{k-1}\frac{1}{\overline{\delta}_m\underline{C}_m}}+\sum_{j=1,j\neq r}^{k-1}\frac{\underline{C}_j}{\overline{\delta}_j\underline{C}_j\sum\limits_{m=1}^{k-1}\frac{1}{\overline{\delta}_m\underline{C}_m}}>\underline{C}_k.$$

Setting $\lambda_j=\frac{1}{\overline{\delta}_j\underline{C}_j\sum\limits_{m=1}^{k-1}\frac{1}{\overline{\delta}_m\underline{C}_m}}$ gives the result as:

$$\lambda_r\left(1-\overline{\delta}_r\right)\underline{C}_r+\sum_{j=1,j\neq r}^{k-1}\lambda_j\underline{C}_j>\underline{C}_k>\underline{C}_l.$$

□

Lemma 2.3 complements lemma 2.2 in characterizing the sites that should be in the mixed Nash equilibrium.

Lemma 2.3. *If* $\sum\limits_{j=1}^{k}\frac{\underline{C}_j-\underline{C}_k}{\overline{\delta}_j\underline{C}_j}<1$*, any strategy profile with* $x_k=0$ *is not a Nash equilibrium.*

Proof. By contradiction. Suppose the Nash equilibrium holds with $x_k=0$. If $y_k=0$, consider a critical $k^*\geq k$ such that $\sum\limits_{j=1}^{k^*}\frac{\underline{C}_j-\underline{C}_{k^*}}{\overline{\delta}_j\underline{C}_j}<1<\sum\limits_{j=1}^{k^*+1}\frac{\underline{C}_j-\underline{C}_{k^*+1}}{\overline{\delta}_j\underline{C}_j}$. Using Lemma 1, we can conclude that both players are playing a mixed strategy. Moreover using lemma 2 we have: $x_j=0, y_j=0$, $\forall j>k^*$. Therefore the adversary is indifferent towards his choices $i=1,...,k^*, i\neq k$, in other words: $\left(1-\overline{\delta}_1x_1\right)\underline{C}_1=\left(1-\overline{\delta}_2x_2\right)\underline{C}_2=...=\left(1-\overline{\delta}_{k-1}x_{k-1}\right)\underline{C}_{k-1}=\left(1-\overline{\delta}_{k+1}x_{k+1}\right)\underline{C}_{k+1}=...=\left(1-\overline{\delta}_{k^*}x_{k^*}\right)\underline{C}_{k^*}$. Solving these equations along with the equation $\sum\limits_{j=1,j\neq k}^{k^*}x_j=1$ yields:

$$x_{k^*}=\frac{1-\sum\limits_{j=1,j\neq k}^{k^*}\frac{\underline{C}_j-\underline{C}_{k^*}}{\overline{\delta}_j\underline{C}_j}}{\overline{\delta}_{k^*}\underline{C}_{k^*}\sum\limits_{j=1,j\neq k}^{k^*}\frac{1}{\overline{\delta}_j\underline{C}_j}}.$$

Since $\sum\limits_{j=1}^{k^*}\frac{\underline{C}_j-\underline{C}_{k^*}}{\overline{\delta}_j\underline{C}_j}<1$ and $\underline{C}_{k^*}\leq\underline{C}_k$, the following inequality holds

$$\sum_{j=1,j\neq k}^{k^*}\frac{\underline{C}_j-\underline{C}_k}{\overline{\delta}_j\underline{C}_j}<1,$$

which could be rewritten as:

$$\sum_{j=1,j\neq k}^{k^*} \frac{\underline{C}_j - \underline{C}_{k^*} + (\underline{C}_{k^*} - \underline{C}_k)}{\overline{\delta}_j \underline{C}_j} < 1.$$

This further simplifies to

$$(\underline{C}_{k^*} - \underline{C}_k) < \frac{1 - \sum\limits_{j=1,j\neq k}^{k^*} \frac{\underline{C}_j - \underline{C}_{k^*}}{\overline{\delta}_j \underline{C}_j}}{\sum\limits_{j=1,j\neq k}^{k^*} \frac{1}{\overline{\delta}_j \underline{C}_j}} = \overline{\delta}_{k^*} \underline{C}_{k^*} x_{k^*},$$

giving $\left(1 - \overline{\delta}_{k^*} x_{k^*}\right) \underline{C}_{k^*} < \underline{C}_k$. Therefore, the adversary can strictly improve his/her payoff by increasing y_k to 1. Hence $y_k = 1$ should hold. Now the defender can strictly increase his/her payoff by increasing x_k to 1. This is in contradiction with our assumption of $x_k = 0$ being in a Nash equilibrium. □

Theorem 2.1 states the uniqueness of the Nash equilibrium.

Theorem 2.1. *The maximum damage game has a unique NE.*

Proof. Consider a critical k^* such that $\sum\limits_{j=1}^{k^*} \frac{\underline{C}_j - \underline{C}_{k^*}}{\overline{\delta}_j \underline{C}_j} < 1 < \sum\limits_{j=1}^{k^*+1} \frac{\underline{C}_j - \underline{C}_{k^*+1}}{\overline{\delta}_j \underline{C}_j}$, if $k^* = 1$ then lemma 1 and lemma 2 imply that the game has a unique pure strategy Nash equilibrium. If $k^* \geq 2$, then using lemma 2 and lemma 3, the mixed strategy Nash equilibrium is determined by solving the following systems of equations:

System 1:

$$\left(1 - \overline{\delta}_1 x_1\right) \underline{C}_1 = \left(1 - \overline{\delta}_2 x_2\right) \underline{C}_2 = ... = \left(1 - \overline{\delta}_{k^*} x_{k^*}\right) \underline{C}_{k^*},$$

$$\sum_{j=1}^{k^*} x_j = 1.$$

System 2:

$$-\left(1 - \underline{\delta}_1\right) \overline{C}_1 y_1 - \sum_{j=1,j\neq 1}^{k^*} \overline{C}_j y_j = ... = -\left(1 - \underline{\delta}_{k^*}\right) \overline{C}_{k^*} y_{k^*} - \sum_{j=1,j\neq k^*}^{k^*} \overline{C}_j y_j,$$

$$\sum_{j=1}^{k^*} y_i = 1.$$

Both systems have unique solutions. □

2.2.2 Model 2: Infiltration Game

In this model, the adversary wants to infiltrate, i.e., the adversary values all sites equally. Let $\tilde{C}$ with $\tilde{C} \in \left[\underline{C}, \overline{C}\right]$ denote this common value. Assume that the defender knows the intention of the adversary. Hence the expected payoff to the adversary under the mixed strategy pair $(\mathbf{x}, \mathbf{y})$ of the defender and the adversary, respectively, is:

$$u_A^2\left(\mathbf{x}, \mathbf{y}\right) = \sum_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i\right) \tilde{C} y_i.$$

Following the robust approach, the adversary seeks to maximize the minimum expected damage. Using the same reasoning as in Model 1, the adversary's best response to the defender's mixed strategy $\mathbf{x}$ is:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} \left(\sum_{i=1}^{N} \left(1 - \overline{\delta}_i x_i\right) \underline{C}\, y_i \right).$$

Similarly, the defender wants to minimize the maximum expected damage, therefore her best response is:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left(\sum_{i=1}^{N} \left(1 - \underline{\delta}_i x_i\right) \overline{C}_i\, y_i \right).$$

The following matrix demonstrates the payoff to both players:

$$\begin{array}{c|c}
i \setminus j & \begin{array}{cccc} 1 & 2 & \cdots & N \end{array} \\
\begin{array}{c} 1 \\ 2 \\ \vdots \\ N \end{array} &
\left(\begin{array}{cccc}
-(1-\underline{\delta}_1)\overline{C}_1, (1-\overline{\delta}_1)\underline{C} & -\overline{C}_2, \underline{C} & \cdots & -\overline{C}_N, \underline{C} \\
-\overline{C}_1, \underline{C} & -(1-\underline{\delta}_2)\overline{C}_2, (1-\overline{\delta}_2)\underline{C} & \cdots & -\overline{C}_N, \underline{C} \\
\vdots & \vdots & \ddots & \vdots \\
-\overline{C}_1, \underline{C} & -\overline{C}_2, \underline{C} & \cdots & -(1-\underline{\delta}_N)\overline{C}_N, (1-\overline{\delta}_N)\underline{C}
\end{array}\right).
\end{array}$$

Note again that, this is a non-zero sum game. It is obvious that the infiltration game does not have a pure NE. Lemma 4 uses this fact to characterize the strategies that take part in the mixed strategy NE. Specifically, this lemma proves that all of the sites will take part in the mixed strategy NE.

Lemma 2.4. *For the infiltration game, any strategy profile with $x_k = 0$ for some $1 \leq k \leq N$ is not a Nash equilibrium.*

Proof. By contradiction. Clearly, such a game does not have a pure Nash equilibrium. Suppose that there is a Nash equilibrium with $x_k = 0$ and $x_j > 0 \, \forall j \neq k$. The mixed strategy of the defender is determined by solving the following system of equations:

$$\left(1-\overline{\delta}_1 x_1\right)\underline{C} = \left(1-\overline{\delta}_2 x_2\right)\underline{C} = \ldots = \left(1-\overline{\delta}_{k-1} x_{k-1}\right)\underline{C} = \left(1-\overline{\delta}_{k+1} x_{k+1}\right)\underline{C} = \ldots = \left(1-\overline{\delta}_N x_N\right)\underline{C},$$

which along with $\sum\limits_{j=1,\, j\neq k}^{N} x_j = 1$, gives:

$$x_j = \frac{\frac{1}{\overline{\delta}_j}}{\left(\sum\limits_{i=1,\, i\neq k}^{N} \frac{1}{\overline{\delta}_i}\right)} \forall j \neq k.$$

Since $x_j > 0$ for $j \neq k$, this implies that:

$$\left(1-\overline{\delta}_j x_j\right)\underline{C} = \left(1 - \frac{1}{\left(\sum\limits_{i=1,\, i\neq k}^{N} \frac{1}{\overline{\delta}_i}\right)}\right)\underline{C} < \underline{C},$$

with the right hand side corresponding to the adversary's payoff if an attack targets node k. Therefore the adversary can strictly increase his payoff by increasing y_k to 1. The defender can also improve her payoff by setting $x_k = 1$, however, this contradicts our assumption that the current set of strategies is a NE. □

Theorem 2.2. *The infiltration game has a unique NE.*

Proof. Lemma 4 implies that all of the sites should be involved in the mixed strategy NE. Therefore, mixed strategy NE is the unique solution to the following system of $2N$ linearly independent equations with $2N$ unknowns:

$$\left(1-\overline{\delta}_1 x_1\right)\underline{C} = \left(1-\overline{\delta}_2 x_2\right)\underline{C} = \ldots = \left(1-\overline{\delta}_N x_N\right)\underline{C}, \tag{2.1}$$

$$\sum_{i=1}^{N} x_i = 1, \tag{2.2}$$

$$-\left(1-\underline{\delta}_1\right)\overline{C}_1 y_1-\sum_{j=1,j\neq 1}^{N}\overline{C}_j y_j=...=-\left(1-\underline{\delta}_N\right)\overline{C}_N y_N-\sum_{j=1,j\neq N}^{N}\overline{C}_j y_j, \qquad (2.3)$$

$$\sum_{i=1}^{N} y_i=1. \qquad (2.4)$$

□

Remark 2.1. *Clearly, $\underline{C}$ can be eliminated in Equation* (2.1)*. Therefore the Nash equilibrium does not depend on the value of $\overline{C}$ or $\underline{C}$ (upper and lower bounds on the infiltrating adversary's valuation). This is natural because for the infiltrating adversary all sites are equal and the value of these sites does not affect his behavior. Moreover, the defender has her own valuation of the sites, therefore the value of $\overline{C}$ or $\underline{C}$ does not affect her behavior either. Hence it is natural that the NE does not depend on the value of $\overline{C}$ or $\underline{C}$. However, this was not the case in the previous infrastructure security game models. This is mainly due to the zero-sum nature of the previous models [35].*

2.2.3 Model 3: Security Game with Private Information

In this model, we assume that the defender does not know about the intention of the adversary (inflict maximum damage or infiltrate). We use a Bayesian-robust approach to model this game. Meaning that all players use a robust approach to contend with uncertainty of $\tilde{C}, \tilde{C}_i$ and $\tilde{\delta}_i$, however, the defender uses a Bayesian approach to contend with the information asymmetry. In other words, the defender knows that the adversary attempts to inflict maximum damage with probability q, and he attempts infiltration with probability $1-q$. Using the Bayesian robust approach and the definition of NE, the following conditions should be satisfied:

$$\mathbf{y}^{1*}=\arg\max_{\mathbf{y}^1}\min_{\substack{\tilde{\delta}_i\in\left[\underline{\delta}_i,\overline{\delta}_i\right],\\ \tilde{C}_i\in\left[\underline{C}_i,\overline{C}_i\right]}}\left(\sum_{i=1}^{N}\left(1-\tilde{\delta}_i x_i^*\right)\tilde{C}_i y_i^1\right),\ \ \sum_{i=1}^{N} y_i^{1*}=1,\ \ y_i^{1*}\geq 0,$$

$$\mathbf{y}^{2*} = \arg\max_{\mathbf{y}^2} \min_{\substack{\tilde{\delta}_i \in \left[\underline{\delta}_i, \overline{\delta}_i\right], \\ \tilde{C} \in \left[\underline{C}, \overline{C}\right]}} \left(\sum_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i^*\right) \tilde{C} y_i^2 \right), \quad \sum_{i=1}^{N} y_i^{2*} = 1, \ \ y_i^{2*} \geq 0,$$

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \max_{\substack{\tilde{\delta}_i \in \left[\underline{\delta}_i, \overline{\delta}_i\right], \\ \tilde{C}_i \in \left[\underline{C}_i, \overline{C}_i\right]}} \left(q \sum_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i\right) \tilde{C}_i y_i^{1*} + (1-q) \sum_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i\right) \tilde{C}_i y_i^{2*} \right),$$

$$\sum_{i=1}^{N} x_i^* = 1, \ \ x_i^* \geq 0,$$

where $\mathbf{y}^1$ is the mixed strategy of the maximum damage adversary, $\mathbf{y}^2$ is the mixed strategy of the infiltrating adversary, and $\mathbf{x}$ is the defender's mixed strategy as before. Note that $\left(\sum\limits_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i^*\right) \tilde{C}_i y_i^1 \right)$ is minimized at $\tilde{\delta}_i = \overline{\delta}_i$ and $\tilde{C}_i = \underline{C}_i$, $\left(\sum\limits_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i^*\right) \tilde{C} y_i^2 \right)$ is minimized at $\tilde{\delta}_i = \overline{\delta}_i$ and $\tilde{C} = \underline{C}$, finally $\left(q \sum\limits_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i\right) \tilde{C}_i y_i^{1*} + (1-q) \sum\limits_{i=1}^{N} \left(1 - \tilde{\delta}_i x_i\right) C_i y_i^{2*} \right)$ is maximized at $\tilde{\delta}_i = \underline{\delta}_i$ and $\tilde{C}_i = \overline{C}_i$. Thus

$$\mathbf{y}^{1*} = \arg\max_{\mathbf{y}^1} \left(\sum_{i=1}^{N} \left(1 - \overline{\delta}_i x_i^*\right) \underline{C}_i y_i^1 \right),$$

$$\mathbf{y}^{2*} = \arg\max_{\mathbf{y}^2} \left(\sum_{i=1}^{N} \left(1 - \overline{\delta}_i x_i^*\right) \underline{C} y_i^2 \right),$$

and

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left(q \sum_{i=1}^{N} \left(1 - \underline{\delta}_i x_i\right) \overline{C}_i y_i^{1*} + (1-q) \sum_{i=1}^{N} \left(1 - \underline{\delta}_i x_i\right) \overline{C}_i y_i^{2*} \right).$$

This optimization problem can be solved by direct application of Karush–Kuhn–Tucker conditions [78]. However, more insight can be gained by analyzing this game. The following theorem characterizes the Nash equilibrium for the security game with private information.

Theorem 2.3. *The following strategy profile is a Nash equilibrium for the security game with private information. Let k be an integer such that $\phi_k \leq 1 < \phi_{k+1}$ where $\phi_i = \sum\limits_{j=1}^{i} \frac{\underline{C}_j - \underline{C}_i}{\overline{\delta}_j \underline{C}_j}$, and m be an integer such that $\psi_{m-1} < q \leq \psi_m$ where $\psi_i = \frac{\left(\sum\limits_{j=1}^{i} \frac{1}{\underline{\delta}_j \overline{C}_j} \right)}{\left(\sum\limits_{j=1}^{N} \frac{1}{\underline{\delta}_j \overline{C}_j} \right)}$.*

If $m \leq k$ then

$$x_j^* = \begin{cases} \dfrac{1+\sum\limits_{j=1}^{m} \frac{\underline{C}_1-\underline{C}_j}{\underline{C}_j\overline{\delta}_j} + \frac{\underline{C}_1-\underline{C}_m}{\underline{C}_m} \sum\limits_{j=m+1}^{N} \frac{1}{\overline{\delta}_j}}{\left(\sum\limits_{j=1}^{m} \frac{\underline{C}_1\overline{\delta}_1}{\underline{C}_j\overline{\delta}_j} + \frac{\underline{C}_1\overline{\delta}_1}{\underline{C}_m} \sum\limits_{j=m+1}^{N} \frac{1}{\overline{\delta}_j}\right)}, & j = 1, \\ \\ \dfrac{\underline{C}_1\overline{\delta}_1}{\underline{C}_j\overline{\delta}_j} x_1 - \dfrac{\underline{C}_1-\underline{C}_j}{\underline{C}_j\overline{\delta}_j}, & 2 \leq j \leq m, \\ \\ \frac{x_m\overline{\delta}_m}{\overline{\delta}_j}, & j > m, \end{cases} \tag{2.5}$$

$$y_j^{*1} = \begin{cases} \dfrac{1}{q\left(\underline{\delta}_j\overline{C}_j\right)\left(\sum\limits_{i=1}^{N} \frac{1}{\underline{\delta}_i\overline{C}_i}\right)}, & j < m, \\ \\ 1 - \dfrac{\sum_{i=1}^{m-1} \frac{1}{\underline{\delta}_i\overline{C}_i}}{q\sum_{i=1}^{N} \frac{1}{\underline{\delta}_i\overline{C}_i}}, & j = m, \\ \\ 0, & j > m, \end{cases} \tag{2.6}$$

and

$$y_j^{*2} = \begin{cases} 0, & j < m, \\ \dfrac{\left(\sum\limits_{j=1}^{m} \frac{1}{\underline{\delta}_j\overline{C}_j}\right) - q\left(\sum\limits_{j=1}^{N} \frac{1}{\underline{\delta}_j\overline{C}_j}\right)}{(1-q)\left(\sum\limits_{j=1}^{N} \frac{1}{\underline{\delta}_j\overline{C}_j}\right)}, & j = m, \\ \\ \dfrac{1}{(1-q)\left(\underline{\delta}_j\overline{C}_j\right)\left(\sum\limits_{i=1}^{N} \frac{1}{\underline{\delta}_i\overline{C}_i}\right)}, & j > m. \end{cases} \tag{2.7}$$

If $m > k$ then

$$x_j^* = \begin{cases} \frac{\frac{1}{\overline{\delta}_j \underline{C}_j}}{\sum\limits_{i=1}^{k} \frac{1}{\overline{\delta}_i \underline{C}_i}} \left(1 - \sum\limits_{i=1}^{k} \frac{\underline{C}_i - \underline{C}_j}{\overline{\delta}_i \underline{C}_i}\right), & j \leq k, \\ 0, & j > k, \end{cases} \tag{2.8}$$

$$y_j^{*1} = \begin{cases} \frac{\frac{1}{\underline{\delta}_j \overline{C}_j}}{\sum\limits_{l=1}^{k} \frac{1}{\underline{\delta}_l \overline{C}_l}}, & j \leq k, \\ 0, & j > k, \end{cases} \tag{2.9}$$

$$y_i^{*2} < \frac{q}{(1-q)} \left(\frac{\frac{1}{\underline{\delta}_i \overline{C}_i}}{\sum\limits_{l=1}^{k} \frac{1}{\underline{\delta}_l \overline{C}_l}}, \right) \forall i > k, \quad \sum_{j=k+1}^{N} y_j^{*2} = 1. \tag{2.10}$$

Proof. See Appendix A.1 □

Remark 2.2. *Similar to the infiltration game, also in this game the Nash equilibrium does not depend on the value of* $\overline{C}$ *or* $\underline{C}$ *(upper and lower bounds on the infiltrating adversary's valuation).*

Remark 2.3. *For the second case, i.e.,* $m > k$*, similar to [35] there is a continuum of NE strategies for the infiltrating adversary.*

2.3 Numerical Analysis

In this section, we apply our approach to real data from [139] which provides estimates of the expected annual terrorism losses for the 10 most valuable urban areas of the United States. We use the proposed robust game model to allocate defensive resources among these urban areas. The data is presented in Table 2.1. In this table, three aspects of the expected damage have been estimated: monetary value (represented by expected property loss), mortality value (represented by total number of fatalities and injuries) and political value (represented by total air departures from major and minor airports). In the following subsections, each one of these dimensions will be investigated individually.

Table 2.1: Expected damage data for 10 urban areas with highest losses

Urban Area	Expected property loss ($million)	Expected Fatalities & Injuries	Air Departures (Major & Minor Airports)
New York (NY)	413	5350	23599
Chicago (CH)	115	1212	39949
San Francisco (SF)	57	472	19142
Washington, DC-MD-VA-WV (WDC)	36	681	17253
Los Angeles-Long Beach (LA)	34	402	28816
Philadelphia, PA-NJ (PHL)	21	199	13640
Boston, MA-NH (BSTN)	18	225	11625
Houston (HSTN)	11	160	20979
Newark (NW)	7.3	74	12827
Seattle-Bellevue-Everett (STL)	6.7	88	13578
Total	719	8863	201408

2.3.1 Analysis for Monetary Value Data

In this section, we perform the analysis based on the monetary data for each urban area. We study how the defender's strategy is affected by the probability of a maximum damage type adversary, q. This probability is an indicator of the uncertainty over type of the adversary i.e. maximum damage or infiltrating. We also study the effect of this probability on the expected property loss at each urban area.

For the probability of detecting an attack set to 0.9, i.e., $\delta = 0.9$, Figure 2.1 displays how defensive strategy may vary among sites for different values of q, and also Figure 2.2 illustrates how the expected property loss at each urban area may vary over q. Figure 2.1 shows that the defensive resources are evenly distributed for low values of q. This is due to the fact that for low values of q, the defender effectively

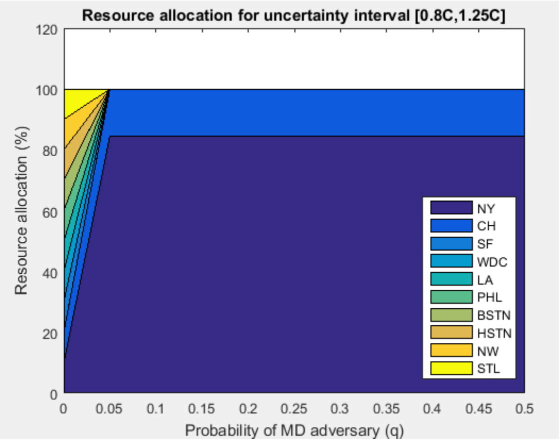


Figure 2.1: Allocation of defensive resources for monetary data

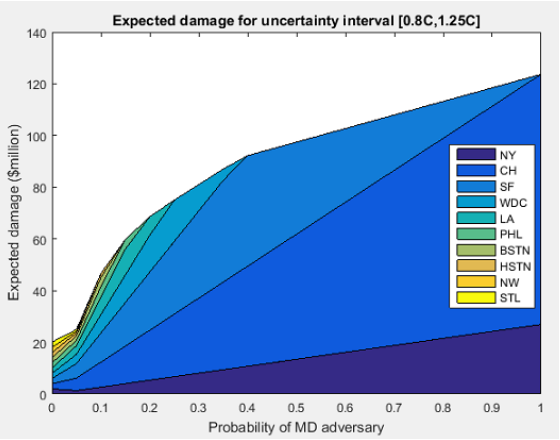


Figure 2.2: Distribution of expected property loss for monetary data

plays the game against an infiltration type attacker, hence the defense resources are distributed proportionally with respect to the detection probabilities at different urban areas. However, since we have assumed the same detection probability for each area as $\delta_i = \delta = 0.9$, the defensive resources are evenly distributed. As q increases beyond a certain level, more resources are allocated to NY and CH, which are the areas with highest property loss, and fewer resources are allocated to other areas. This shift in resource allocation happens around $q = 0.05$, that corresponds to a threshold point. As q increases further, the game is effectively turned into a maximum damage game and all defensive resources are distributed between two areas, namely NY and CH. Further increase in q does not change the allocation of resources. Figure 2.2 shows distribution of the expected property loss at each urban area as a function of q. As seen in this figure, for low values of q, since the attacks are distributed among all areas and the defensive strategy is also to distribute the defensive resources evenly among all areas, the expected damage is roughly the same for all areas. As the value of q increases beyond a certain level, the defensive strategy changes to play the MD game. As the value of q increases further, the expected damage to important areas (such as NY, CH and SF) increase and the expected damage to other areas decrease.

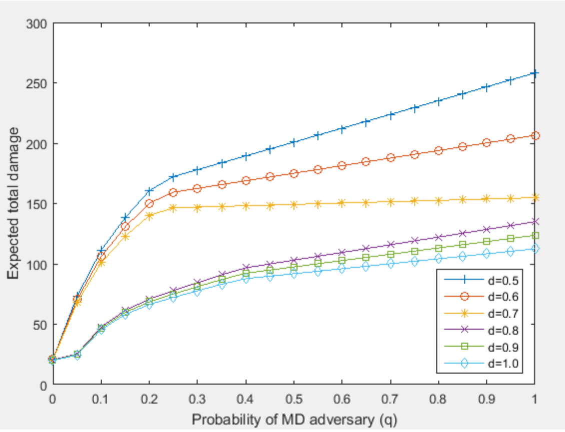


Figure 2.3: Expected property loss over various detection probabilities

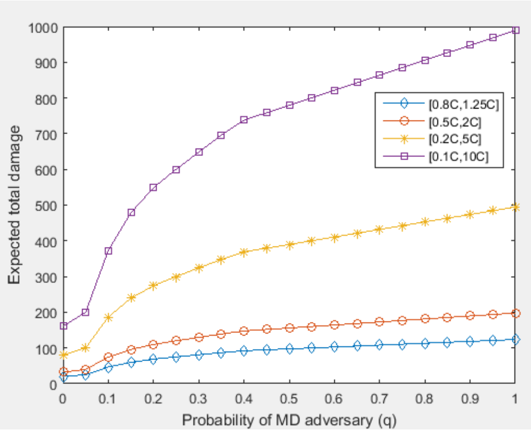


Figure 2.4: Expected property loss over various uncertainty ranges

This effect is observed because the adversary's attacks get more targeted towards high impact areas as q increases. Figure 2.3 shows the expected total property loss as a function of q for different values of probability of detection, δ. As seen in this figure, the damage is higher for smaller values of δ and the difference increases as q increases. This is due to the increasing importance of the efficiency of defensive resources as the adversary targets high impact areas with higher probability.

Figure 2.4 displays how expected total property loss changes as a function of q over the various uncertainty ranges. As seen in this figure, for wider ranges of uncertainty, the expected total damage is higher than scenarios with smaller uncertainty ranges.

2.3.2 Analysis for Mortality Value Data

In this section, we perform the robust game analysis based on the mortality value of each urban area. We study how the defenders strategy is affected by q. For $\delta_i = \delta = 0.9$, Figure 2.5 illustrates how the defensive strategy changes for various values of q. As seen in the figure, for low values of q, because the game is effectively an infiltration game, defensive resources are evenly distributed among urban areas. As q increases beyond a certain level, more resources are allocated to NY and CH,

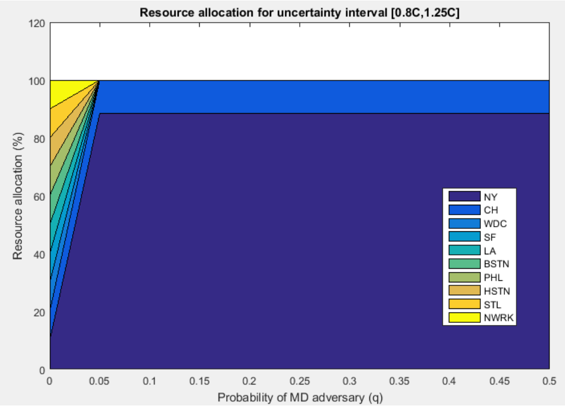


Figure 2.5: Allocation of defensive resources for mortality data

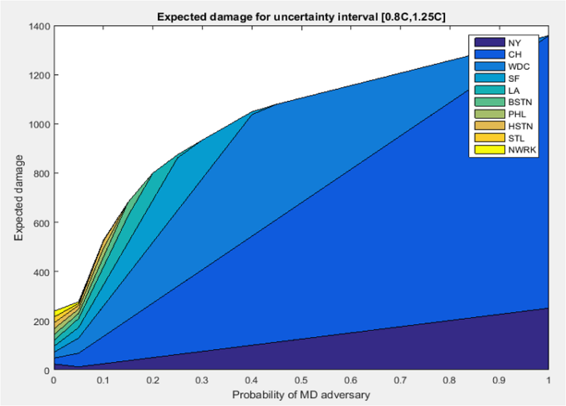


Figure 2.6: Distribution of expected damage for mortality data

which are the areas with highest population density, and fewer resources are allocated to other areas. This shift in resource allocation happens around $q = 0.05$, which corresponds to a threshold point. As q increases further, the game is effectively turned into a MD game and all defensive resources are distributed between two major areas, namely NY and CH. Further increase in q does not change the allocation of resources. Figure 2.6 shows how the expected number of fatalities and injuries at each urban area changes for different values of q. As seen in this figure for low values of q, the expected damage is roughly the same for all areas. As the value of q increases beyond a certain level, the defensive strategy changes to play the MD game. As the value of q increases further, the expected damage on important areas (such as NY, CH and SF) increases and the expected damage for other areas decrease. This is due to the fact that as q increases, the adversary targets more important areas with higher probability. Figure 2.7 displays the expected total damage as a function of q for various values of δ. As seen in this figure, the damage is higher for smaller value of δ and the difference increases as q increases. Figure 2.8 illustrates how the expected total damage changes as a function of q for various uncertainty ranges. As seen in this figure, for wider ranges of uncertainty the expected total damage is higher than

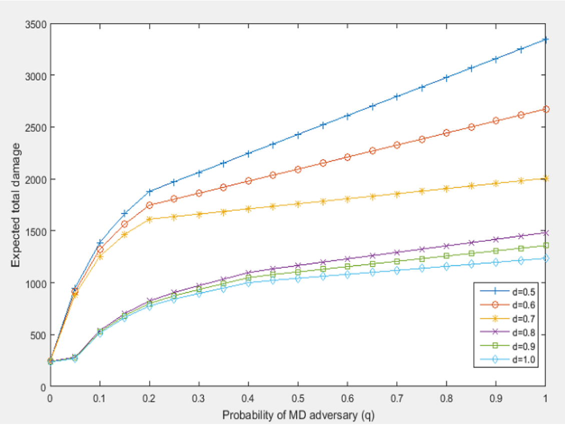


Figure 2.7: Expected total damage for various detection probabilities

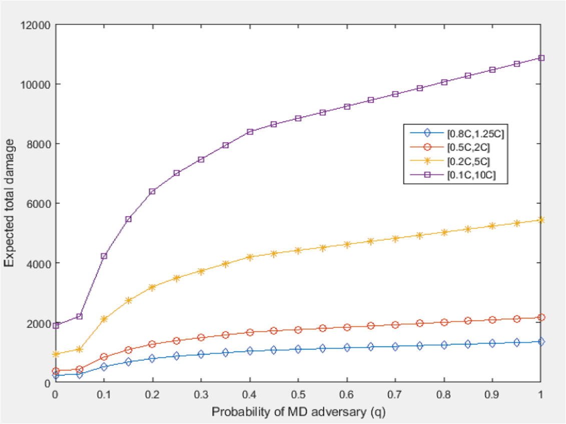


Figure 2.8: Expected total damage for various uncertainty ranges

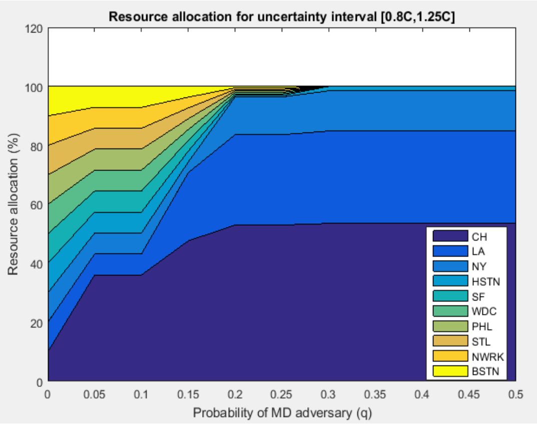


Figure 2.9: Allocation of defensive resources for political data

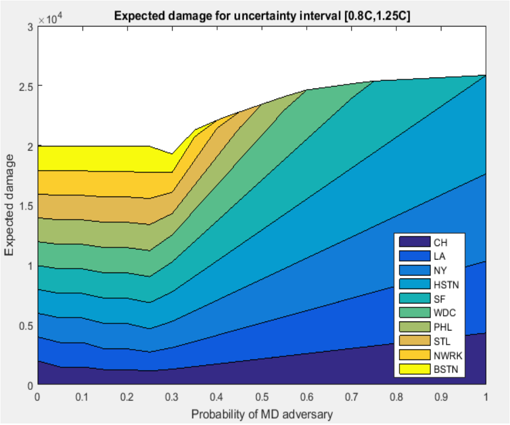


Figure 2.10: Distribution of expected damage for political data

scenarios with smaller uncertainty ranges.

2.3.3 Analysis for Political Value Data

In this section, we perform the analysis based on the political value of each urban area. We study the effect of q on the defenders strategy. For $\delta_i = \delta = 0.9$, Figure 2.9 shows how the defensive strategy changes for different values of q, and also Figure 2.10 shows how the expected damage on each urban area may vary for different values of q.

As seen in Figure 2.9, for low values of q, defensive resources are evenly distributed. As q increases beyond a certain level, more resources are allocated to CH, LA and NY which are the most important areas in terms of political value, and fewer resources are allocated to other areas. This shift in resource allocation happens around $q = 0.05$, which corresponds to a threshold point. Further shifts in defensive strategy happen at around $q = 0.1$, $q = 0.15$ and $q = 0.2$. At each of these threshold points, more defensive resources are assigned to the most important areas and fewer resources are allocated to other areas. As q increases further, the game is effectively a MD game and all defensive resources are distributed among four areas, namely CH, LA, NY and HSTN. After a certain point, further increase in q does not change the allocation of resources.

Figure 2.10 displays the average expected damage to each urban area as a function of q, for low values of q, the expected damage is roughly the same for all areas. As the value of q increases further, both defender and the adversary focus more on the most important areas, therefore the expected damage on important areas (such as CH, LA, NY and HSTN) increases and the expected damage for other areas decrease.

Figure 2.11 shows the expected total damage as a function of q for various values of δ. As seen in this figure, the damage is higher for smaller value of δ and the difference increases as q increases.

Figure 2.12 illustrates how the expected total damage changes as a function of q for various uncertainty ranges. As seen in this figure, for wider ranges of uncertainty the expected total damage is higher than scenarios with smaller uncertainty ranges.

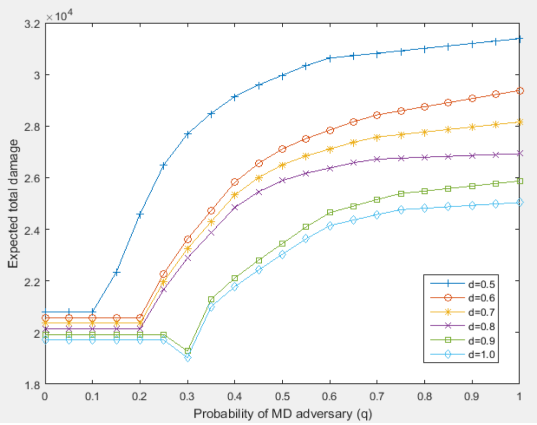


Figure 2.11: Expected total damage for various probability of detection

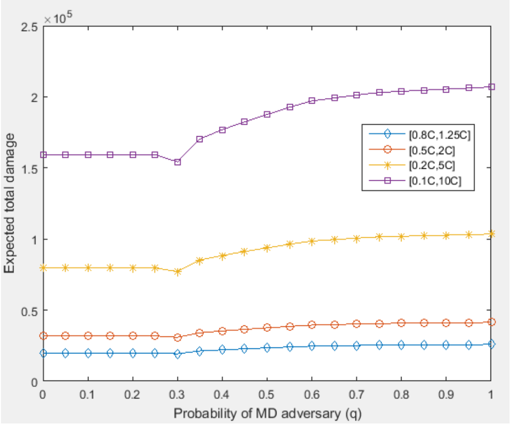


Figure 2.12: Expected total damage for various uncertainty ranges

Chapter 3

A Two-Stage Invest-Defend Game: Balancing Strategic and Operational Decisions

3.1 Introduction and Literature Review

Protecting infrastructures against disruptions involves decision making at different levels in an organization's hierarchy: strategic and operational decisions. However, most of the research papers only focus on either purely strategic decisions [63, 107] or purely operational decisions [16, 35]. Baykal-Gürsoy *et al.* [16] consider an infrastructure containing multiple sites with a single defender and a single attacker. Both players make operational decisions of which site to defend/attack. They assume that the detection probabilities are fixed and cannot be changed. Garnaev and Baykal-Gürsoy [35] study operational decision of which sites to defend/attack with consideration given to the uncertainty of the attack type. Shan and Zhuang [131] study defender's operational decisions such as container screening rate to deter nuclear smuggling. They show how the inspection rates should be modified in presence of non-credible retaliation threat to deter nuclear smuggling. There are other papers in the literature that focus on purely operational decisions such as allocating defenders [38], patrolling [118, 134] and scheduling [137]. The literature on strategic decisions include Nikoofal and Zhuang [107] who consider a game in which a defender makes strategic decision of allocating resources to harden a set of target sites so as to minimize the maximum damage of an attack. Hausken and Zhuang [63] analyze a two-stage resource allocation game between a government and a terrorist. In this Stackelberg game, the government moves first and allocates its resources between attacking to downgrade the terrorist's resources and defending against the terrorist attack. Then, the terrorist allocates his/her resources between attack and defend options. Other papers study strategic decision of resource assignment to protect targets against attacks [47, 130, 145]. In the context of making strategic decisions in security, several papers consider multi-period models where strategic decisions are made throughout multiple periods [63, 69, 132, 146]. These multi-period security games consider the same strategic decisions throughout multiple periods, and hence they focus on the effect of timing of these strategic decisions.

Even though most papers study models with purely strategic or purely operational decisions, these decisions influence each other. For instance, installing a CCTV camera in a certain area might render patrolling that area unnecessary. Or different allocations of metal detectors and screening systems to target sites may affect optimal scheduling of patrol units in those targets. Moreover, investing on a new technology to enhance security of a certain target site may reduce its probability of being attacked and affect the optimal probability of defending that target. Therefore, considering strategic and operational decisions in the same model would yield a more realistic analysis.

3.2 Proposed Model

We consider a two-stage invest-defend game between a single defender and a single attacker. We assume that both players are fully rational and aim to maximize their own payoff values. In the first stage, both the defender and the adversary simultaneously make strategic decision of investing on targets to change the detection probabilities in their own favor and then in the second stage the defender and the adversary simultaneously make operational decision of selecting which target to assign defender/attacker i.e. which target to defend/attack. In other words, in the second stage, the defender and the attacker play a matrix game where they have to decide which site to defend or attack.

Each target i has a value of C_i for the defender. This value could be determined by occupancy levels or any other valuation criterion e.g. monetary or political value. We assume that the adversary's target valuations are the same as the defender's valuations as in [45, 119, 129] and [130]. While we acknowledge that the adversary may value the targets differently, using the same target valuations for the adversary results in a game in which the players' payoffs are in opposite direction and it is useful in a worst case analysis. Wang and Bier [138] use multi-attribute utility functions to model the attacker's preferences. Robust games against an attacker with private target preference have been studied by [107] and [142].

The second stage game is a matrix game where players make operational decision of choosing which target to defend/attack. If the defender defends site i and the adversary attacks site $j, j \neq i$, a successful attack on site j will be launched. Therefore payoff of the defender will be $-C_j$ and the adversary will get a payoff of C_j. However, if both players choose the same site i, the attacker will be detected (and thwarted) with probability δ_i. We assume that the attacker suffers a disutility of P in case of a failed attack. Therefore the defender will get a payoff of $-\left(1-\delta_j\right)C_j$ and the adversary will get $(1-\delta_j)\,C_j - \delta_j P$. This means that even when both rivals are at the same site, there is a probability that the defender may not detect the adversary. Other studies have also modeled detection in different contexts such as deterring smuggling of nuclear weapons through containers [50], detecting concealed targets [84], detecting genuine target from false targets [85] and detecting outcome of attacks [87, 88].

The objective of each player is to maximize his/her own total payoff, which is equal to sum of the payoff from first stage and the payoff from the second stage. If $C_i = C, \;\; \forall i = 1, \ldots, N$, then the second stage game is called Infiltration/Harassment game, otherwise it is called Maximal Damage game. In other words, in the Infiltration/Harassment game, because all targets have the same value, the players do not differentiate the sites from each other and only care about minimizing/maximizing the probability of a successful attack.

The parameters of our model are listed as follows:

- N : number of target sites.
- C_i : value of site i. We can, without loss of generality, assume that C_is are sorted in a decreasing order, i.e., $C_1 \geq C_2 \geq \ldots \geq C_N$. If the site values are not sorted, we can renumber their indices so that they are sorted. This renumbering does not change the problem.
- P : disutility (penalty) of an unsuccessful attack for the attacker.
- A : total budget for defensive investments in the budget constrained model.
- B : total budget for attack investments in the budget constrained model.

Decision variables and functions that use these variables are listed as follows:

- α_i : strategic decision for the defender. Amount of investment on defending site i in the first stage, where $0 \leq \alpha_i < \infty$ for all $i = 1, ..., N$. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_N)$ represent the defensive investment vector.

- β_i : strategic decision for the attacker. Amount of investment on attacking site i in the first stage, where $0 \leq \beta_i < \infty$ for all $i = 1, ..., N$. Let $\boldsymbol{\beta} = (\beta_1, \beta_2, ..., \beta_N)$ represent the attack investment vector.

- $\delta_i(\alpha_i, \beta_i)$: probability of detection at site i in the second stage. If both players select site i, then with probability $\delta_i(\alpha_i, \beta_i)$ the attack will successfully be thwarted and the adversary will be detected; with probability $1 - \delta_i(\alpha_i, \beta_i)$, the attacker will successfully destroy the target. Assume that $\delta_i(\alpha_i, \beta_i)$ is a continuous, strictly increasing and concave function of defensive investments in the first stage i.e. α_i. Also assume that $\delta_i(\alpha_i, \beta_i)$ is a continuous, strictly decreasing and convex function of attack investments in the first stage i.e. β_i.

- $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$: operational decision for the defender. Mixed policy of the defender with x_i denoting the probability of defending site i in the second stage game, where $0 \leq x_i \leq 1$ for all $i = 1, \ldots, N$ and $\sum_{i=1}^{N} x_i = 1$.

- $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$: operational decision for the attacker. Mixed policy of the attacker with y_i denoting the probability of attacking site i in the second stage game, where $0 \leq y_i \leq 1$ for all $i = 1, \ldots, N$ and $\sum_{i=1}^{N} y_i = 1$.

- $u_1^d(\boldsymbol{\alpha})$: first stage payoff of the defender in the unconstrained model, where $u_1^d(\boldsymbol{\alpha}) = -\sum_{i=1}^{N} \alpha_i$.

- $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$: second stage payoff of the defender, we have: $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = -\sum_{i=1}^{N} (C_i(1 - \delta_i(\alpha_i, \beta_i)x_i)y_i)$ [16].

- $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$: total payoff of the defender. In the unconstrained model the defender's total payoff is given by $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_1^d(\boldsymbol{\alpha}) + u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$. In the budget constrained model, the defender's total payoff is given by $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$ with budget constraint i.e. $\sum_{i=1}^{N} \alpha_i \leq A$.

- $u_1^a(\boldsymbol{\beta})$: first stage payoff of the attacker in the unconstrained model, where $u_1^a(\boldsymbol{\beta}) = -\sum_{i=1}^N \beta_i$.

- $u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$: second stage payoff of the attacker, we have: $u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^N \left(C_i(1 - \delta_i(\alpha_i, \beta_i)x_i) - \delta_i(\alpha_i, \beta_i)x_i P\right) y_i$.

- $u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$: total payoff of the attacker. In the unconstrained model the attacker's total payoff is given by $u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_1^a(\boldsymbol{\beta}) + u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$. In the budget constrained model the attacker's total payoff is given by $u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$ with budget constraints i.e. $\sum_{i=1}^N \beta_i \leq B$.

3.3 A Backward Induction Approach to Solve the Two-Stage Invest-Defend Game

We assume that players make their decisions simultaneously at both stages. In other words, at each stage the players will not know about the other player's decision. However, first stage decisions will be revealed to both players at the beginning of the second stage. To solve this game, we use the backward induction method and start from the last stage i.e. the second stage. The second stage game is solved assuming fixed values for the first stage decisions, i.e. $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, and the equilibrium policy of each player in the second stage, $\boldsymbol{x}$ and $\boldsymbol{y}$, are obtained in terms of $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. The second stage solution, $(\boldsymbol{x}, \boldsymbol{y})$, is then used in the first stage game to compute the first stage equilibrium point.

3.3.1 Second Stage Game

At the second stage game, the first stage decisions, i.e., strategic decisions $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, are assumed to be fixed parameters and the second stage decisions, i.e., operational decisions $(\boldsymbol{x}, \boldsymbol{y})$, are made. The following matrix demonstrates the payoff to both players:

$$\begin{array}{c}
i \setminus j \\ 1 \\ 2 \\ \vdots \\ N
\end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & \cdots & N \end{array} \\
\begin{pmatrix}
-(1-\delta_1)C_1, (1-\delta_1)C_1 - \delta_1 P & -C_2, C_2 & \cdots & -C_N, C_N \\
-C_1, C_1 & -(1-\delta_2)C_2, (1-\delta_2)C_2 - \delta_2 P & \cdots & -C_N, C_N \\
\vdots & \vdots & \ddots & \vdots \\
-C_1, C_1 & -C_2, C_2 & \cdots & -(1-\delta_N)C_N, (1-\delta_N)C_N - \delta_N P
\end{pmatrix}
\end{array}.$$

In this matrix, the first element is the payoff to the defender and the second element is the payoff to the attacker. If we assume that $P = 0$, then this matrix game turns into a zero sum game, i.e. $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = -u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$. For this zero-sum game [16] give a unique saddle-point equilibrium. In this section, we extend their result to the case where the attacker suffers a disutility for an unsuccessful attack i.e. $P > 0$.

Theorem 3.1. *The Nash Equilibrium for the second stage game is given in terms of an index $k \in \{1, \ldots, N\}$ such that $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq 1 < \phi_{k+1}(\boldsymbol{\alpha}, \boldsymbol{\beta})$, where $\phi_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is defined as $\phi_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{i} \frac{C_j - C_i}{\delta_j(\alpha_j, \beta_j)(C_j + P)}$ for $i \in 1, \ldots, N$ and $\phi_{N+1}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \infty$. The strategy of the defender is of threshold type:*

$$x_i^* = \begin{cases} \dfrac{\dfrac{1}{\delta_i(\alpha_i, \beta_i)(C_i + P)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)(C_j + P)}} \left(1 - \sum_{j=1}^{k} \frac{C_j - C_i}{\delta_j(\alpha_j, \beta_j)(C_j + P)}\right), & i \leq k, \\ 0, & i > k. \end{cases} \tag{3.1}$$

The strategy of the attacker is also of threshold type:

$$y_i^* = \begin{cases} \dfrac{\dfrac{1}{\delta_i(\alpha_i, \beta_i)C_i}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)C_j}}, & i \leq k, \\ 0, & i > k. \end{cases} \tag{3.2}$$

And the equilibrium payoffs are given as:

$$u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}}, \tag{3.3}$$

$$u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = -\frac{1 - \sum_{j=1}^{k} \dfrac{C_j}{\delta_j(\alpha_j, \beta_j)(C_j + P)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)(C_j + P)}}. \tag{3.4}$$

Proof. See Appendix A.2. □

Remark 3.1. *If $C_1 = \cdots = C_N = C$, i.e. the second stage game is of Infiltration/Harassment type, then the Nash Equilibrium requires the use of all target sites, since $\phi_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = 0, \ \forall i = 1, \ldots, N$, i.e., $k = N$. In fact, in this case the defense and attack probabilities are $1/\delta_i(\alpha_i, \beta_i)$ portion of sum of all $1/\delta_i(\alpha_i, \beta_i)$'s, i.e.,*

$$x_i^* = y_i^* = \frac{M}{\delta_i(\alpha_i, \beta_i)}, \ \ \forall i = 1, \ldots, N, \tag{3.5}$$

and:

$$u_d^2(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = C(M - 1), \tag{3.6}$$

$$u_a^2(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = (C + P)(\frac{C}{C + P} - M), \tag{3.7}$$

where $M = \dfrac{1}{\sum_{j=1}^{N} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}$.

Corollary 3.1. *An increase in attacker's investment in site i, i.e. β_i, leads to an increase in probability of both attacking and defending site i. However, an increase in defender's investment in site i leads to a decrease in probability of both attacking and defending site i.*

Proof. From equation (3.2), an increase in β_i leads to a decrease in $\delta_i(\alpha_i, \beta_i)$. Therefore $1/\delta_i(\alpha_i, \beta_i)$ increases, which leads to an increases in y_i^*. Using equation (3.1), an increase in β_i leads to a decrease in $\delta_i(\alpha_i, \beta_i)$. Therefore $1/\delta_i(\alpha_i, \beta_i)$ increases, which leads to an increases in x_i^*. Similarly we can prove the effect of increasing α_i on x_i^* and y_i^*. □

Remark 3.2. *The effect of an increase in defender's investment in site i, seems counter-intuitive at first. However, it can be explained with intuitive arguments. An increase in defender's investment will lead to a decrease in attack probability, therefore the defender will decrease her defence probability. In other words, knowing that the attacker is less likely to attack a site leads the defender to defend that site with lower probability. In extreme case, if the defender knows that the attacker will never attack site i, then the defender will never defend site i.*

3.3.2 First Stage Game

Knowing the outcome of the second stage, we can immediately write down the payoff functions at the first stage for both players. We consider two models: unconstrained and budget constrained model. In the unconstrained model, there is no budget constraint but the players receive a disutility when investing in the first stage. In the budget constrained model, both players have limited budgets that cannot be exceeded.

Unconstrained Model

In this section, we study the unconstrained model with no budget constraints, but there is investment disutility which is considered in the players' respective payoff functions. Hence, the payoff functions for both players are given as follows:

$$u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_1^d(\boldsymbol{\alpha}) + u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = -\sum_{i=1}^{N} \alpha_i + \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}},$$

$$u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_1^a(\boldsymbol{\beta}) + u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = -\sum_{i=1}^{N} \beta_i - \frac{1 - \sum_{j=1}^{k} \dfrac{C_j}{\delta_j(\alpha_j, \beta_j)(C_j + P)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)(C_j + P)}}.$$

The following lemmas characterize the conditions under which the payoff functions are continuous and concave.

Lemma 3.1. *If $P = 0$ or if the second stage game is of Infiltration/Harassment type, i.e. $C_1 = \cdots = C_N = C$, then $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ and $u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ are continuous in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.*

Proof. See Appendix A.3. □

Lemma 3.2. *If $P = 0$ or if the second stage game is of Infiltration/Harassment type, i.e. $C_1 = \cdots = C_N = C$, then $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ and $u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ are strictly concave in each α_i and β_i, respectively.*

Proof. See Appendix A.4. □

Lemma 3.3. *If the second stage game is of Infiltration/Harassment type, i.e., $C_1 = \cdots = C_N = C$ then $u_d^t(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ and $u_a^t(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ are concave in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively.*

Proof. See Appendix A.5. □

The following theorem characterizes the conditions under which there exist a Nash equilibrium for the two-stage invest-defend game.

Theorem 3.2. *If the second stage game is of Infiltration/Harassment type, i.e., for $C_1 = ... = C_N = C$, then the overall invest-defend game has a Nash Equilibrium.*

Proof. It is easy to confirm that the strategy space for both players is compact and convex (note that investment values are bounded). In lemma 3.1 and lemma 3.3 we have established that the payoff function for both players is continuous and concave with respect to their own strategy. Therefore, applying Debreu's existence theorem (see [27]), there exist at least one Nash equilibrium. □

Remark 3.3. *Proving uniqueness of the Nash equilibrium is challenging, however, based on some numerical experiments, we conjecture that it is true.*

We now consider the following detection probability function:

$$\delta_i(\alpha_i, \beta_i) = \frac{e_i^d \alpha_i + L_i}{e_i^d \alpha_i + e_i^a \beta_i + U_i}, \quad 0 \leq L_i \leq U_i, \quad U_i \neq 0. \tag{3.8}$$

In this formula, parameters $e_i^d > 0$ and $e_i^a > 0$ are investment efficiency factors of site i for the defender and the attacker, respectively. Parameters L_i and U_i are there so that when both investment efforts are zero, we have $0 \leq \delta_i(\alpha_i, \beta_i) \leq 1$. This function satisfies our assumptions for a detection probability function, i.e., it is a continuous, strictly increasing and concave function of defensive investments α_i and it is a continuous, strictly decreasing and convex function of attack investments β_i.

Corollary 3.2. *If the detection probability function is given in equation* (3.8)*, and we have* $\frac{(C+P)\frac{e_i^a}{e_i^d}}{\left(N+\frac{C+P}{C}\sum_{j=1}^N \frac{e_j^a}{e_j^d}\right)^2} \geq \frac{L_i}{e_i^d}$ *and* $\frac{(C+P)\frac{e_i^a}{e_i^d}\frac{C+P}{C}}{\left(N+\frac{C+P}{C}\sum_{j=1}^N \frac{e_j^a}{e_j^d}\right)^2} \geq \frac{U_i-L_i}{e_i^a}$*, and the second stage game is of Infiltration/Harassment type, i.e.* $C_1 = \cdots = C_N = C$*, then the first stage game has a unique closed form solution given by:*

$$\alpha_i^* = \frac{(C+P)\frac{e_i^a}{e_i^d}}{\left(N + \frac{C+P}{C}\sum_{j=1}^N \frac{e_j^a}{e_j^d}\right)^2} - \frac{L_i}{e_i^d}, \tag{3.9}$$

$$\beta_i^* = \frac{(C+P)\frac{e_i^a}{e_i^d}\frac{C+P}{C}}{\left(N + \frac{C+P}{C}\sum_{j=1}^{N}\frac{e_j^a}{e_j^d}\right)^2} - \frac{U_i - L_i}{e_i^a}, \tag{3.10}$$

$$\delta_i^* = \frac{e_i^d}{e_i^d + e_i^a\frac{C+P}{C}}. \tag{3.11}$$

Proof. The conditions $\frac{(C+P)\frac{e_i^a}{e_i^d}}{\left(N+\frac{C+P}{C}\sum_{j=1}^{N}\frac{e_j^a}{e_j^d}\right)^2} \geq \frac{L_i}{e_i^d}$ and $\frac{(C+P)\frac{e_i^a}{e_i^d}\frac{C+P}{C}}{\left(N+\frac{C+P}{C}\sum_{j=1}^{N}\frac{e_j^a}{e_j^d}\right)^2} \geq \frac{U_i-L_i}{e_i^a}$ ensure that the obtained solution is non-negative. Now it is easy to check that the provided solution satisfies the first order conditions and it is the only solution that can be derived from the first order conditions. □

Corollary 3.3. *If the detection probability function is given in equation* (3.8)*, the second stage game is of Infiltration/Harassment type, i.e.* $C_1 = \cdots = C_N = C$*,* $L_i, U_i \ll C$ *and* $\frac{C+P}{C}\frac{e_i^a}{e_i^d} < N$ *then increasing* e_i^d *will decrease both* α_i^* *and* β_i^**. On the other hand, increasing* e_i^a *will increase both* α_i^* *and* β_i^**.*

Proof. Conditions $L_i, U_i \ll C$ ensure that the solution in equations (3.9) and (3.10) is valid. Now, from these equations, it is easy to take the first derivatives with respect to e_i^d and e_i^a and verify the following: $\frac{\partial\alpha_i^*}{\partial e_i^d} \leq 0$, $\frac{\partial\beta_i^*}{\partial e_i^d} \leq 0$, $\frac{\partial\alpha_i^*}{\partial e_i^a} \geq 0$ And $\frac{\partial\beta_i^*}{\partial e_i^a} \geq 0$. □

Remark 3.4. *Corollary 3.3 states that if investment efficiency factor for the attacker increases, the investment levels for both players increase. On the other hand, if investment efficiency factor for the defender increases, the investment levels for both players decrease. This is an interesting result which is also valid for the budget constrained case (see corollary 3.5). If we consider the increase in investment efficiency as discovering a new technology, if a hostile agent, i.e. the attacker, obtains this new technology, then we observe a proliferation in security investments. However, if this new technology is obtained by a non-hostile agent, i.e. the defender, it leads to a reduction in security investments.*

In the next example, we analyze the Nash Equilibrium for the case with two targets.

Example. We consider an example with two targets where the second stage game is of Infiltration/Harassment type, i.e. $C_1 = C_2 = C$, and the detection probability function form is given in equation (3.8). We assume that $P = 100, L_1 = L_2 = 0.9, U_1 = U_2 = 1, e_1^d = e_2^d = 1$ and $e_1^a = e_2^a = 1$. Using corollary 3.2 we can compute the unique Nash Equilibrium for this example. We further analyze the effect of players' deviations in their utility and best response strategies. First, we compute the effects of such deviations on players' total utility. The total payoff for the attacker is given as:

$$u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_1^a(\boldsymbol{\beta}) + u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = -\beta_1 - \beta_2 - \frac{C + P}{\frac{1}{\delta_1(\alpha_1, \beta_1)} + \frac{1}{\delta_2(\alpha_2, \beta_2)}} + C. \quad (3.12)$$

Figure 3.1 shows the attacker's total utility as a function of his investment on target 1 when all other decision variables are at their equilibrium level. This figure shows that the attacker's utility has an inverse U shaped form. This is a well-known shape that has been identified by many papers in the literature for attacker's utility (e.g. [86]), attacker's investments (e.g. [60]) and defender's investments (e.g. [55]). Moreover, Figure 3.1 shows that the payoff is higher for higher target values and optimal attack investments increase for higher target values. The total payoff for the defender is given as:

$$u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_1^d(\boldsymbol{\beta}) + u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = -\alpha_1 - \alpha_2 + \frac{C}{\frac{1}{\delta_1(\alpha_1, \beta_1)} + \frac{1}{\delta_2(\alpha_2, \beta_2)}} - C. \quad (3.13)$$

Figure 3.2 shows the defender's payoff as a function of her investments on target 1 when all other decision variables are at their equilibrium level. This function is also concave, as was proved in lemma 3.2. Moreover for higher target values, the optimum investment value is higher. This is in line with other results in the literature (e.g. [61]).

Attacker's best response

We now compute the attacker's best investment level in target 1 as a function of defender's investment in target 1. We use the first order condition to obtain the best response:

$$\frac{\partial u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})}{\partial \beta_1} = -1 - (C+P) \frac{\frac{\frac{\partial \delta_1(\alpha_1,\beta_1)}{\partial \beta_1}}{\delta_1^2(\alpha_1,\beta_1)}}{\left(\frac{1}{\delta_1(\alpha_1,\beta_1)} + \frac{1}{\delta_2(\alpha_2,\beta_2)}\right)^2} = 0, \tag{3.14}$$

$$\frac{\frac{\frac{\partial \delta_1(\alpha_1,\beta_1)}{\partial \beta_1}}{\delta_1^2(\alpha_1,\beta_1)}}{\left(\frac{1}{\delta_1(\alpha_1,\beta_1)} + \frac{1}{\delta_2(\alpha_2,\beta_2)}\right)^2} = \frac{-1}{C+P}. \tag{3.15}$$

For our example:

$$\beta_1^* = \frac{\sqrt{(C+P)e_1^a(e_1^d\alpha_1 + L_1)} - (e_1^d\alpha_1 + L_1)(\frac{1}{\delta_2(\alpha_2,\beta_2)} + 1) - (U_1 - L_1)}{e_1^a}. \tag{3.16}$$

Figure 3.3 shows the attacker's best response as a function of the defender's investments when all other decision variables are in equilibrium. According to this figure, as defensive investments increase, the attacker at first increases the attack investments to keep up with the defender, but after a certain point, the attacker starts decreasing his investments, until he is completely deterred from investing.

Defender's best response

We use the first order condition to obtain the defender's best response function. We have:

$$\frac{\partial u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})}{\partial \alpha_1} = -1 + C \frac{\frac{\frac{\partial \delta_1(\alpha_1,\beta_1)}{\partial \alpha_1}}{\delta_1^2(\alpha_1,\beta_1)}}{\left(\frac{1}{\delta_1(\alpha_1,\beta_1)} + \frac{1}{\delta_2(\alpha_2,\beta_2)}\right)^2} = 0, \tag{3.17}$$

$$\frac{\frac{\frac{\partial \delta_1(\alpha_1,\beta_1)}{\partial \alpha_1}}{\delta_1^2(\alpha_1,\beta_1)}}{\left(\frac{1}{\delta_1(\alpha_1,\beta_1)} + \frac{1}{\delta_2(\alpha_2,\beta_2)}\right)^2} = \frac{1}{C}. \tag{3.18}$$

Giving:

$$\alpha_1^* = \frac{1}{e_1^d}\left(\frac{\sqrt{Ce_1^d(e_1^a\beta_1 + U_1 - L_1)} - (e_1^a\beta_1 + U_1 - L_1)}{\frac{1}{\delta_2(\alpha_2,\beta_2)} + 1}\right) - \frac{L_1}{e_1^d}. \tag{3.19}$$

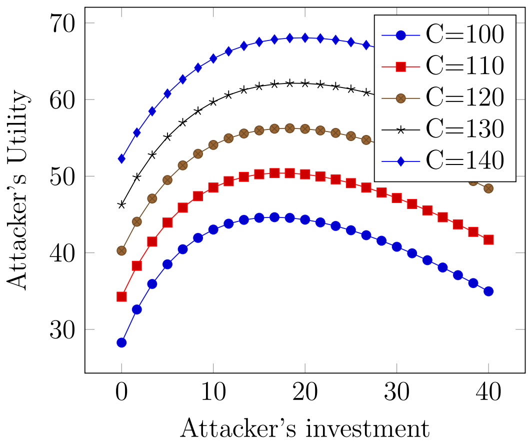


Figure 3.1: Attacker's utility as a function of his investment

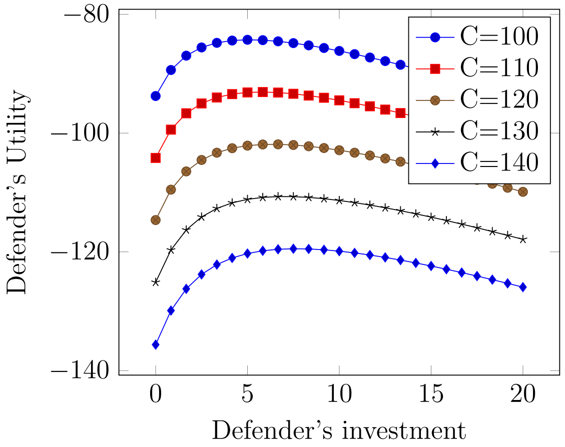


Figure 3.2: Defender's utility as a function of her investment

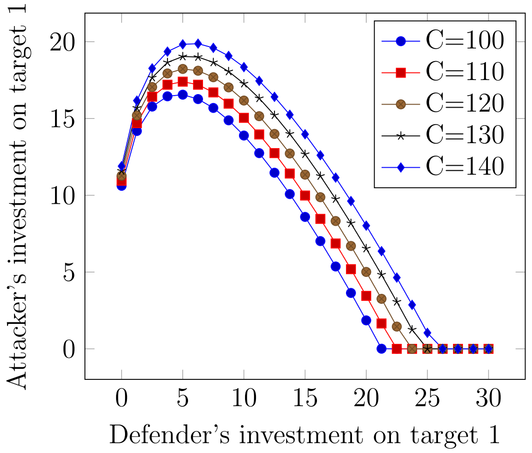


Figure 3.3: Attacker's best response

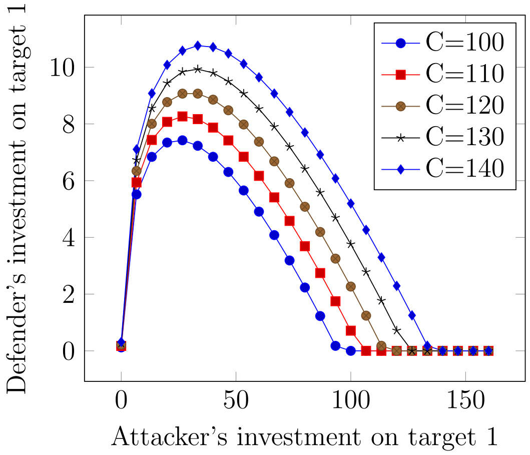


Figure 3.4: Defender's best response

Figure 3.4 shows the defender's best response as a function of the attacker's investments when all other decision variables are in equilibrium. According to this figure, as attack investments increase, the defender at first increases the defensive investments to keep up with the attacker, but after a certain point, the defender starts decreasing her investments, until she is completely deterred from investing. Moreover, defender's optimum investment level is higher for higher target values. This is in line with other results in the literature (e.g. [61]).

Budget Constrained Model

In this section, we investigate the budget constrained model where there is no investment disutility in the first stage, however, both players have a budget limit. Equations (3.20) and (3.21) show the player's total payoff functions.

$$u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}}, \tag{3.20}$$

$$u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y}) = u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = -\frac{1 - \sum_{j=1}^{k} \dfrac{C_j}{\delta_j(\alpha_j, \beta_j)(C_j + P)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)(C_j + P)}}. \tag{3.21}$$

The following lemma shows quasi-concavity of payoff function for both players.

Lemma 3.4. *$u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ and $u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ are quasi-concave in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively.*

Proof. See Appendix A.6. □

The following theorem establishes existence and uniqueness of the Nash equilibrium for the budget constrained invest-defend game.

Theorem 3.3. *The budget constrained game has a unique Nash Equilibrium $(\boldsymbol{\alpha}, \boldsymbol{\beta})$.*

Proof. See Appendix A.7. □

To compute the unique Nash Equilibrium, we use the Karush-–Kuhn—Tucker (KKT) [78] conditions for both the defender and the adversary. The optimization problem for the defender is as follows:

$$max \quad u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{3.22}$$

$$\sum_{j=1}^{k} \alpha_j = A, \tag{3.23}$$

$$\alpha_j \geq 0. \tag{3.24}$$

KKT conditions for this optimization problem are as follows:

$$\frac{\partial u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \alpha_j} = \lambda - \mu_j, \tag{3.25}$$

$$\sum_{j=1}^{k} \alpha_j = A, \tag{3.26}$$

$$\mu_j \alpha_j = 0, \tag{3.27}$$

$$\mu_j \geq 0, \quad \alpha_j \geq 0. \tag{3.28}$$

Remark 3.5. *If a site receives investment $0 < \alpha_i \leq A$, at optimality we have: $\frac{\partial u_2^d(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_i} = \lambda$. This implies that if $\alpha_i, \alpha_j > 0$ for $i \neq j$ then: $\frac{\partial u_2^d(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_i} = \frac{\partial u_2^d(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \alpha_j}$.*

The optimization problem for the adversary is as follows:

$$max \quad u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{3.29}$$

$$\sum_{j=1}^{k} \beta_j = B, \tag{3.30}$$

$$\beta_j \geq 0. \tag{3.31}$$

KKT conditions for this optimization problem are as follows:

$$\frac{\partial u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \beta_j} = \gamma - \pi_j, \tag{3.32}$$

$$\sum_{j=1}^{k} \beta_j = B, \tag{3.33}$$

$$\pi_j \beta_j = 0, \tag{3.34}$$

$$\pi_j \geq 0, \quad \beta_j \geq 0. \tag{3.35}$$

Remark 3.6. *If a site receives investment* $0 < \beta_i \leq B$*, at optimality we have:* $\frac{\partial u_2^a(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_i} = \gamma$*. This implies that if* $\beta_i, \beta_j > 0$ *for* $i \neq j$ *then:* $\frac{\partial u_2^a(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_i} = \frac{\partial u_2^a(\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \beta_j}$.

Although there is no closed-form equilibrium for the invest-defend game when the second stage game is of the Maximal Damage type, under certain detection probability functions, a closed-form equilibrium exists when the second stage game is of Infiltration/Harassment type .

Corollary 3.4. *For the budget constrained game, if the second stage game is of Infiltration/Harassment type, i.e. when* $C_1 = \cdots = C_N = C$*, and the detection probability function is given in equation* (3.8)*, if* $\frac{\frac{e_i^a}{e_i^d}}{\sum_{j=1}^N \frac{e_j^a}{e_j^d}}\Big(B + \sum_{j=1}^N (\frac{U_j}{e_j^a} - \frac{L_j}{e_j^a})\Big) - (\frac{U_i}{e_i^a} - \frac{L_i}{e_i^a}) \geq 0$ *and* $\frac{\frac{e_i^a}{e_i^d}}{\sum_{j=1}^N \frac{e_j^a}{e_j^d}}\Big(A + \sum_{j=1}^N \frac{L_j}{e_j^d}\Big) - \frac{L_i}{e_i^d} \geq 0$ *for* $i = 1, \ldots, N$*, then the first stage game has a unique closed form solution given by:*

$$\beta_i^* = \frac{\frac{e_i^a}{e_i^d}}{\sum_{j=1}^N \frac{e_j^a}{e_j^d}}\left(B + \sum_{j=1}^N (\frac{U_j}{e_j^a} - \frac{L_j}{e_j^a})\right) - (\frac{U_i}{e_i^a} - \frac{L_i}{e_i^a}), \quad i = 1, \ldots, N, \tag{3.36}$$

$$\alpha_i^* = \frac{\frac{e_i^a}{e_i^d}}{\sum_{j=1}^N \frac{e_j^a}{e_j^d}}\left(A + \sum_{j=1}^N \frac{L_j}{e_j^d}\right) - \frac{L_i}{e_i^d}, \quad i = 1, \ldots, N, \tag{3.37}$$

with:

$$\delta_i^* = \frac{A + \sum_{j=1}^N \frac{L_j}{e_j^d}}{A + \sum_{j=1}^N \frac{L_j}{e_j^d} + B + \sum_{j=1}^N (\frac{U_j}{e_j^a} - \frac{L_j}{e_j^a})}, \quad i = 1, \ldots, N. \tag{3.38}$$

Proof. The conditions $\frac{\frac{e_i^a}{e_i^d}}{\sum_{j=1}^N \frac{e_j^a}{e_j^d}} \left(B + \sum_{j=1}^N (\frac{U_j}{e_j^a} - \frac{L_j}{e_j^a})\right) - (\frac{U_i}{e_i^a} - \frac{L_i}{e_i^a}) \geq 0$ and $\frac{\frac{e_i^a}{e_i^d}}{\sum_{j=1}^N \frac{e_j^a}{e_j^d}} \left(A + \sum_{j=1}^N \frac{L_j}{e_j^d}\right) - \frac{L_i}{e_i^d} \geq 0$ for $i = 1, \dots, N$ ensure that the optimal investment strategies for both players are non-negative. Based on the assumptions, and using the KKT conditions with $\pi_j = \mu_j = 0$ for $j = 1, \dots, N$, the equations in the corollary are obtained. □

Remark 3.7. *Note that, when the second stage game is of Infiltration/Harassment type, the equilibrium investment strategy succeeds in making all detection probabilities the same, hence making the optimal defend-attack strategies uniformly distributed over the targets.*

Corollary 3.5. *For the budget constrained game, if the detection probability function is given in equation* (3.8)*, the second stage game is of Infiltration/Harassment type, i.e.* $C_1 = \cdots = C_N = C$*,* $L_i, U_i \ll A, B$*, then increasing* e_i^d *will decrease both* α_i^* *and* β_i^**. On the other hand, increasing* e_i^a *will increase both* α_i^* *and* β_i^**.*

Proof. The condition $L_i, U_i \ll A, B$ ensures that the solution in equations (3.36) and (3.37) is always valid. From these equations, it is easy to take the first derivatives with respect to e_i^d and e_i^a and verify the following: $\frac{\partial \alpha_i^*}{\partial e_i^d} \leq 0$, $\frac{\partial \beta_i^*}{\partial e_i^d} \leq 0$, $\frac{\partial \alpha_i^*}{\partial e_i^a} \geq 0$ And $\frac{\partial \beta_i^*}{\partial e_i^a} \geq 0$. □

Remark 3.8. *Corollary 3.5 states that, if investment efficiency factor for the attacker increases, the investment levels for both players increase. On the other hand, if investment efficiency factor for the defender increases the investment levels for both players decrease. This is the budget constrained equivalent of corollary 3.3.*

3.4 Application to Real Data

In this section, we apply the budget constrained model to real data from [139] presented in Table 3.1. This table provides estimates on the expected annual terrorism losses to the 10 most valuable urban areas of the United States. It also presents the grant allocation data for these areas. We consider two aspects of the expected damage: monetary value (represented by expected property loss) and fatality value (represented by total number of fatalities and injuries). For each of these two aspects, we use the proposed two-stage approach to allocate defensive resources among these urban areas. We use the total grant allocation (for all 10 urban areas i.e. 270 million dollars) as the total available budget for the defender and consider different values for adversary's budget. [18] have also used this data set to study the effect of different factors on the optimal allocation of resources. We compare our results with the results obtained by [18] whenever possible throughout our experiments. We assume that the detection probability function is of the form presented in equation (3.8). Unless stated otherwise, we use the following values for parameters of the game: $L_i = 0.9, U_i = 1, e_i^d = e_i^a = 1$ for $i = 1, 2, \ldots, N,$ and $B = 0.3A$. Also note that because the target valuations are not the same, i.e. $C_1 = \cdots = C_N = C$ does not hold, the second stage game in this section is of Maximal Damage type.

3.4.1 Analysis for Monetary Value Data

In this section, we perform the analysis based on the monetary value of each urban area. Table 3.2 shows the optimal strategies of both players for $P = 400$ and different values for attacker's budget. This table shows that, for $B = 0.3A$, the defender distributes the investments among the first six most important areas and the level of investments decreases as the value of the area decreases. No investment is allocated to the next important area, i.e. BSTN, however, the second stage strategy comple-

Table 3.1: Expected damage data for 10 urban areas with highest losses

Urban Area	Expected property loss ($million)	Expected Fatalities & Injuries	FY2004 UASI Grant Allocation ($ million)
New York (NY)	413	5350	47
Chicago (CH)	115	1212	34
San Francisco (SF)	57	472	26
Washington DC (WDC)	36	681	29
Los Angeles (LA)	34	402	40
Philadelphia (PHL)	21	199	23
Boston (BSTN)	18	225	19
Houston (HSTN)	11	160	20
Newark (NW)	7.3	74	15
Seattle (STL)	6.7	88	17
Total	719	8863	270

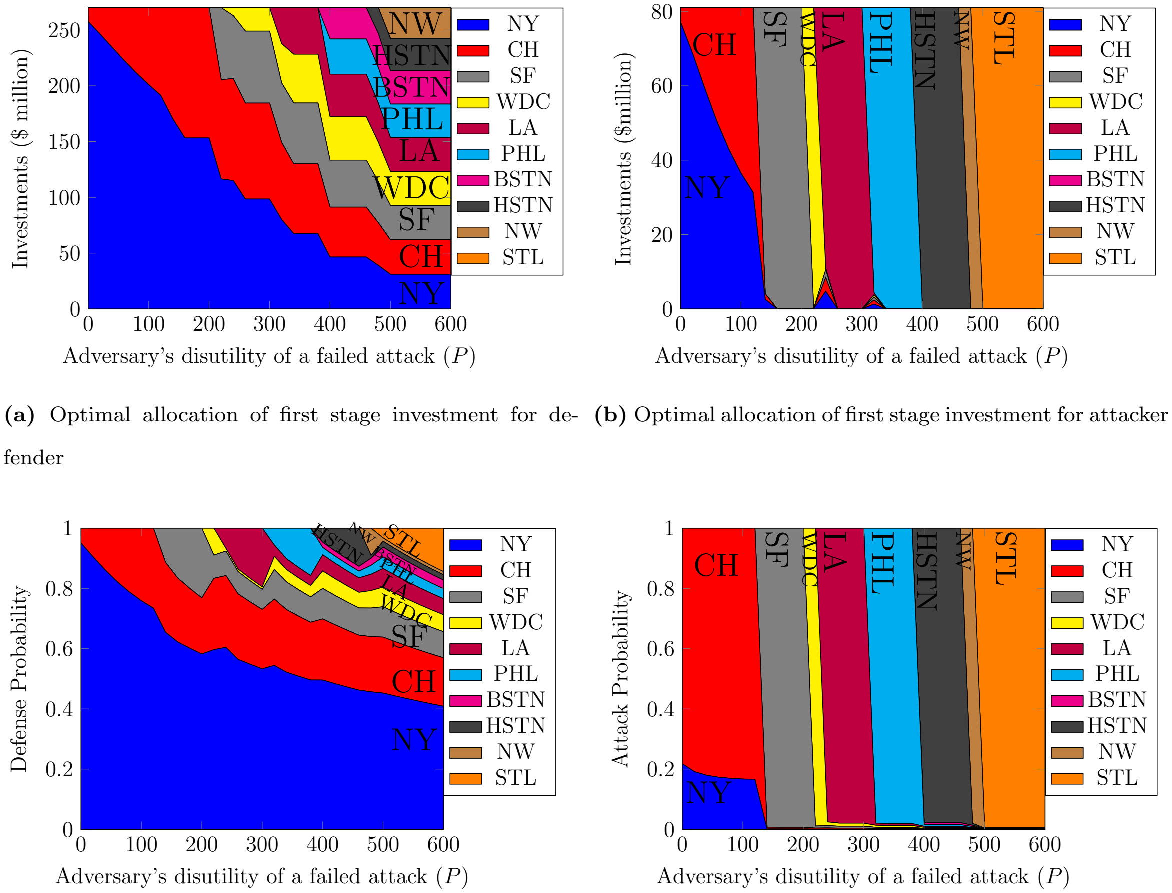
ments the first stage investment decision by covering BSTN with a relatively high probability. Bar this exception, the second stage defense probabilities also decrease as the value of the area decreases. For the attacker, all of the first stage investments go to BSTN and most of the second stage effort is concentrated in BSTN. This is, roughly speaking, in line with the assumptions of other models, including [18], that the adversary concentrates his efforts on one area. However, the complementary interaction between the first stage and second stage decisions has not been observed in previous studies. Another interesting observation is that as the attacker's budget increases, both first stage and second stage decisions shift towards more important areas.

Next, we study the effect of attacker's disutility from a failed attack, P, on the optimal first stage and second stage decisions for both players. Figure 3.5a shows

Table 3.2: Optimal investment and defend/attack strategies for monetary data with $P = 400$

	$B = 0.3A$				$B = 0.6A$				$B = 0.9A$			
	α_i^*	β_i^*	x_i^*	y_i^*	α_i^*	β_i^*	x_i^*	y_i^*	α_i^*	β_i^*	x_i^*	y_i^*
NY	59.82	0.00	0.487	0.000	67.28	0.00	0.483	0.000	97.59	0.00	0.467	0.000
CH	56.01	0.00	0.190	0.002	62.31	0.00	0.184	0.001	85.35	0.00	0.158	0.001
SF	50.16	0.00	0.087	0.003	54.54	0.00	0.080	0.002	64.29	0.00	0.051	0.001
WDC	42.42	0.00	0.043	0.005	43.91	0.00	0.035	0.003	22.78	0.00	0.005	0.002
LA	41.05	0.00	0.038	0.006	41.97	0.00	0.031	0.003	0.00	243.00	0.318	0.995
PHL	20.53	0.00	0.009	0.009	0.00	162.00	0.187	0.990	0.00	0.00	0.000	0.000
BSTN	0.00	81.00	0.145	0.974	0.00	0.00	0.000	0.000	0.00	0.00	0.000	0.000

the effect of P on defender's optimal investment decisions. As seen in this figure, as attacker's disutility of a failed attack increases, the defender distributes the investments to cover more targets. This is due to the fact that as disutility of a failed attack increases, the attacker is less willing to risk being caught and more willing to attack more vulnerable targets where he is less likely to have an unsuccessful attack. In response, the defender distributes her investments to cover more targets. Figure 3.5b shows the effect of P on attacker's optimal investment decisions. According to this figure, attacker's investments generally concentrate on a single area which is an unprotected area (in terms of defender's first stage investments) with the highest value. Figure 3.5c shows the defender's second stage defense probability assignments as a function of P. As seen in this figure, defender's probability assignments are similar to her first stage investment assignments in the sense that more areas get covered as P increases. Moreover, the complementary interaction observed in table 3.2, is also visible in figure 3.5c. For example, for around the point with $P = 150$, the



(a) Optimal allocation of first stage investment for defender

(b) Optimal allocation of first stage investment for attacker

(c) Optimal allocation of second stage probabilities for defender

(d) Optimal allocation of second stage probabilities for attacker

Figure 3.5: Analysis for monetary value data

investments are distributed between two most important areas i.e. NY and CH. The next most important area, SF, receives no investments in the first stage. However, the second stage defense probability assignment complements the first stage decision by defending SF with a relatively high probability. Figure 3.5d shows the attacker's second stage probabilities as a function of P. As seen in this figure, the second stage attack probabilities are in line with the first investment decisions. In other words, the second stage probabilities are concentrated on the same target that received majority of the investments in the first stage.

3.4.2 Analysis for Fatality Value Data

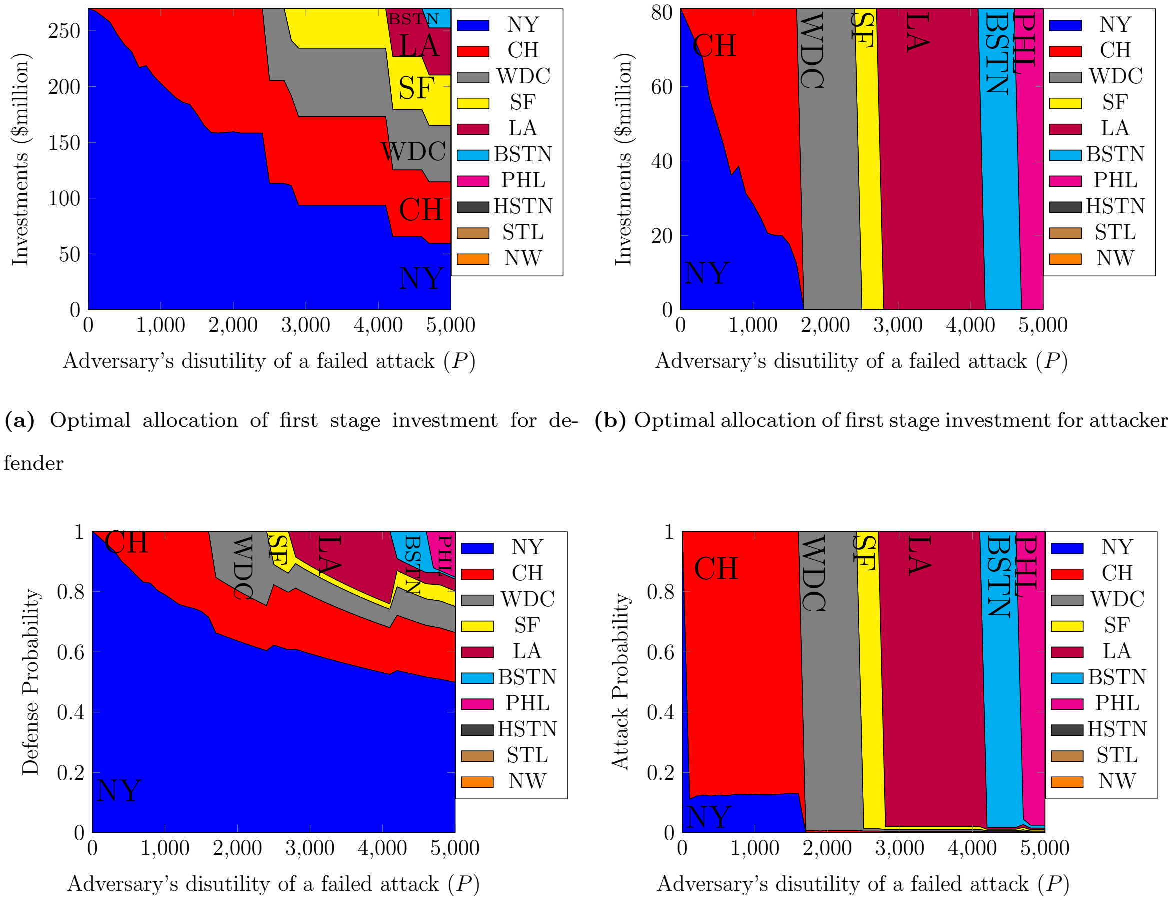
In this section, we perform the two-stage game analysis based on the fatality value of each urban area. We study the effect of the attacker's budget on both players' strategies. This budget is represented by a percentage of the defender's budget. Table 3.3 shows the optimal strategies of both players for $P = 5000$ and different values for attacker's budget. This table shows that, for $B = 0.3A$, the defender distributes the investments among the first six most important areas and the level of investments decrease as the value of the area decreases. No investment is allocated to the next important area, i.e. PHL, however, the second stage strategy complements the first stage investment decision by covering PHL with a relatively high probability. Ignoring this exception, the second stage defense probabilities are also distributed proportional to the value of the area. For the attacker, all of the first stage investments go to PHL and most of the second stage effort is concentrated in PHL. This is, roughly speaking, in line with the assumptions of other models, including [18], that the adversary concentrates his efforts on one area. Moreover, similar to the results of [18], different valuations of targets lead to different investment allocations. Similar observations can be made for other values of attacker's budget. Another interesting observation is that as the attacker's budget increases, both first stage and second stage decisions shift towards more important areas.

Next, we study the effect of attacker's disutility from a failed attack, P, on the optimal first stage and second stage decisions. Figure 3.6a shows the effect of P on defender's optimal investment decisions. As seen in this figure, similar to the case of monetary value analysis, as attacker's disutility of a failed attack increases, the defender distributes the investments to cover more targets. Figure 3.6b shows the effect of P on attacker's optimal investment decisions. According to this figure, attacker's investments generally concentrate on a single unprotected area (in terms

Table 3.3: Optimal investment and defend/attack strategies for fatality data with $P = 5000$

	$B = 0.3A$				$B = 0.6A$				$B = 0.9A$			
	α_i^*	β_i^*	x_i^*	y_i^*	α_i^*	β_i^*	x_i^*	y_i^*	α_i^*	β_i^*	x_i^*	y_i^*
NY	59.37	0.00	0.499	0.000	75.95	13.48	0.574	0.000	158.23	0.00	0.452	0.000
CH	55.13	0.00	0.165	0.002	64.56	11.55	0.185	0.002	111.77	0.00	0.087	0.002
WDC	50.49	0.00	0.087	0.003	53.09	9.60	0.094	0.003	0.00	243	0.461	0.997
SF	45.34	0.00	0.052	0.005	41.49	7.63	0.053	0.005	0.00	0.00	0.00	0.00
LA	42.15	0.00	0.039	0.005	34.91	6.52	0.039	0.006	0.00	0.00	0.00	0.00
BSTN	17.52	0.00	0.007	0.010	0.00	113.23	0.055	0.984	0.00	0.00	0.00	0.00
PHL	0.00	81.00	0.152	0.975	0.00	0.00	0.000	0.000	0.00	0.00	0.00	0.00

of defender's first stage investments) with the highest value. Figure 3.6c shows the defender's second stage defense probability assignments as a function of P. As seen in this figure, similar to her first stage investment assignments, more areas get covered as P increases. Moreover, the complementary interaction between the firsat stage and second stage decisions, as observed in table 3.3, is also visible in figure 3.6c. Figure 3.6d shows the attacker's second stage probabilities as a function of P. According this figure, the second stage attack probabilities are in line with the first investment decisions. In other words, the second stage probabilities are concentrated on a single target which is the same target that received majority of the investments in the first stage.



(a) Optimal allocation of first stage investment for defender

(b) Optimal allocation of first stage investment for attacker

(c) Optimal allocation of second stage probabilities for defender

(d) Optimal allocation of second stage probabilities for attacker

Figure 3.6: Analysis for fatality value data

Chapter 4

A Decomposable Resource Allocation Model with Generalized Overarching Protections

4.1 Introduction and Literature Review

Defensive resource allocation to protect a set of valuable assets against terrorist attacks or natural disasters has been the subject of intensive studies [18, 21, 62, 126, 145]. One aspect of this problem is to strike a balance between protecting individual assets and the overarching protection options. Overarching protections refer to the options that protect multiple assets at the same time. For example, a country can allocate resources to protect its borders to reduce the potential damage from international terrorism. Similarly, expending resources on gathering information and intelligence to counter terrorism is another form of overarching protection.

Powell [120], and Haphuriwat and Bier [49] conducted the early studies on the trade-off between individual target hardening and overarching protection. Powell investigated a model in which the defender has the option of allocating resources to harden the targets individually or to protect all of the targets via enhancing the border security. Haphuriwat and Bier introduced a model to allocate resources between target hardening and an overarching protection option covering all targets. They studied the effect of various factors on the relative desirability of each option. Golalikhani and Zhuang [44] developed a model with a defender simultaneously protecting any subset of targets based on their functional similarity or geographical proximity. Hausken [57] presented a two-period resource allocation game. In the first period, both players allocate their resources to engage in an overarching contest covering all of the targets. If the attacker wins the overarching contest, in the second period, the players decide on resource allocation to defend/attack individual targets. Hausken [58] considered a system consisting of two components either in series or in parallel. In his model, the players can either allocate resources to special efforts to protect individual components or a general effort to protect both components in the system. The difference of this model from the existing ones with overarching protection is that, the existing models regard the overarching protection as an extra layer of protection that the attacker has to breach to have a successful attack. However, in his proposed model, there is only one protection layer and the special and general protection efforts operate additively to contribute to a single joint protection. Hausken [59] investigated a similar system with two independent components.

There are a number of studies that considered systems consisting of logically linked components. For example, Levitin and Hausken [86] examined individual and overarching protections for series and parallel systems. Hausken [56] expanded this model to include heterogeneous unit protection costs. Levitin *et al.* [89] introduced a model that generalizes the k-out-of-n system. In this model, the damage to the system depends on the number of destroyed elements as well as the unfulfilled demand. Levitin *et al.* [90] developed a three-stage minimax game model with multiple overarching protections and a system consisting of identical elements. In this model, the defender decides the number of groups of targets to protect using overarching protections as well as the number of targets to protect individually within each group. Peng *et al.* [117] considered the resource allocation problem to individual, overarching protection and replacement for a parallel system of heterogeneous components.

Most existing models in literature assume that there is only one overarching protection option that protects all of the targets. However, this may not be true in reality. For example, in case of emergency response, investment is not limited to only one option that covers the entire country. It is possible to make targeted investments that are focused on a city or an area inside a city. Moreover, investment in border security can be divided into different points of entry, each of which is expected to benefit areas that are closer to that particular point of entry. The only model with multiple overarching protections is proposed by Levitin *et al.* [90]. However, this model assumes that the targets are identical and each overarching protection covers a fixed number of targets. Therefore, the overarching protection options are identical and the defender decides on how many times to use this option. In reality, the targets may not be identical and, depending on the subset of targets that are covered, various options for overarching protections may be available. To this end, we introduce overarching protection options that protect a subset of targets. We consider two types of overarching protections: country-level overarching protections and city-level overarching protections. Each country-level overarching protection option protects all of the assets in a set of cities. And each city-level overarching protection option in a city protects a subset of assets. Moreover, there are different types of natural disasters and the defender has to decide on how much to invest to protect against each disaster type in each city. Another consideration in this area is that the number of targets maybe very large and a practical resource allocation model needs to be scalable for problems of realistically large size. We show that our proposed resource allocation model is a convex optimization problem that can be solved in polynomial time. Moreover, we also demonstrate that the proposed model can be decomposed into smaller city-level subproblems. Using this observation, we develop an efficient decomposition approach to optimally solve the proposed resource allocation problem.

The rest of this chapter is organized as follows. Section 4.2 introduces the proposed resource allocation model. Section 4.3 develops a solution approach based on decomposing the problem into city-level subproblems to solve the proposed model. Section 4.4 demonstrates numerical experiments to investigate the efficiency of the proposed algorithms and to gain insight into properties of the model.

4.2 Problem Description

A defender has a budget, say B, to allocate in order to protect cities in a country against both natural and man-made disasters. Each asset j in city i has a value C_{ij} that will be lost in case of a successful attack or a natural disaster. Against man-made attacks, the defender can either protect assets in cities individually or collectively through overarching protection options. Overarching protection refers to alternatives that lead to protecting more than one individual asset, e.g., border security, public health, emergency response, or intelligence. Two types of overarching protections exist: country-level overarching protections and city-level overarching protections. Each country-level overarching protection option o protects all of the assets in a set of cities Γ_o. On the other hand, each city-level overarching protection option l in city i protects a set of assets Λ_{il}. A single adversary is the perpetrator of a man-made disaster and chooses the asset with the highest expected damage to attack. If there are multiple assets with the highest expected damage, we assume that the adversary chooses one of them arbitrarily. In order to successfully destroy an asset, all of the protection measures need to be breached. There are different types of natural disasters and the defender decides how much to invest for protection against each disaster type in each city.

Model parameters are listed as follows:

- i : Index for cities, $i = 1, \ldots, I$.
- j : Index for assets in city i, $j = 1, \ldots, J_i$.
- k : Index for the type of natural disaster, $k = 1, \ldots, K$.
- ρ : Probability of an intentional attack.
- ω_k : Probability of a type k natural disaster.
- B : Defender's budget.
- C_{ij} : Value of asset j in city i.
- x_i : Amount of resource allocated to protect city i against intentional attacks.
- x_{ij}^H: Amount of resource allocated to harden asset j in city i against intentional attacks.
- x_o^C : Amount of resource allocated to country-level overarching protection option o.
- x_{ik}^N: Amount of resource allocated to protect city i against natural disaster of type k.
- x_{il}^L: Amount of resource allocated to city-level overarching protection option l, in city i.
- Γ_o: Set of cities that are protected in country-level overarching protection option o.

- Ψ_i : Set of country-level overarching protection options that protect city i.
- Λ_{il} : Set of assets in city i that are protected through city-level overarching protection option l.
- Ω_{ij} : Set of city-level overarching protection options in city i that protect asset j.
- $f_i(x_i)$: Expected damage from a man-made attack in city i, given that budget level is x_i, and all of the country-level overarching protections are breached.
- $P_o^C(x_o^C)$: Probability of breaching country-level overarching protection option o.
- $P_{il}^L(x_{il}^L)$: Probability of breaching city-level overarching protection in city i option l.
- $P_{ij}^H(x_{ij}^H)$: Probability of breaching hardening protection in for asset j in city i.
- $P_{ik}^N(x_{ik}^N)$: Probability of failure of protection against natural hazard type k in city i.

Using this notation, the resource allocation problem can be formulated as follows:

$$\text{Min } \rho\left[\max_{(i,j)}\left\{C_{ij}P_{ij}^H(x_{ij}^H)\prod_{l\in\Omega_{ij}}P_{il}^L(x_{il}^L)\prod_{o\in\Psi_i}P_o^C(x_o^C)\right\}\right]+\sum_k\omega_k\sum_i P_{ik}^N(x_{ik}^N)\sum_j C_{ij} \quad (4.1)$$

subject to

$$\sum_i\left[\sum_j\left(x_{ij}^H+\sum_{l\in\Omega_{ij}}x_{il}^L\right)+\sum_{o\in\Psi_i}x_o^C\right]+\sum_i\sum_k x_{ik}^N\leq B, \quad (4.2)$$

$$x_{ij}^H, x_{il}^L, x_o^C, x_{ik}^N \geq 0,\ \forall\ k=1,\ldots,K,\ \ o\in\Psi_i,\ \ l\in\Omega_{ij}\ \text{ for }\ i=1,\ldots,I,\ \ j=1,\ldots,J. \quad (4.3)$$

In this formulation, the objective function is to minimize the expected damage from both man-made and natural disasters. The first term is the expected damage from man-made disasters. For each asset j in city i, the probability of a successful attack is $P_{ij}^{H}(x_{ij}^{H})\prod_{l\in\Omega_{ij}} P_{il}^{L}(x_{il}^{L})\prod_{o\in\Psi_i} P_{o}^{C}(x_{o}^{C})$. Therefore, the expected damage of an attack on asset j in city i is $C_{ij}P_{ij}^{H}(x_{ij}^{H})\prod_{l\in\Omega_{ij}} P_{il}^{L}(x_{il}^{L})\prod_{o\in\Psi_i} P_{o}^{C}(x_{o}^{C})$. The attacker chooses the asset that leads to the maximum expected damage. The second term in the objective function is the expected damage from natural disasters. We assume that natural disasters affect entire cities, thus investments to protect against them need to cover all assets in a city. Therefore, for each city i, the expected damage from a natural disaster of type k is equal to $\sum_i P_{ik}^{N}(x_{ik}^{N})\sum_j C_{ij}$. Clearly, total investment is constrained by the budget.

The following lemma shows the conditions under which the above formulation is a convex optimization program.

Lemma 4.1. *If the success probability functions are log-convex, then the resource allocation problem is a convex optimization problem.*

Proof. For each pair (i,j), the expression $P_{ij}^{H}(x_{ij}^{H})\prod_{l\in\Omega_{ij}} P_{il}^{L}(x_{il}^{L})\prod_{o\in\Psi_i} P_{o}^{C}(x_{o}^{C})$ is a log-convex function. Note that log-convex functions are also convex.
Thus $\max_{(i,j)}\left\{C_{ij}P_{ij}^{H}(x_{ij}^{H})\prod_{l\in\Omega_{ij}} P_{il}^{L}(x_{il}^{L})\prod_{o\in\Psi_i} P_{o}^{C}(x_{o}^{C})\right\}$ is a point-wise maximum of a set of convex functions. This means that
$\max_{(i,j)}\left\{C_{ij}P_{ij}^{H}(x_{ij}^{H})\prod_{l\in\Omega_{ij}} P_{il}^{L}(x_{il}^{L})\prod_{o\in\Psi_i} P_{o}^{C}(x_{o}^{C})\right\}$ is convex. Therefore, the objective function is a linear combination of convex functions, which is convex. Moreover, the constraint is linear. Therefore, the resource allocation problem (4.1)-(4.3) is to minimize a convex function with linear constraints. This completes the proof. □

4.3 A Decomposition Approach to Solve the Resource Allocation Problem

If functions $P_o^C(x_o^C)$, $P_{il}^L(x_{il}^L)$, and $P_{ij}^H(x_{ij}^H)$ are log-convex, then the resource allocation problem can be decomposed into smaller city-level resource allocation problems. The assumption of log-convexity is not very limiting and many of the existing functions in literature have this property [18, 48, 49, 138, 144]. We can rewrite the defender's resource allocation problem as follows:

$$\min_{x_i, x_o^C, x_{ik}^N} \quad \left[\left(\max_i \rho f_i(x_i) \right) \prod_{o \in \Psi_i} P_o^C(x_o^C) + \sum_k \omega_k \sum_i P_{ik}^N(x_{ik}^N) \sum_j C_{ij} \right] \tag{4.4}$$

$$\text{subject to} \quad \sum_i \left(x_i + \sum_{o \in \Psi_i} x_o^C \right) + \sum_i \sum_k x_{ik}^N \leq B, \tag{4.5}$$

$$x_i, x_o^C, x_{ik}^N \geq 0, \ \forall \, o \in \Psi_i, \ \ l \in \Omega_{ij} \ \text{ for } \ i = 1, \ldots, I. \tag{4.6}$$

In this formulation, $f_i(x_i)$ is the expected damage of a man-made attack in city i if x_i amount has been allocated to this city for its protection against intentional attacks and all country-level overarching protections have been breached. The value of $f_i(x_i)$ is obtained by solving the following city-level resource allocation problem against intentional attacks:

$$f_i(x_i) = \min_{x_{ij}^H, x_{il}^L} \quad \max_j C_{ij} P_{ij}^H(x_{ij}^H) \prod_{l \in \Omega_{ij}} P_{il}^L(x_{il}^L) \tag{4.7}$$

$$\text{subject to} \quad \sum_j x_{ij}^H + \sum_l x_{il}^L \leq x_i, \tag{4.8}$$

$$x_{ij}^H, x_{il}^L \geq 0, \ \forall \, l \in \Omega_{ij}, \text{ for } i = 1, \ldots, I, \ j = 1, \ldots, J. \tag{4.9}$$

We refer to the above optimization problem as the city-level subproblem and show that this problem, under the conditions of Lemma 1, is a convex optimization problem. First, we need the following lemma, which is adapted from [19].

Lemma 4.2. *Let function $g_0 : \mathbb{R}^n \to \mathbb{R}$ be log-convex and $g_1, g_2, \ldots, g_h : \mathbb{R}^n \to \mathbb{R}$ be convex. Then, $f^*(\mathbf{x}) = \inf_{\mathbf{u}}\{g_0(\mathbf{u}) | \mathbf{u} \in D, g_i(\mathbf{u}) \leq x_i, i = 1, 2, \ldots, h\}$ is log-convex.*

Proof. Let function $G(\mathbf{u}, \mathbf{x}) \equiv g_0(\mathbf{u})$ be defined in the domain, $\{(\mathbf{u}, \mathbf{x}) | \mathbf{u} \in D, g_i(\mathbf{u}) \leq x_i, i = 1, 2, \ldots, h\}$. Define the domain of $f^*(\mathbf{x})$ as $\mathbf{dom}\ f^*(\mathbf{x}) = \{\mathbf{x} | (\mathbf{u}, \mathbf{x}) \in \mathbf{dom}\ G$ for some $\mathbf{u} \in \mathbb{R}^n\}$. It is easy to see that the domain of $G(\mathbf{u}, \mathbf{x})$ is convex. Therefore, $G(\mathbf{u}, \mathbf{x})$ is log-convex. Next, we show that $f^*(\mathbf{x}) = \inf_{\mathbf{u}} G(\mathbf{u}, \mathbf{x})$ is log-convex. Consider two points (x_1) and (x_2) both in $\mathbf{dom}\ f^*$. For $\epsilon > 0$ there are u_1 and u_2 in $\mathbf{dom}\ G$ such that $G(u_i, x_i) \leq f^*(x_i) + \epsilon$. We have:

$$\begin{aligned} f^*(\theta x_1 + (1-\theta)x_2) &= \inf_u G(\mathbf{u}, \theta x_1 + (1-\theta)x_2) \\ &\leq G(\theta u_1 + (1-\theta)u_2, \theta x_1 + (1-\theta)x_2) \\ &\leq G(u_1, x_1)^\theta G(u_2, x_2)^{(1-\theta)} \leq (f^*(x_1) + \epsilon)^\theta (f^*(x_2) + \epsilon)^{(1-\theta)} \\ &\leq f^*(x_1)^\theta f^*(x_2)^{(1-\theta)} + \delta(\epsilon), \end{aligned}$$

with $\delta(\epsilon)$ that converges to zero as ϵ goes to zero. Since this holds for any $\epsilon > 0$, we have:

$$f^*(\theta x_1 + (1-\theta)x_2) \leq f^*(x_1)^\theta f^*(x_2)^{(1-\theta)}.$$

This completes the proof. □

Based on this lemma for $h = 1$, the following corollary holds:

Corollary 4.1. *If $P^H_{ij}(x^H_{ij})$ and $P^L_{il}(x^L_{il})$ are log-convex, then $f_i(x_i)$ is also log-convex.*

Using Corollary 4.1, an iterative outer approximation method can be used to solve the decomposed problem. Given a set of points x^m_i for $m \in \Phi$, we can develop the following master problem:

$$\min_{x_i, x_o^C, x_{ik}^N} \quad z + \sum_k \omega_k \sum_i P_{ik}^N(x_{ik}^N) \sum_j C_{ij} \tag{4.10}$$

$$\text{subject to} \quad z \geq \rho f_i(x_i^m) e^{\frac{f_i'(x_i^m)}{f_i(x_i^m)}(x_i - x_i^m)} \prod_{o \in \Psi_i} P_o^C(x_o^C), \quad \forall i = 1, \ldots, I, \ m \in \Phi, \tag{4.11}$$

$$\sum_i x_i + \sum_{o \in \Psi_i} x_o^C + \sum_i \sum_k x_{ik}^N \leq B, \tag{4.12}$$

$$x_i, x_o^C, x_{ik}^N, z \geq 0, \ \forall \ k = 1, \ldots, K, \ o \in \Psi_i, \ \text{for } i = 1, \ldots, I. \tag{4.13}$$

In this formulation, $f_i'(x_i^m)$ is the first derivative of $f_i(x_i)$ with respect to x_i evaluated at $x_i = x_i^m$. Note that, because $f_i(x_i)$ is a log-convex function, the solution of this master problem gives a lower bound to the optimal solution of the resource allocation problem. We use the obtained x_i values to set $x_i^{M+1} = x_i$, $\Phi = \Phi \bigcup \{M+1\}$ and $M = M + 1$. We then use the $\mathbf{x}^M$ to solve the subproblems:

$$f_i(x_i^M) = \min_{x_{ij}^H, x_{il}^L} \quad \max_j C_{ij} P_{ij}^H(x_{ij}^H) \prod_{l \in \Omega_{ij}} P_{il}^L(x_{il}^L) \tag{4.14}$$

$$\text{subject to} \quad \sum_j x_{ij}^H + \sum_l x_{il}^L \leq x_i^M, \tag{4.15}$$

$$x_{ij}^H, x_{il}^L \geq 0. \tag{4.16}$$

Note that, in the subproblem, x_i^M values are fixed and they are treated as parameters. Solving the subproblems gives an upper bound which can be computed as $UB = \max_i \rho f_i(x_i) \prod_{o \in \Psi_i} P_o^C(x_o^C) + \sum_k \omega_k \sum_i P_{ik}^N(x_{ik}^N) \sum_j C_{ij}$. We continue iteratively solving the master problem and the subproblems until the lower and upper bounds converge. Algorithm 1 provides the pseudo-code for the overall decomposition procedure. The algorithm starts by initializing $M = 0$ and $\Phi = \emptyset$. Then the master problem is solved to obtain the optimal solution as $\mathbf{x}^* = [x_i^*]$, $[x_o^{C*}]$ and $[x_{ik}^{N*}]$. The algorithm then sets the current lower bound LB as the optimal objective function obtained by solving the master problem. In the next step, the algorithm adds the current point $\mathbf{x}^*$ to the set of points x_i^m, $m \in \Phi$, and updates M and Φ. We then use $\mathbf{x}^M$ to solve the subproblems (4.14)-(4.16) and obtain $[x_{ij}^{H*}]$ and $[x_{il}^{L*}]$. In the next step, the algorithm uses the current solution to compute an upper bound on the optimal objective function. In the next step, the current lower and upper bounds are compared to check if they are close enough. If the difference between the bounds is smaller than ϵ, then the algorithm terminates and the current solution is returned. Otherwise, we go back to line 2 to repeat the procedure until the bounds converge.

Algorithm 1: Pseudo-code for the overall decomposition algorithm

```
1  Initialize $M = 0$ and $\Phi = \emptyset$.
2  Solve master problem (10)-(13) to obtain $\mathbf{x}^* = [x_i^*]$, $[x_o^{C*}]$ and $[x_{ik}^{N*}]$.
3  Set the lower bound $LB$ as the optimal objective function of the master
     problem.
4  Set $\mathbf{x}^{M+1} = [x_i^{M+1}] = \mathbf{x}^*$, $\Phi = \Phi \bigcup \{M+1\}$ and $M = M+1$.
5  Use $\mathbf{x}^M$ to solve the subproblems (14)-(16) to obtain $[x_{ij}^{H*}]$ and $[x_{il}^{L*}]$.
6  Compute the upper bound
     $UB = \max_i \rho f_i(x_i^*) \prod_{o \in \Psi_i} P_o^C(x_o^{C*}) + \sum_k \omega_k \sum_i P_{ik}^N(x_{ik}^{N*}) \sum_j C_{ij}$.
7  if $(UB - LB) \leq \epsilon$ then
8      Return the current solution as the optimal solution of the problem.
9      Terminate the procedure.
10 else
11     Go to Line 2.
12 end
```

Remark 4.1. *At every iteration of the decomposition algorithm, the values of $f_i(x_i^m)$ and $f_i'(x_i^m)$ give an aggregation of the asset-level data for each city i. Using these values, one can compare the cost effectiveness of investments in different cities with* *differing numbers of assets (and differing asset values). Specifically, at the current level of investments, a lower bound on the expected damage from man-made disasters in city i is given in the form of $C_i e^{-\lambda_i (x_i - x_i^m)}$, where $\lambda_i = -\frac{f_i'(x_i^m)}{f_i(x_i^m)}$ and $C_i = f_i(x_i^m) \prod_{o \in \Psi_i} P_o^C(x_o^C)$. This bound is tight at the current level of investments. Moreover, λ_i can be interpreted as the cost effectiveness of the new investments in city i. For example, if $\lambda_i = 0.01$, an extra unit of investment in city i will lead to a reduction of about 1% in the expected damage in city i.*

Remark 4.2. *Note that a similar decomposition approach can be developed for the case in which, instead of using a budget constraint, investment costs are added to the objective function.*

4.4 Numerical Experiments

In this section, we perform computational experiments to investigate the efficiency of the proposed algorithm and to gain insight into the properties of the game. The algorithms are coded in GAMS and the IPOPT (Interior Point OPTimizer) solver has been used to solve the NLPs. The computational experiments are performed on a computer with 2.6 GH processor and 32 GB of RAM. Throughout this section, unless mentioned otherwise, we use the following parameter values. Similar to [49], power-law functions represent the success probability of an attack and the failure probability of protection against natural hazards. Specifically, assume $P_{ij}^H(x_{ij}^H) \equiv \left(\frac{\alpha_{ij}^H}{\alpha_{ij}^H + x_{ij}^H}\right)^{\kappa_{ij}^H}$, where α_{ij}^H and κ_{ij}^H are positive-valued parameters that determine the cost effectiveness of defensive investment. Similarly, let $P_{il}^L(x_{il}^L) \equiv \left(\frac{\alpha_{il}^L}{\alpha_{il}^L + x_{il}^L}\right)^{\kappa_{il}^L}$, $P_o^C(x_o^C) \equiv \left(\frac{\alpha_o^C}{\alpha_o^C + x_o^C}\right)^{\kappa_o^C}$, and $P_{ik}^N(x_{ik}^N) \equiv \left(\frac{\alpha_{ik}^N}{\alpha_{ik}^N + x_{ik}^N}\right)^{\kappa_{ik}^N}$. In addition, assume $\kappa_{ij}^H = \kappa_{il}^L = \kappa_o^C = \kappa_{ik}^N = 7$, $\alpha_{ij}^H = 0.01$, $\alpha_{il}^L = 0.1$, and $\alpha_o^C = \alpha_{ik}^N = 1$. The acceptable gap of the optimum objective function value, ϵ, is set as 0.001. Thus, in all experiments, the run time represents the time it takes the algorithm to reach a gap of less than or equal to ϵ. Furthermore, assume that all types of disasters are equally likely to happen with a probability of 0.001.

Table 4.1: Average run times (in seconds) of the decomposition approach (DA) and the direct optimization (DO) of the mathematical model

I	$J = 200$		$J = 250$		$J = 300$		Mean	
	DO	DA	DO	DA	DO	DA	DO	DA
100	71.79	**30.49**	79.73	**38.33**	97.19	**46.74**	82.91	**38.52**
150	151.76	**48.03**	153.11	**58.42**	190.24	**72.19**	165.04	**59.55**
200	189.37	**67.79**	258.04	**81.06**	264.86	**97.50**	237.42	**82.12**
Mean	137.64	**48.77**	163.63	**59.27**	184.10	**72.14**	161.79	**60.06**

In the first experiment, we compare the performance of the decomposition approach with directly solving the mathematical model. We generate the instances for these experiments randomly. Asset values are uniform random variables in the range $[43, 56]$. This range includes the minimum and maximum risk scores given in the case study by Haphuriwat and Bier [49]. We use $L1$ and $L2$ to denote the number of country-level and city-level overarching protections, respectively. For all possible combinations of $I \in \{100, 150, 200\}$, $J \in \{200, 250, 300\}$, $K, L1, L2 \in \{10, 15, 20\}$, we generated an instance of the problem to obtain a data set of 243 problem instances. We then used our proposed decomposition approach as well as the direct optimization approach to solve all of these problem instances. Table 4.1 exhibits the average run times for different number of cities (I) and number of assets in each city ($J_i = J, \forall i = 1, \ldots, I$). The columns DA and DO show the average run times for the decomposition approach and the direct optimization method, respectively. As seen in this table, the decomposition approach performs significantly better than the direct optimization approach. This table also reveals that the run times for both DA and DO increase as the number of cities increases. Moreover, the run times also increase as the number of assets inside each city increases.

Table 4.2: Average run times (in seconds) of the decomposition approach (DA) and the direct optimization (DO) of the mathematical model

I	$K = 10$		$K = 15$		$K = 20$		Mean	
	DO	DA	DO	DA	DO	DA	DO	DA
100	83.42	**38.80**	84.77	**38.60**	80.53	**38.16**	82.91	**38.52**
150	149.47	**59.24**	194.48	**59.25**	151.17	**60.16**	165.04	**59.55**
200	240.14	**81.28**	237.35	**82.90**	234.77	**82.18**	237.42	**82.12**
Mean	157.68	**59.77**	172.20	**60.25**	155.49	**60.16**	161.79	**60.06**

Table 4.2 shows the average run times for different number of cities and number of natural disasters (K). The columns DA and DO present the average run times for the decomposition approach and the direct optimization method, respectively. The decomposition approach performs significantly better than the direct optimization approach. The run times for DO increase as the number of natural disaster types increases. However, the number of natural disaster types does not seem to influence the run times of DA.

Table 4.3 exhibits the average run times for different number of country-level (L1) and city-level overarching protections (L2). The columns DA and DO show the average run times for the decomposition approach and the direct optimization approach, respectively. The decomposition approach performs significantly better than direct optimization of the mathematical model. In general, the run times for both DA and

Table 4.3: Comparison of the decomposition approach with the mathematical model

L1	L2=10		L2=15		L2=20		Mean	
	DO	DA	DO	DA	DO	DA	DO	DA
10	123.90	**48.09**	148.76	**62.69**	176.83	**74.43**	149.83	**61.74**
15	129.58	**46.12**	152.57	**61.59**	180.66	**71.84**	154.27	**59.85**
20	178.34	**45.45**	165.37	**60.47**	200.09	**69.90**	181.27	**58.60**
Mean	143.94	**46.55**	155.57	**61.58**	185.86	**72.05**	161.79	**60.06**

DO increases as the number of city level overarching protections increases. Increasing the number of country-level overarching protections, leads to an increase in the run times of DO but decreases the run times of DA.

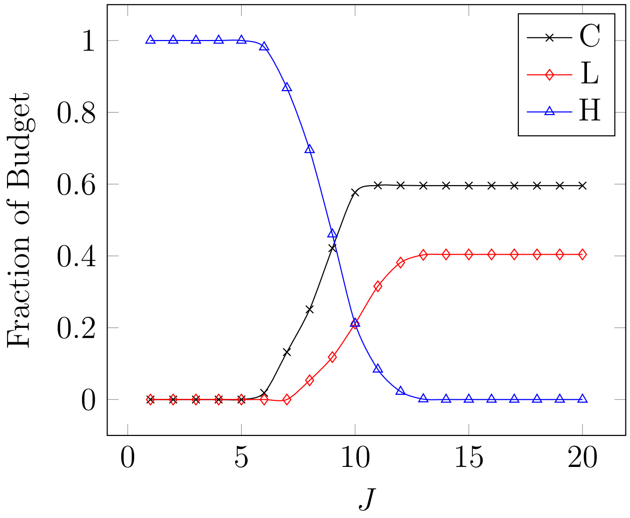


Figure 4.1: The effect of number of assets per city on the optimal resource allocation

In our next experiment, we study the effect of some of the model parameters on the optimal resource allocation. For this experiment, we assume that all assets have the same valuations, i.e., $C_{ij} = 1$. We consider 5 randomly generated country-level overarching protection options. We also assume that, for each city, there is only one city-level overarching protection option, and it covers all of the assets inside the city. Figure 4.1 demonstrates the effect of number of assets per city on the optimal allocation of resources. In this figure, C and L refer to the portion of the budget that has been assigned to country-level and city-level overarching protection options, respectively. Moreover, H refers to the proportion of the budget that has been assigned to individual target hardening. According to this figure, as J increases, the resource amount allocated to individual target hardening decreases and the resources are shifted toward the overarching protection types. This is in line with existing observations in the literature. Moreover, as J increases, the optimal resource allocation levels converge and after a certain point, the optimal allocation of resources does not change.

Figure 4.2 displays the effect of number of cities on the optimal allocation of resources. In this figure, the optimal resource level for country-level overarching protection options increases as I increases. However, as I increases, the amount of resource allocated to city-level overarching protection options decreases. Moreover, the effect of I on the amount of resource allocated to target hardening options is not monotonic. Specifically, as I increases, the amount of resource allocated to target hardening options increases at first, then decreases.

The effect of I on the resource allocated to different protection options depends on the cost efficiency of these options. For example, Figure 4.3 shows the effect of number of cities on the optimal allocation of resources, for the case with $\alpha_{ij}^{L} = 0.075$ parameter. This is a slight change from Figure 4.2, in which we had $\alpha_{ij}^{L} = 0.1$. As seen in this figure, the effect of I on the amount of resource allocated to city-level overarching protections and target hardening is different from Figure 4.2. This highlights the importance of having accurate estimates of the parameters that determine the cost efficiency of the protection options.

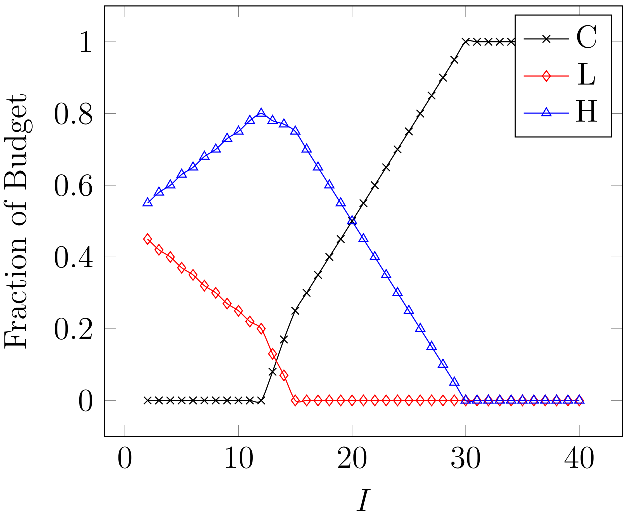


Figure 4.2: The effect of number of cities on the optimal resource allocation for $\alpha_{ij}^{L} = 0.1$

Figure 4.4 exhibits the effect of α_{ij}^{H} on the optimal resource allocation. As seen in this figure, as α_{ij}^{H} increases, the amount of resource allocated to harden individual targets decreases while amounts allocated to city-level and country-level overarching options both increase. This is due to the fact that, as α_{ij}^{H} increases, the cost efficiency of target hardening decreases. Therefore, allocating resources to other protection options becomes more appealing.

Figure 4.5 displays the effect of α_{o}^{C} on the optimal resource allocation. As α_{o}^{C} increases, the amount of resources allocated to country-level overarching options decreases and the amount allocated to other protection options increases. This is due to the fact that, as α_{o}^{C} increases, the cost efficiency of country-level overarching options decreases. Therefore, allocating resources to other protection options becomes more appealing.

In many realistic situations, the cities are not identical and they differ in the number and valuation of their assets. In such cases, an interesting question to address is how to compare the cost effectiveness of investments in different cities. In this experiment, we highlight the ability of the decomposition approach to ag-

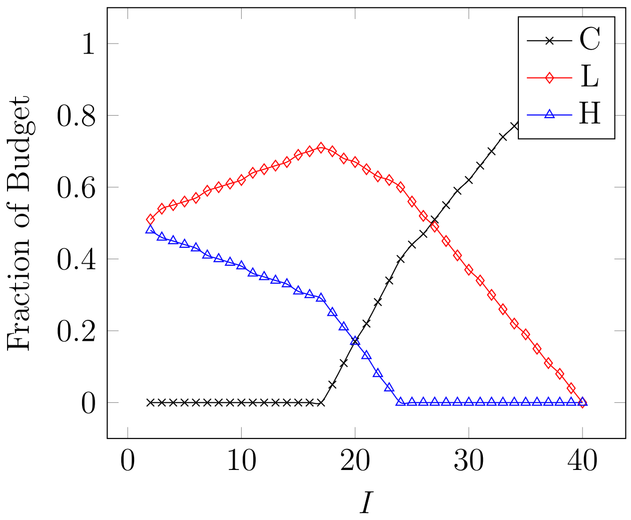


Figure 4.3: The effect of number of cities on the optimal resource allocation for $\alpha_{ij}^{L} = 0.075$

gregate asset-level data to compare cities in terms of cost effectiveness. For this experiment, we ignore the natural disasters and assume that there are no overarching protection options. Therefore, all of the available resources will be assigned to individual target hardening against intentional attacks. Given a budget of 100 units, we consider two cities; the first one has 10 assets with the values given as $(C_{1,1}, C_{1,2}, \ldots, C_{1,10}) = (10, 10, 5, 8, 10, 10, 7, 5, 10, 10)$. Moreover, we have $(\alpha_{1,1}^{H}, \alpha_{1,2}^{H}, \ldots, \alpha_{1,10}^{H}) = (1, 3, 4, 2, 4, 1, 5, 3, 4, 3)$. The other city has only one asset with $C_{2,1} = 5$ and $\alpha_{2,1}^{H} = 2$. Using the decomposition approach we solve this instance of the problem and obtain the expected damage from an intentional attack as 1.94 units under the optimal resource allocation policy. The optimal policy is to assign 96.84 units to the first city and 3.16 units to the second city. At the optimal solution, we have $f_1(x_1) = f_2(x_2) = 1.94$, $f_1'(x_1) = -0.0153$ and $f_2'(x_2) = -0.3761$. These values aggregate the asset-level data and enable us to compare these two cities in terms of the current expected damage due to an attack and the cost effectiveness of new investments. Both cities have the same level of expected damage from an attack, $f_1(x_1) = f_2(x_2) = 1.94$.

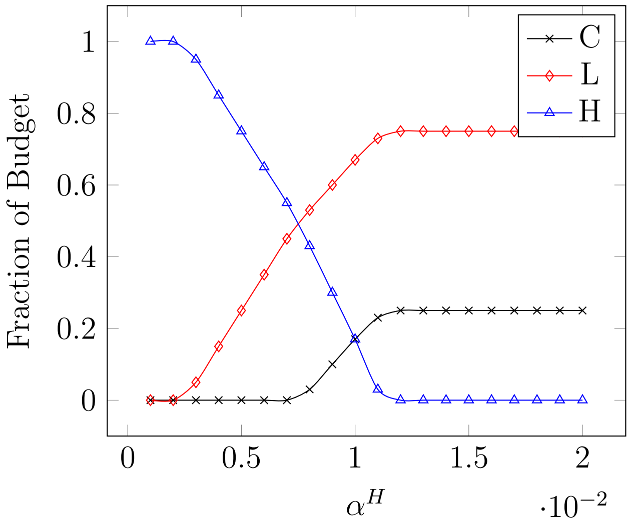


Figure 4.4: The effect of α_{ij}^H on the optimal resource allocation

The decomposition approach also gives us an idea about the cost effectiveness of investment in each city. Because the first city has more assets, which in general have higher values than the asset in the second city, we expect protecting the first city to be more costly than protecting the second city. In other words, for each unit of extra investment, we expect the rate of reduction in the expected damage for the first city to be smaller than for the second city. However, quantifying the difference is not a trivial task. The decomposition approach offers a way to address this issue. Specifically, the values $f_1'(x_1) = -0.0153$ and $f_2'(x_2) = -0.3761$ give us an idea about the cost effectiveness of the two cities for new investments. Based on these numbers, protecting the first city is roughly 24 times more costly than protecting the second city.

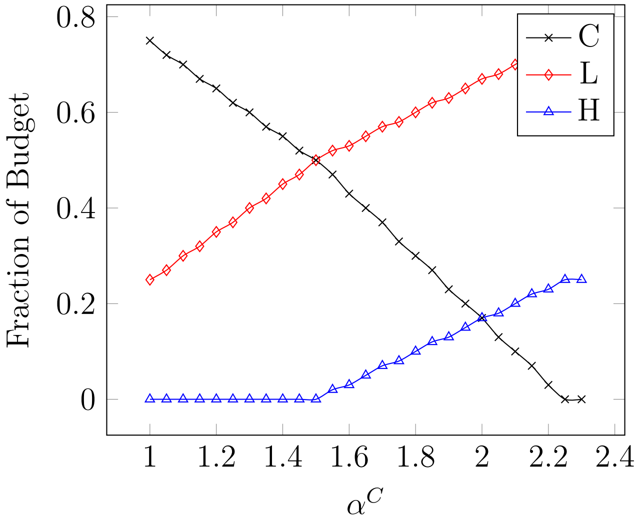


Figure 4.5: The effect of α_o^C on the optimal resource allocation

Part II

Patrolling and Search Games

Chapter 5

Patrolling Games on General Graphs with Time-Dependent Node Values

5.1 Introduction and Literature Review

One of the most important issues in homeland security is protecting critical infrastructures against terrorist attacks [104]. Among these infrastructures, transportation systems, serving 32 million passengers daily in the United States, are critical for supporting the national security and economic well-being. Public surface transportation systems such as trains, metros, subways and buses offer terrorists easy access to crowds of people. This makes them especially attractive to terrorists seeking high body counts. Such open systems are considered to be soft targets by the terrorists. Bombings in Brussels and Istanbul along with many other cases indicate that terrorists tend to target such large crowds to cause mass human casualties in addition to panic and chaos. Therefore, it is important to protect such infrastructures.

Analyzing the risk associated with attack to each infrastructure component, mitigation planning and designing efficient response policies could substantially reduce the threat to these infrastructures. One component of such planning is designing efficient patrols to secure vulnerable areas. One of the challenges involved in designing patrol schedules to safeguard open mass transit systems and other soft targets is time-dependent node values. Because the adversary's primary objective is to inflict human casualties, the node values depend on the number of people residing in those nodes. These numbers change over time and the terrorists tend to time their attacks according to these changes [68]. Another challenge is to develop efficient methods to design patrols for a general network. In this chapter, we try to address these challenges in a patrolling game setting.

Patrolling problems arise in many situations in real life. Police officers patrol cities; security officers patrol terminals at airports and transportation centers; security guards patrol museums and shopping malls. Patrolling problems involve decisions on how to route a patroller through many locations in order to safeguard the area from adversarial intruders or illicit activity. With recent advancements in technology, patrolling decisions arise even more frequently with applications in routing unmanned aerial vehicles and robots.

The patrolling problems have been studied since 1970s. Several studies have focused on allocating patrols to different areas to optimize performance measures such as patrol delays, average waiting time and total response time [24, 25, 82, 109]. These studies assume that crime frequency in different regions remain fixed and known to the patroller. However, this is not a realistic assumption due to the strategic behaviour of the adversaries. In other words, the adversaries can change their strategy in response the patroller's strategy. Therefore, game theoretic analysis of such problems yields more realistic results. Basilico *et al.* [14] introduce a two-player multi-stage security game with an underlying infinite horizon setting in which there are potentially infinitely many decision nodes. In this model, the attacker has the complete knowledge of the strategy to which the patroller committed to. The attacker can also observe the location and movements of the patroller at any time and chooses his best attack strategy based on this information. They study Markovian strategies of different orders for this problem and show that, even though first order Markovian strategies may not always be optimal, they have comparable quality with respect to higher order Markovian strategies. Basilico *et al.* [13] consider a similar patrolling game model with the patroller employing spatially uncertain alarm signals. They prove that this problem is NP-hard for a general graph, they also show that for special graphs, like paths or cycle graphs, the optimal strategy can be found in polynomial time. Infinite horizon nature of the games studied in [14] and [13] leads to the application of stationary Markovian strategies by the patroller. This means that the timing of attacks becomes irrelevant in such games. However, this may not be valid in realistic situations, for example, when node values change over time.

Alpern *et al.* [10] introduce a finite horizon patrolling game played on a graph, Q, with n nodes. The game has two players, a defender patrolling a set of nodes on Q, and an adversary targeting a node to attack. The adversary needs m consecutive periods, uninterrupted by the defender, to successfully damage the node. He aims to maximize the probability of a successful attack, while the defender tries to minimize this probability. Hence, the proposed model is a zero-sum game and the solution to this game is called a saddle point [29]. Papadaki *et al.* [111] study the same patrolling problem on a line graph. They solve this patrolling game for any values of m and n, to find a saddle point.

Both [10] and [111] assume that all nodes have the same value, they also assume that the attack time, m, is fixed and does not depend on the node under attack. However, as we have discussed earlier in this section, these assumptions may not be valid in reality. Especially in public transportation systems and other soft targets when node values represent the number of people present, called occupancy level, different nodes may have different values and these values may change over time. Morevover, some nodes may be harder to attack than others, therefore, it may take more time to launch a successful attack.

Lin *et al.* [93] attempt to address this gap by studying patrolling models with different node values and attack time distributions. They consider both random and strategic attackers. A random attacker uses a fixed and known probability distribution to launch attacks on nodes; while a strategic attacker plays a zero-sum game with the patroller. The authors develop linear programming models to find the optimal solutions for both players. They also propose index-based heuristics to solve the problems of larger size. Lin *et al.* [94] extend the model of [93] by allowing imperfect detection. In other words, there is a possibility of observing a false negative when the patroller inspects a node. They introduce efficient index based heuristics to obtain near optimal policies in a reasonable amount of time. Although [93, 94] resolve some of the shortcomings of the previous models, their models still do not consider time-dependent node values and the importance of the timing of attacks.

There are a number of studies accommodating multiple patrollers in their models. Jain *et al.* [67] study Stackelberg security games with arbitrary schedules and multiple patrollers. They develop a branch and price algorithm to efficiently solve this game. Their algorithm involves a column generation step that exploits a novel network flow representation avoiding the combinatorial explosion of schedule assignments. Korzhyk *et al.* [77] investigate the case with multiple defenders and multiple attackers where the attacker can attack multiple targets. They propose a polynomial time algorithm to find the Nash equilibrium for this game. Hochbaum *et al.* [64] consider a patrolling problem with multiple patrollers (vehicles) on a network with edges targeted by a strategic adversary. They present a novel decomposition approach that requires the solution of a multivehicle rural Chinese postman problem [46]. Lou *et al.* [96] model a security game with multiple decentralized defenders in charge of defending disjoint subsets of, possibly interdependent, targets. They analyze the existence of a Nash equilibrium for this game under various conditions. Lagos *et al.* [79] study a Stackelberg security problem with multiple patrols and multiple targets. They propose a branch and price approach to efficiently solve this problem. McGrath and Lin [101] investigate a patrol problem with multiple patrollers and dispersed heterogeneous attack locations. Their model accounts for the travel time between nodes, and includes node-specific features such as the inspection time, the time required for the adversary to carry out an attack, and the cost of a successful attack. They show that, for the case of a single patroller, the optimal solution can be obtained by solving a linear program. For the case with multiple patrollers, they propose heuristic solutions based on shortest paths and set partitions.

Majority of the papers in the literature of patrolling games assume that a single adversary chooses a target to attack, the target values are fixed over time, and some even assume that all targets are indistinguishable, i.e., they all have the same value. However, this is not the case in many realistic situations. For example, at a transportation facility, the number of people, occupancy level, at each location may be considered as the value of that location. Moreover, occupancy levels may change over time, it is expected that during the rush hours the occupancy levels would be higher than normal hours. In this chapter, we study a patrolling game model with time-dependent node values, node-specific attack times, multiple patrollers and multiple attackers. We propose a solution approach to efficiently solve the game under general graphs. The computational results show the efficiency of the proposed approach. The rest of this chapter is organized as follows. In section 5.2, the problem under consideration is described. The proposed solution approach is explained in section 5.3. Section 5.4 presents the computational results.

5.2 Proposed Model

In this section, we describe the problem under consideration. Our work extends the model of [10] by considering different and time-dependent node values, node-specific attack times, multiple patrollers and multiple attackers. Here is a list of parameters of the model:

- N : Number of nodes.
- $\mathcal{N} = \{1, 2, \ldots, N\}$: Set of nodes.
- $i, j \in \mathcal{N}$: Node indices.
- $s \in \mathcal{S}$: Index of patrollers (security personnel).
- $a \in \mathcal{A}$: Index of attackers.
- T : Number of patrolling time periods.
- $\mathcal{T} = \{0, 1, \ldots, T-1\}$: Set of time periods in the time horizon.
- $t, \tau \in \mathcal{T}$: Time period indices.
- m_i : Attack time, consecutive number of periods needed to attack node i. Let $\mathbf{m} = (m_1, m_2, \ldots, m_N)$.
- $\mathcal{E}$: Set of edges, where $(i, j) \in \mathcal{E}$ if there is an edge between nodes i and j.
- C_{it} : Value of node i at time t. Let $\mathbf{C} = [C_{it}]$ be a $N \times T$ matrix containing all of C_{it} values, with element in row i and column t being C_{it}.

The patrolling game $G = G(Q, T, \mathbf{m}, \mathbf{c})$ introduced in this chapter is a zero-sum game between a defender (she) and an adversary (he). The defender controls a set of patrollers $\mathcal{S}$. The adversary controls a set of attackers $\mathcal{A}$. The game is played on a connected graph $Q = (\mathcal{N}, \mathcal{E})$ with the set of nodes $\mathcal{N}$ and the set of edges $\mathcal{E}$ over the time horizon $\mathcal{T}$.

A pure strategy for the adversary is to select a pair (i^a, I^a) for each attacker, $a \in \mathcal{A}$, where $i^a \in \mathcal{N}$ is the target node and I^a is the attack interval defining the beginning, τ^a, and the end of the attack, which is a set of m_j consecutive time periods, i.e., $I^a = \{\tau^a, \tau^a + 1, \ldots, \tau^a + m_j - 1\} \in \mathcal{T}$, where $j = i^a$. We can also represent an attack strategy as (i^a, τ^a). Note that, for each attacker a, the start of attack interval, τ^a, should be early enough for the attack interval to finish before the end of the time horizon $\mathcal{T}$, i.e., $\tau \leq T - m_j$, where $j = i^a$. We assume that the adversary cannot assign an attack pair to more than one attacker.

We define a patrol as a walk $P : \mathcal{T} \to Q$ on graph Q during the time horizon $\mathcal{T}$. A pure strategy for the defender is to select a patrol P^s for each patroller $s \in \mathcal{S}$. If $i^a \in P^s(I^a)$ for some $s \in \mathcal{S}$, i.e., patroller d interrupts the attacker a, the attack will be unsuccessful. Otherwise, if $i^a \notin P^s(I^a) \quad \forall s \in \mathcal{S}$ the attacker a successfully damages node i^a, the adversary gains a payoff of C_{j,τ^a+m_j-1}, where $j = i^a$, and the defender loses a payoff of C_{j,τ^a+m_j-1}. The defender aims to minimize the expected total damage incurred from all attackers and the adversary wants to maximize it.

The players play a zero-sum matrix game with the defender playing as the row player and the set of all possible defense strategies constituting the rows of the matrix. The adversary plays as the column player, with the set of all possible attack strategies constituting the columns of the game matrix. We use $\mathcal{K}$ to denote the set of all possible defense strategies and k to index them. Let x_k be the probability of using defense strategy k in the defender's mixed strategy. Hence $\mathbf{x} = (x_1, x_2, \ldots, x_{|\mathcal{K}|})$

represents a mixed strategy of the defender, where $|\mathcal{K}|$ denotes the cardinality of $\mathcal{K}$, $x_k \geq 0 \; \forall k \in \mathcal{K}$ and $\sum_{k=1}^{k=|\mathcal{K}|} x_k = 1$. Similarly, we use $\mathcal{L}$ to denote the set of all possible attack strategies and index them by l. Let y_l denote the probability of using attack strategy l. Hence, a mixed strategy of the adversary is denoted as $\mathbf{y} = (y_1, y_2, \ldots, y_{|\mathcal{L}|})$, $y_l \geq 0 \; \forall l \in \mathcal{L}$, and $\sum_{l=1}^{l=|\mathcal{L}|} y_l = 1$. The saddle point (Nash equilibrium) of the game is a point $(\mathbf{x}^*, \mathbf{y}^*)$ at which the following inequalities hold:

$$v(\mathbf{x}^*, \mathbf{y}) \leq v(\mathbf{x}^*, \mathbf{y}^*) \leq v(\mathbf{x}, \mathbf{y}^*),$$

where $v(\mathbf{x}, \mathbf{y})$ is the expected damage if the defender and the adversary use mixed strategies $\mathbf{x}$ and $\mathbf{y}$, respectively.

Although our model is a generalization of the model proposed by [10], some of their results are still valid for our model. We will use the following lemma that has been proved in [10] directly since the proof does not depend on the node values.

Lemma 5.1. *Suppose Q is connected, $T \geq 3$ and $m_i \geq 2, \forall i$. Then patrols that stay on any node for three consecutive periods are dominated.*

The game can be solved by generating all of the possible strategies for both players, however, this may not be efficient for games of larger size. In the next section, we develop a solution approach to obtain a saddle-point equilibrium for this game.

5.3 Solution Procedure

In this section, a solution algorithm based on column and row generation [105, 122] is developed to obtain a saddle point for the patrolling game described in the previous section. The main challenge that may arise when developing a column and row generation algorithm is that the structure of the subproblems may be destroyed due to the addition of new rows [12]. However, in our case, since the new rows only affect the objective function coefficients, this difficulty does not arise. The solution method can also be described as a modification of the algorithm proposed by [42, 43].

Because this is a zero-sum game, a linear program (LP) can be developed to obtain a saddle-point equilibrium of this game. To formulate the LP for this game, we use the following binary parameters:

- $w^k_{i\tau} = \begin{cases} 1 & \text{if defense strategy } k \text{ interrupts attack pair } (i, \tau), \\ 0 & \text{Otherwise.} \end{cases}$

- $z^l_{i\tau} = \begin{cases} 1 & \text{if attack strategy } l \text{ involves attack pair } (i, \tau), \\ 0 & \text{Otherwise.} \end{cases}$

Using this notation, the following LP formulation can be developed to obtain a saddle-point equilibrium for this game:

$$\begin{aligned} \text{Minimize} \quad & u \\ \text{subject to} \quad & u \geq \sum_{k \in \mathcal{K}} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^k_{i\tau}) x_k, \quad \forall l \in \mathcal{L}, \\ & \sum_{k \in \mathcal{K}} x_k = 1, \\ & x_k \geq 0, \quad \forall k \in \mathcal{K}. \end{aligned}$$

This problem is called the linear programming master (LPM). In this formulation, x_k is a decision variable representing the probability of using defense strategy $k \in \mathcal{K}$ in the defender's mixed strategy. In general, the sets $\mathcal{K}$ and $\mathcal{L}$ may be exponentially large; however, the number of used strategies is expected to be much smaller. The proposed solution algorithm uses this idea to start with a small subsets $\mathcal{K}' \subset \mathcal{K}$ and $\mathcal{L}' \subset \mathcal{L}$ of defense and attack strategies and generates them as needed. In other words, we generate the defense strategies (columns) and attack strategies (rows) on the fly.

The starting subsets $\mathcal{K}'$ and $\mathcal{L}'$ could be any set of strategies. Using the restricted set of strategies $\mathcal{K}'$ and $\mathcal{L}'$, we obtain the following LP:

$$\text{Minimize} \quad u \tag{5.1}$$

$$\text{subject to} \quad u \geq \sum_{k \in \mathcal{K}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^k_{i\tau}) x_k, \quad \forall l \in \mathcal{L}', \tag{5.2}$$

$$\sum_{k \in \mathcal{K}'} x_k = 1, \tag{5.3}$$

$$x_k \geq 0, \quad \forall k \in \mathcal{K}'. \tag{5.4}$$

This problem is called Restricted LPM (RLPM). The dual of RLPM is:

$$\text{Maximize} \quad v \tag{5.5}$$

$$\text{subject to} \quad v \leq \sum_{l \in \mathcal{L}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^k_{i\tau}) y_l, \quad \forall k \in \mathcal{K}', \tag{5.6}$$

$$\sum_{l \in \mathcal{L}'} y_l = 1, \tag{5.7}$$

$$y_l \geq 0, \quad \forall l \in \mathcal{L}'. \tag{5.8}$$

In this formulation, y_l is the dual variable corresponding to constraint (5.2) in RLPM. This variable represents the probability of using attack strategy l in the adversary's mixed strategy. Moreover, v is the dual variable corresponding to constraint (5.3) which represents the minimum expected damage. Next step is to find new strategies in $\mathcal{K} \setminus \mathcal{K}'$ and $\mathcal{L} \setminus \mathcal{L}'$ that could improve the current optimal solution for the corresponding players. Given the optimal dual solution y_l of RLPM, the reduced cost of defense strategy $k \in \mathcal{K} \setminus \mathcal{K}'$ is given by $\sum_{l \in \mathcal{L}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^k_{i\tau}) y_l - v$. Based on the concept of duality in linear programming, optimality of RLPM is equivalent to the feasibility of its dual. Therefore, defense strategies that violate the constraint $v \leq \sum_{l \in \mathcal{L}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^k_{i\tau}) y_l$ can improve the current optimal solution. Thus, one should look for a defense strategy k with $w^k_{i\tau}$ such that: $v > \sum_{l \in \mathcal{L}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^k_{i\tau}) y_l$. Note that y_l's are fixed, and the problem is to find a defense strategy k with $w^k_{i\tau}$ such that $v > \sum_{l\in\mathcal{L}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1-w^k_{i\tau})y_l$. In other words, one looks for a new defense strategy k that leads to a smaller expected total damage, $\sum_{l\in\mathcal{L}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1-w^k_{i\tau})y_l$, than the current expected total damage, v. To obtain an improving attack strategy for the adversary, consider RLPM. Given the optimal solution x_k of RLPM, the current total expected damage is u. The total expected damage incurred by using attack strategy l is $\sum_{k\in\mathcal{K}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1-w^k_{i\tau})x_k$. Therefore, for fixed values of x_k, one should find a new attack strategy l with $z^l_{i\tau}$ such that $\sum_{k\in\mathcal{K}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1-w^k_{i\tau})x_k > u$. In the following subsections, we develop mathematical programs to solve the players' subproblems, and describe the overall solution algorithm.

5.3.1 Mathematical Formulations for the Defender's Subproblem

In this section, we present two mathematical formulations to solve the defender's subproblem: A hop-type formulation and a flow-type formulation. We will compare the performance of these formulations numerically in section 5.4. Here is a list of binary variables used to formulate the defender's subproblem:

- $v^s_{it} = \begin{cases} 1 & \text{if patroller } s \text{ visits node } i \text{ at time } t, \\ 0 & \text{Otherwise.} \end{cases}$

- $w_{i\tau} = \begin{cases} 1 & \text{if a patroller visits node } i \text{ at time interval } [\tau, \tau+m_i-1], \\ 0 & \text{Otherwise.} \end{cases}$

Using this notation, the following hop-type formulation can be developed for the defender's subproblem:

$$\text{Maximize} \quad \sum_{i,\tau} C_{i,\tau+m_i-1} w_{i\tau} \sum_{l\in\mathcal{L}'} y_l z_{i\tau}^l \tag{5.9}$$

$$\text{subject to} \quad w_{i\tau} \leq \sum_{s\in\mathcal{S}} \sum_{t=\tau}^{\tau+m_i-1} v_{it}^s, \quad \forall i,\tau, \tag{5.10}$$

$$\sum_{i=1}^{N} v_{it}^s = 1, \quad \forall t,s, \tag{5.11}$$

$$v_{it}^s + v_{j,t+1}^s \leq 1, \quad \forall i,j,t,s | i \neq j, (i,j) \notin \mathcal{E}, \tag{5.12}$$

$$w_{i\tau} \in \{0,1\}, \quad \forall i,\tau, \tag{5.13}$$

$$v_{it}^s \in \{0,1\}, \quad \forall i,t,s. \tag{5.14}$$

In this formulation, equation (5.9) represents the objective function which is minimizing the expected damage. Note that, the expected damage is equal to $\sum_{l\in\mathcal{L}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z_{i\tau}^l (1-w_{i\tau}) y_l = \sum_{l\in\mathcal{L}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z_{i\tau}^l y_l - \sum_{l\in\mathcal{L}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z_{i\tau}^l w_{i\tau} y_l$ where the first term is constant; hence, minimizing the expected damage is equivalent to maximizing $\sum_{l\in\mathcal{L}'}\sum_{i,\tau} C_{i,\tau+m_i-1} z_{i\tau}^l w_{i\tau} y_l = \sum_{i,\tau} C_{i,\tau+m_i-1} w_{i\tau} \sum_{l\in\mathcal{L}'} y_l z_{i\tau}^l$. The term $\sum_{l\in\mathcal{L}'} y_l z_{i\tau}^l$ in the objective function can be interpreted as the probability of using attack pair (i,τ) by the adversary. Equation (5.10) ensures that, if no patroller interrupts attack pair (i,τ), then $w_{i\tau}$ is equal to zero. Equation (5.11) indicates that, for each patroller s, at each time t the patroller can be at exactly 1 node. Equation (5.12) ensures that, the patroller can not move from node i to node j if there is no edge between these nodes. Constraints (5.13) and (5.14) are the integrality constraint for variables $w_{i\tau}$ and v_{it}^s.

Note that, lemma 5.1 can be used to incorporate a new constraint to this formulation. Specifically, the constraint $v_{i,t}^s + v_{i,t+1}^s + v_{i,t+2}^s \leq 2, \quad \forall i,t,s$ can be added to the formulation to eliminate the patrols that stay in the same node for three consecutive time periods. In the numerical experiments section, we will study the effect of this constraint on the overall performance of the algorithm.

A flow-type mathematical formulation can also be developed to solve the defender's subproblem. The subproblem is formulated as follows:

$$\text{Maximize} \quad \sum_{i,\tau} C_{i,\tau+m_i-1} w_{i\tau} \sum_{l\in\mathcal{L}'} y_l z^l_{i\tau} \tag{5.15}$$

$$\text{subject to} \quad w_{i\tau} \leq \sum_{t=\tau}^{\tau+m_i-1} \sum_j f^t_{ij}, \quad \forall i,\tau, \tag{5.16}$$

$$\sum_{i,j} f^0_{ij} = |\mathcal{S}|, \tag{5.17}$$

$$\sum_i f^t_{ij} = \sum_l f^{t+1}_{jl}, \quad \forall j,t, \tag{5.18}$$

$$w_{i\tau}, \in \{0,1\}, \quad \forall i,\tau, \tag{5.19}$$

$$f^t_{ij}, \in \mathbb{Z}^+, \quad \forall i,j,t. \tag{5.20}$$

In this formulation, f^t_{ij} is an integer variable that represents the flow of patrollers from node i to node j at time t. Equation (5.15) presents the objective function, which is identical to the objective function in equation (5.9). Constraint (5.16) ensures that, if no patroller interrupts (i,τ) attack pair, then $w_{i\tau}$ is equal to zero. Constraint (5.17) indicates that, the initial flow of patrollers should be equal to the number of available patrollers, i.e. $|\mathcal{S}|$. Constraint (5.18) is the flow conservation constraint. It ensures that, for each node j, the total incoming flow at time t is equal to the outgoing total flow at time $t+1$. Constraints (5.19) and (5.20) are the integrality constraints for variables $w_{i\tau}$ and f^t_{ij}, respectively. The flows obtained from this formulation can be transformed into patrols using the well-known flow decomposition algorithm [110].

Theorem 5.1. *The defender's subproblem is NP-hard.*

Proof. See Appendix A.8 □

Remark 5.1. *Theorem 5.1 is valid even if $C_{i\tau} = 1, \forall i, \tau$ and m_is are equal to each other. In other words, even if we solve the model proposed by [10] using our proposed* *solution method, under a general graph, the defender's subproblem will still remain NP-hard.*

Even though Theorem 5.1 indicates that the defender's subproblems are hard to solve, the computational results show that, for problems with up to 30 nodes, the solution algorithm is able to find the Nash equilibrium by directly solving the subproblem formulations.

5.3.2 Mathematical Formulation for the Adversary's Subproblem

The adversary's subproblem is formulated as follows:

$$\text{Maximize} \quad \sum_{i,\tau} C_{i,\tau+m_i-1} z_{i\tau} \sum_{k \in \mathcal{K}'} (1 - w_{i\tau}^k) x_k \tag{5.21}$$

$$\text{subject to} \quad \sum_{i,\tau} z_{i\tau} \leq |\mathcal{A}|, \tag{5.22}$$

$$z_{i\tau} \in \{0, 1\}, \quad \forall i, \tau. \tag{5.23}$$

In this formulation, $z_{i\tau}$ is a binary variable equal to 1 if an attacker is assigned attack pair (i, τ). Equation (5.21) presents the objective function, which is maximizing the expected damage. In this expression, the term $\sum_{k \in \mathcal{K}'}(1 - w_{i\tau}^k)x_k$ can be interpreted as the probability of not interrupting attack pair (i, τ). Constraint (5.22) indicates that, at most $|\mathcal{A}|$ attack pairs can be chosen to assign to attacker. Finally, constraint (5.23) is the integrality constraint for variable $z_{i\tau}$.

The attacker's subproblem is a special case of 0-1 knapsack problem with unit item weights. Note that this problem can be solved in polynomial time by sorting the attack pairs (i, τ) in a non-increasing order of $C_{i,\tau+m_i-1} \sum_{k \in \mathcal{K}'}(1 - w_{i\tau}^k)x_k$ and choosing the first $|\mathcal{A}|$ attack pairs.

5.3.3 Overall Solution Procedure

Algorithm 2 provides the pseudo-code for the overall solution procedure. The algorithm starts by randomly generating a set of initial strategies. Then, using this set of strategies, the RLPM is solved to obtain a solution $\bar{\mathbf{x}}$ and a vector of dual values $\bar{\mathbf{y}}$. Dual values $\bar{\mathbf{y}}$ are then used in the defender's subproblem to generate a new defense strategy. If a new defense strategy with a smaller expected damage is obtained, it is added to $\mathcal{K}'$. Then the adversary's subproblem is solved to generate a new attack strategy. If a new attack strategy with a greater expected damage is obtained, it is added to $\mathcal{L}'$. If, during the last two steps, either $\mathcal{K}'$ or $\mathcal{L}'$ has been updated, then the process is repeated; otherwise the procedure terminates. Because the number of possible strategies for both players is finite, the algorithm terminates after a finite number of iterations. Moreover, when the algorithm terminates, no player can improve the expected damage in their own favor by changing their strategies. Therefore, by definition, the algorithm returns a saddle-point upon termination.

5.4 Numerical Experiments

In this section, we perform computational experiments to investigate the efficiency of the proposed solution approaches and gain insight on some of its properties. The algorithms are coded in C++ and CPLEX 12.6 solver is used to solve the LPs and the defender's subproblems. A computer with 2.4 GH processor and 4 GB of RAM is used to run the numerical experiments. Our base set of test instances consists of randomly generated instances with underlying graph types including paths, cycles, grids and general planar graphs. To generate general planar graphs, the expected edge density (measured as $\frac{|\mathcal{E}|}{|\mathcal{N}|(|\mathcal{N}|-1)}$, where we do not consider self-loop edges in calculating the edge density) of 15% is used, and the number of nodes, N, ranges from 10 to 30. In

Algorithm 2: Pseudo-code for the overall solution algorithm

```
1  Initialize sets $\mathcal{K}'$ and $\mathcal{L}'$.
2  Solve RLPM. Let $\overline{\mathbf{x}} = [x_k]$, $\overline{\mathbf{y}} = [y_l]$ and $u$ be the optimal primal solution,
     dual solution and objective function value, respectively.
3  Solve the defender's subproblem using $\overline{\mathbf{y}}$ as dual values and let $\mathbf{w}^* = [w^*_{i\tau}]$
     denote the optimal solution.
4  if $v > \sum_{l \in \mathcal{L}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^l_{i\tau}(1 - w^*_{i\tau}) y_l$ then
5      Add the new defense strategy $\mathbf{w}^*$ to $\mathcal{K}'$.
6  end
7  Solve the attacker's subproblem using $\overline{\mathbf{x}}$ as primal values and let $\mathbf{z}^* = [z^*_{i\tau}]$ be
     the optimal solution.
8  if $\sum_{k \in \mathcal{K}'} \sum_{i,\tau} C_{i,\tau+m_i-1} z^*_{i\tau}(1 - w^k_{i\tau}) x_k > v$ then
9      Add the new attack strategy $\mathbf{z}^*$ to $\mathcal{L}'$.
10 end
11 if $\mathcal{K}'$ or $\mathcal{L}'$ has been updated then
12     Go to Line 2.
13 else
14     Return $v$ as the value of the game.
15     Terminate the procedure.
16 end
```

generating general graphs, we first start with a random tree and then add random edges (such that the graph remains planar) until the edge density reaches 15%.

In our first experiment, we compare the performances of different subproblem formulations for the defender. Specifically, we consider three cases: the hop-type formulation of (5.9) to (5.13) without the use of lemma 5.1 (HT), the hop-type formulation of (5.9) to (5.13) with the use of lemma 5.1 (HTL) and the flow-type formulation of (5.15) to (5.20) (FT). We consider 45 instances for each problem size, with various values of $T \in \{5, 6, \dots, 9\}$, $|\mathcal{S}| \in \{1, 2, 3\}$ and $|\mathcal{A}| \in \{1, 2, 3\}$. Tables 5.1 and 5.2 show the average computation times observed for these three approaches for paths, cycles, general planar graphs and grids. In these tables, the best results are highlighted in bold. As seen in the tables, the flow-type formulation performs significantly better than the other formulations. Moreover, for most of the instances, the use of lemma 5.1 is not helpful and it leads to higher CPU times than when the lemma is not used. In the remaining experiments in this section, we will always use the FT formulation to solve the defender's subproblem; as it is the most efficient formulation out of the three possible ones. Figure 5.1 shows the convergence trajectory of the solution algorithm for the general planar graphs under various values of N. In this figure, the vertical axis denotes the relative percent deviation (RPD) from the expected damage in equilibrium. One can see that, expected damage values stabilize way before the algorithm terminates, implying that, after the expected damage values stabilize, we can terminate the algorithm without undermining the solution quality drastically.

Next, we study the effect of the number of patrollers and attackers on the expected damage in equilibrium. An experiment is designed with general planar graphs, $|\mathcal{S}| \in \{1, 2, \dots, 10\}$ and $|\mathcal{A}| \in \{1, 2, \dots, 5\}$. Figure 5.2 exhibits the effect of number of defenders, $|\mathcal{S}|$, on the equilibrium expected damage for various number of attackers, $|\mathcal{A}|$. Figure 5.2 demonstrates that, as the number of defenders increases, the expected

Table 5.1: CPU run times (in seconds) for paths and cycles

	Path			Cycle		
N	HT	HTL	FT	HT	HTL	FT
10	32.83	29.94	**11.71**	38.61	41.44	**29.44**
15	120.03	119.11	**25.91**	301.74	332.30	**35.91**
20	252.39	408.31	**30.12**	402.73	436.94	**36.11**
25	435.88	464.59	**46.97**	412.95	427.83	**65.92**
30	496.84	529.23	**61.98**	542.12	572.56	**65.13**

Table 5.2: CPU run times (in seconds) for general graphs and grids

	General			Grid		
N	HT	HTL	FT	HT	HTL	FT
10	18.64	18.84	**8.22**	22.37	23.83	**10.30**
15	207.52	221.02	**37.49**	153.42	162.17	**34.98**
20	404.96	431.14	**100.41**	396.91	463.22	**41.79**
25	539.08	574.49	**124.97**	543.61	573.61	**68.41**
30	700.48	742.31	**341.36**	690.21	714.65	**93.46**

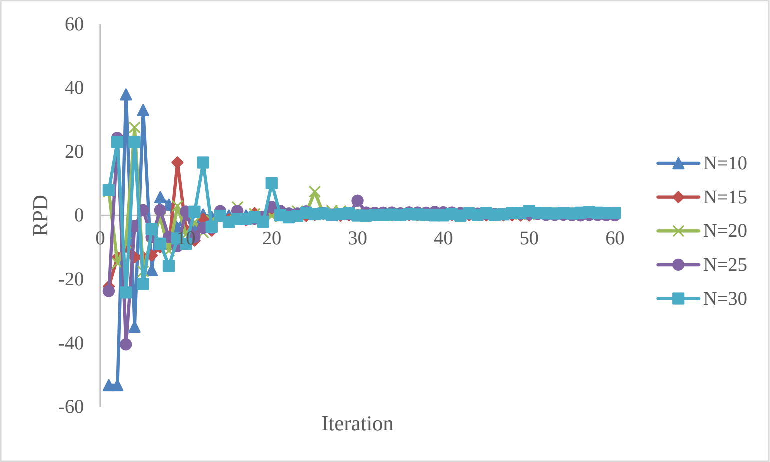


Figure 5.1: Convergence of the solution algorithm

damage decreases. Moreover, a diminishing returns effect is visible in the reduction in expected damage for each unit increment in $|\mathcal{S}|$.

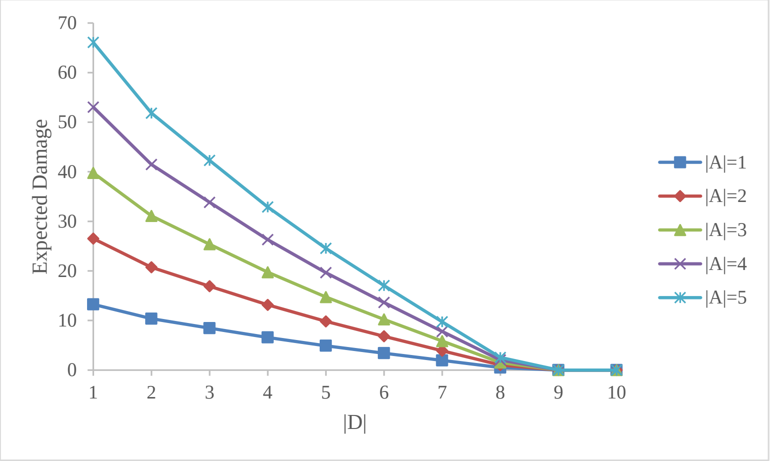


Figure 5.2: The effect of number of defenders on the expected damage in equilibrium

Next, we study the size of patrol portfolio for the case of $|\mathcal{S}| = |\mathcal{A}| = 1$. As mentioned earlier, the total number of defense strategies may be exponentially large; however, the number of defense strategies used in the saddle-point equilibrium is expected to be much smaller. In fact, when $|\mathcal{S}| = |\mathcal{A}| = 1$, the number of defense strategies used in the equilibrium with a positive probability is at most equal to the number of constraints in the LPM, which is limited from above by $N \times T + 1$. Figure

5.3 displays the size of patrol portfolio as a percentage of $N \times T + 1$ for different values of N and T. As seen in this figure, the actual size of the patrol portfolio is significantly smaller than $N \times T + 1$. It is always less than 50 percent of $N \times T + 1$ and can be as low as 10 percent. Moreover, as T increases, the percentage decreases. However, not much can be said about the effect of N on the patrol portfolio size as a percentage of $N \times T + 1$.

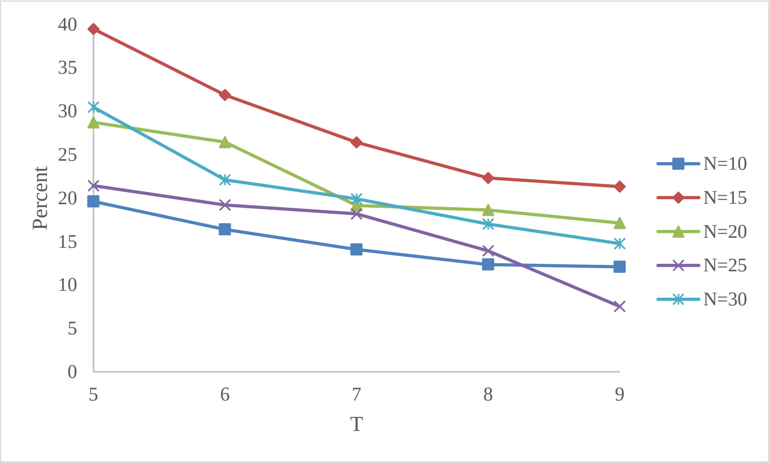


Figure 5.3: Effect of N and T on the size of patrol paortfolio

In our next experiment, we demonstrate that one of the results obtained in [10] is not valid for our more general model. Specifically, [10] prove that if all nodes have the same fixed value, then attacks on penultimate nodes are dominated. A penultimate node is defined as a non-leaf node adjacent to a leaf node. We show that, if the node values are different, then attacks on penultimate nodes may not be dominated. To this end, we use an instance of a game played on a line graph with $T = 5, N = 8, |\mathcal{S}| = |\mathcal{A}| = 1$ and $m_i = 3, \forall i \in \mathcal{N}$. Node values are assumed to be fixed over the time horizon. Figure 5.4 shows the graph of this game with corresponding node values represented above each node. As seen in this figure, node 2 is a penultimate node with value c; other nodes all have a value of 1 unit.

We solve this game for different values of c under two cases: unconstrained case

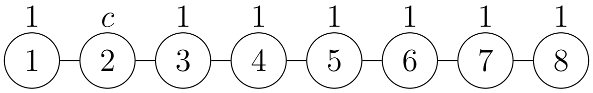


Figure 5.4: Patrolling game example on a line graph

where the attacker is free to attack any node, and constrained case where the attacker cannot attack node 2. Figure 5.5 shows the results for various values of c. As seen in this figure, the unconstrained attacker can cause more damage than the constrained attacker. Moreover, as the value of c increases, the difference between constrained and unconstrained case increases and the attacker has more incentive to attack node 2. In other words, by being able to attack node 2, the attacker can increase the damage. This means that attacking node 2 dominates not attacking node 2. Therefore, attacks on penultimate nodes may not be dominated.

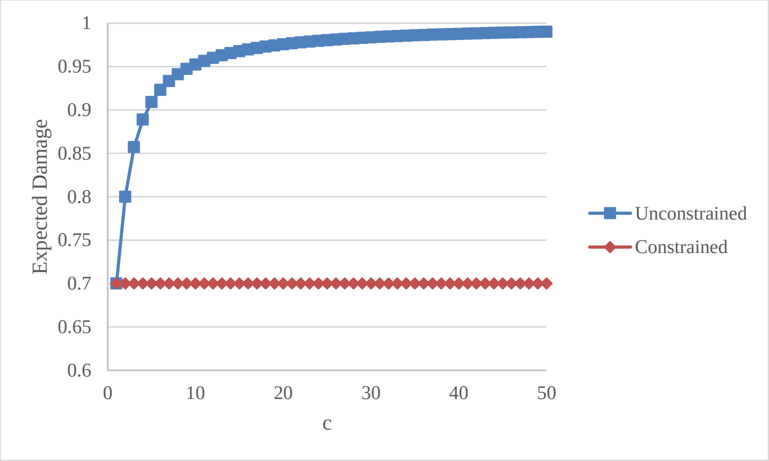


Figure 5.5: Comparison of constrained and unconstrained cases

In our final set of experiments, we study a real case of an urban rail network with 51 nodes used by [143]. In this case, the nodes in the network represent the stations and the edges represent the connections among these stations. The rail network for this case consists of two main lines that are connected with a free interchange point between them. Node values represent the time-dependent occupancy levels in each station. We consider a 12-hour work-shift, starting from 5:00 AM and ending at 5:00 PM, for this case. For more details about this case, please refer to [143]. We used our proposed solution approach to solve this problem for $|\mathcal{A}| = 3$ and $|\mathcal{S}| = 10$. For this instance of the problem, the algorithm terminates after 88 minutes. Figure 5.6 shows the obtained expected damage in the first 100 iterations of the solution algorithm for this case. As seen in this figure, after around 90 iterations, the expected damage value stabilizes and does not change drastically after this point. This observation is in line with our previous experiments.

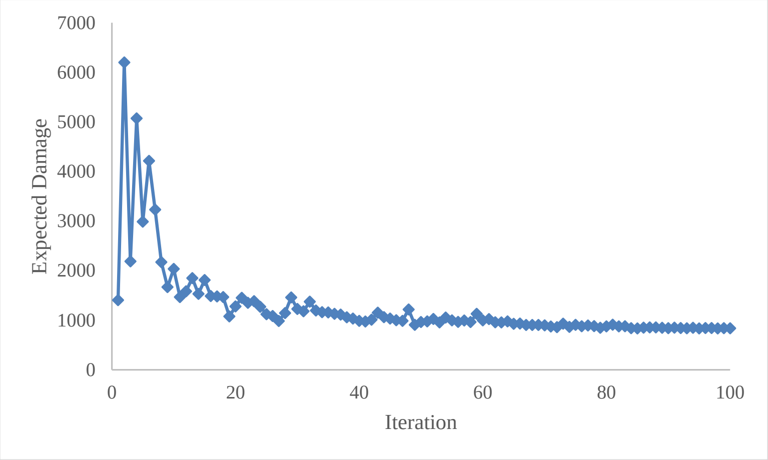


Figure 5.6: Convergence of the solution algorithm for the case study

Next, we study the distribution of patroller visits across different stations. Figure 5.7 shows the expected number of visit in a 1-month period for 5 most important stations for the case with 10 patrollers and 3 attackers. As seen in this figure, for most stations, the visits are almost equally distributed throughout the time horizon with a noticeable valley in the beginning hours and two slight peaks: one starts around 7:00 AM and ends around 9:00 AM, another one starts around 2:00 PM and ends around 4:00 PM. Next, we study the distribution of expected damage across most vulnerable stations. Figure 5.8 shows the distribution of expected damage over the time horizon for 10 most vulnerable stations, for the case with 10 patrollers and 3

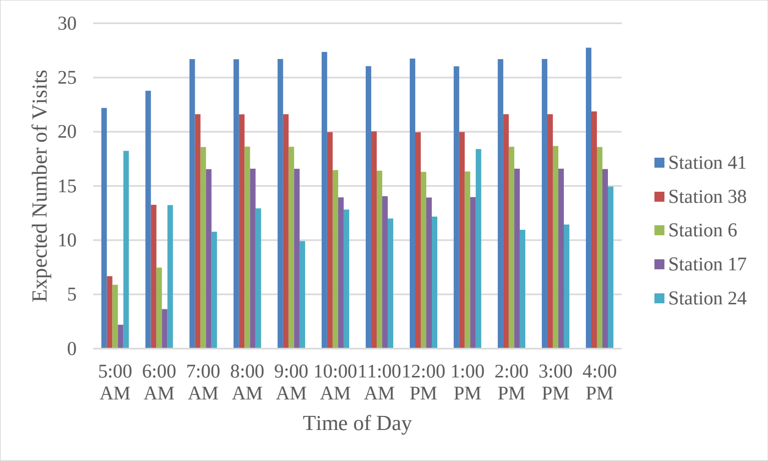


Figure 5.7: Expected number of visits for 5 most visited stations

attackers. As seen in this figure, for most stations, the expected damage concentrates on two time intervals: one starts around 7:00 AM and ends around 9:00 AM, another one starts around 1:00 PM and ends around 4:00 PM.

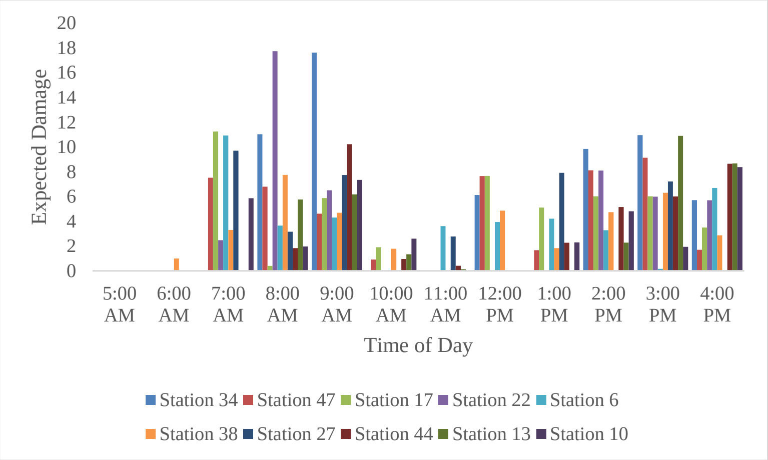


Figure 5.8: Distribution of expected damage for 10 stations with highest expected damage values

Next, we study the effect of the number of patrollers and attackers on the expected damage. Figure 5.9 shows the expected damage in equilibrium for different values of number of patrollers, $|\mathcal{S}|$, and number of attackers, $|\mathcal{A}|$. As seen in this figure, as the number of patrollers increases, the expected damage decreases. There is also a visible diminishing returns effect. Meaning that, as the number of patrollers increases, the reduction in expected damage by adding one more patroller, decreases.

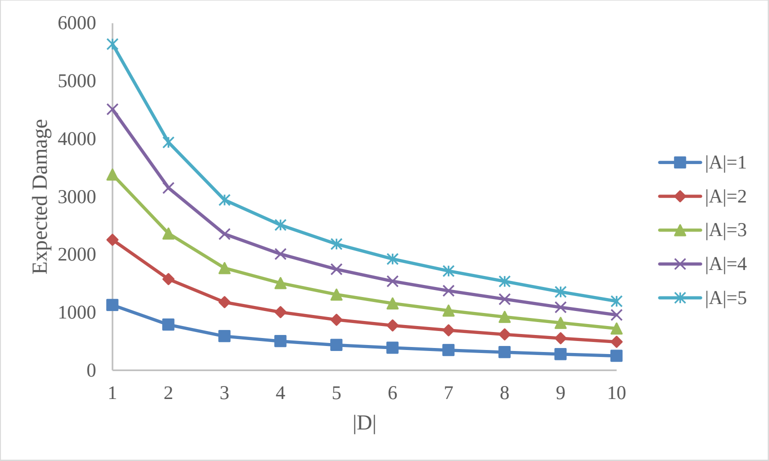


Figure 5.9: The effect of number of patrollers

Chapter 6

A Patrolling Model for Urban Rail Networks

6.1 Introduction and Literature Review

Protecting critical infrastructures against terrorism is one of the top priorities in homeland security [104]. Among these critical infrastructures, transportation systems, which serve 32 million passengers every day in the United States, are critical for supporting the national security and economic well-being. For decades, public transit systems around the world have been considered as a principal target for terrorist acts [136]. Among these systems, airliners and airports are considered to be hard targets due to the implementation of security checkpoints and increased security measures. Over the years, the number of attempted hijackings and bombings has declined gradually (although the public areas of airports still remain vulnerable). Unlike airports, where security checkpoints screen passengers and luggage, mass transit options like subways, passenger trains, and buses, are designed to be easily accessible and are therefore harder to protect. Ground transportation systems, which often include enclosed spaces packed with people, could prove attractive targets for terrorists.

Therefore, such open transit systems are considered to be soft targets for the terrorists. The attacks in Brussels and Istanbul along with many other incidents indicate that terrorists tend to target such large crowds to cause mass human casualties in addition to panic and chaos. These incidents along with many others highlight the importance of protecting such infrastructures. The threat to these infrastructures could be substantially reduced by analyzing the risk associated with attack to each infrastructure component, mitigation planning and designing efficient response policies. This includes assigning security teams and designing efficient patrol schedules to protect vulnerable areas.

Patrol scheduling involves the process of constructing optimized work timetables for security staff in order to minimize the potential damage of possible attacks. Designing patrols to protect public transport systems and other soft targets poses unique challenges that have not been properly addressed in the literature of patrol scheduling so far. One of these challenges is the dynamic nature of crowd size inside these systems. Because the adversary's primary objective is to inflict human casualties, the attacker's payoff value for each station depends on the number of people residing in the station. These numbers may change over time. Another challenge is to develop schedules that observe the constraints regarding human resources, for example the generated schedules may be required to include breaks for the security teams and these breaks should not be consecutive. Moreover, efficient methods are needed to design patrols for a general network. In this section, we address these challenges in a patrolling game setting.

The most relevant paper to our study is conducted by Lau *et al.* [83]. They study the problem of generating patrolling schedules for security teams to patrol a mass rapid transit rail network of an urban area. Their objective is to deploy patrolling units to the stations in different time units so that some scheduling and security related constraints are satisfied. They develop various mathematical models and apply it to a real rail network. The shortcoming of their model is that, because it is not a game based model, it is not designed to generate randomized schedules. To remedy this, they propose to generate randomized solutions by varying some of the problem parameters such as the start time and break time for each team. However, this may lead to sub-optimal patrol schedules. Moreover, the adversary's attack probabilities are assumed to be fixed and known. Again this is not a realistic assumption because terrorists can change the location and timing of their attacks in response to the patrol strategy. Game theoretic models are designed to generate randomized strategies and are more suited for such adversarial settings.

In this study, we develop a game theoretic model to schedule security teams in order to protect an urban rail network against terrorist attacks. We develop column generation based algorithms to efficiently solve the game under general network structures. The computational results show the efficiency of the proposed algorithms. The rest of this section is organized as follows. In section 6.2, the problem under consideration is described. The proposed column generation approach is explained in section 6.3. In section 6.4, a heuristic algorithm is presented to efficiently solve the pricing sub-problem. Section 6.5 demonstrates the computational results.

6.2 Proposed Model

The patrolling problem considered in this study involves scheduling a set of security teams $\mathcal{S}$ to protect a set of stations $\mathcal{N}$ on an urban rail network over a time horizon of T time periods. The time periods can represent the working hours in a day. Figure 6.1 shows some examples of urban rail networks. Most of these networks consists of multiple lines that are connected via interchange stations. The patrolling problem is modelled as a simultaneous game between a defender and a single adversary. The

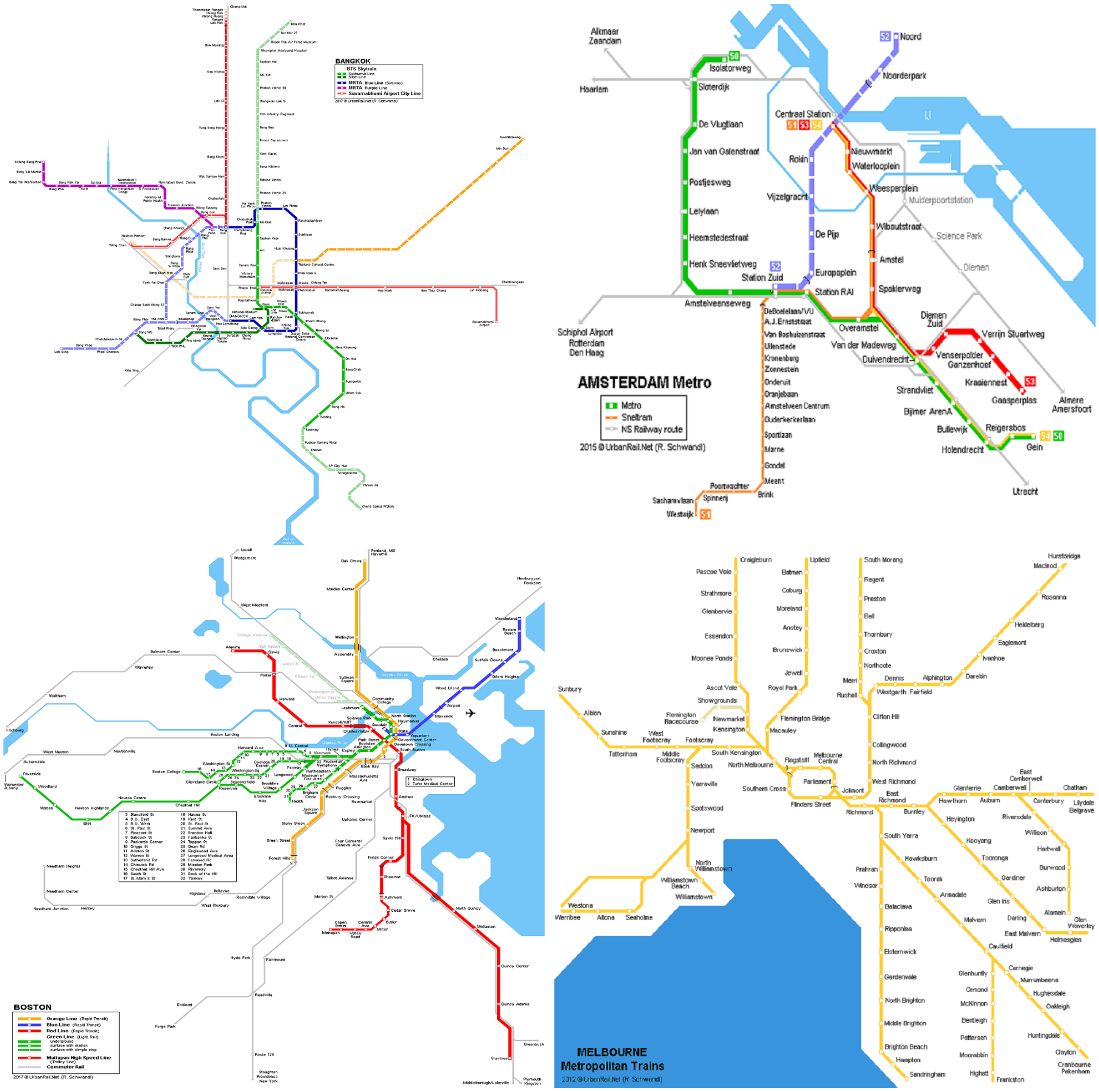

BANGKOK
BTS Skytrain
Sukhumvit Line
Silom Line
MRTA Blue Line
MRTA Purple Line
Suvarnabhumi Airport City Line
2017 © UrbanRail.Net (R. Schwandl)
AMSTERDAM Metro
Metro
Sneltram
NS Railway route
2015 © UrbanRail.Net (R. Schwandl)
BOSTON
Orange Line (Rapid Transit)
Blue Line (Rapid Transit)
Red Line (Rapid Transit)
Green Line (Light Rail)
Mattapan High Speed Line (Trolley Line)
Commuter Rail
2017 © UrbanRail.Net (R. Schwandl)
MELBOURNE
Metropolitan Trains
2012 © UrbanRail.Net (R. Schwandl)

Figure 6.1: Examples of urban rail networks in metropolitan cities such as Bangkok, Amsterdam, Boston and Melbourne

defender controls the security teams and chooses a schedule to minimize the damage from the adversary's attack, while the adversary chooses the station and time to attack. A pure strategy for the adversary is represented by a pair (j, t) which indicates the station j and time t to attack. A pure strategy for the defender is a schedule that determines the complete course of actions for all teams throughout the time horizon. These actions include patrolling different stations or taking a break. Each team should have a prespecified number of breaks and these breaks should not be consecutive or scheduled at the beginning or end of the time horizon. The payoffs to the players are determined by the expected damage to the network. While the defender wants to minimize the expected damage, the adversary wants to maximize it. We denote the value of station j at time t by C_{jt}, this value can represent the number of affected people if a successful attack is launched. If the adversary decides to attack station j at time t and the station is not being patrolled by a security team, the adversary wins a payoff of C_{jt}. On the other hand, if the station is being patrolled by a security team at the time of attack, with some probability δ_j the attack will be thwarted and with probability $1 - \delta_j$ the attack will be successful. Therefore the expected damage is $(1 - \delta_j)C_{jt}$. We represent the set of all possible schedules by $\mathcal{K}$ and index them by k with $k = 1, 2, \ldots, |\mathcal{K}|$.

The players play a zero-sum matrix game where the defender plays as the row player; with the set of all possible schedules constituting the rows of the matrix. The adversary is the column player, with the set of all possible attack pairs (j, t) constituting the columns of the game matrix. The game can be solved by generating all of the possible strategies for both players. However, for the games of large size, the set of all possible strategies is exponentially large for the defender and generating all of them becomes impractical. In the next section we develop an efficient column generation approach to obtain the Nash equilibrium for this game.

6.3 Column Generation Procedure

In this section, we develop a column generation algorithm to obtain the Nash equilibrium point for the patrolling game described in section 6. We can write the linear program (LP) to obtain the Nash equilibrium of this game as:

$$\begin{aligned} \text{Minimize} \quad & u \\ \text{subject to} \quad & u \geq \sum_{k \in \mathcal{K}} C_{jt}(1 - w_{jt}^k d_j) x_k, \quad \forall j, t, \\ & \sum_{k \in \mathcal{K}} x_k = 1, \\ & x_k \geq 0, \quad \forall k \in \mathcal{K}. \end{aligned}$$

In this formulation x_k is the probability of using schedule $k \in \mathcal{K}$ in the defender's mixed strategy. w_{jt}^k is a binary parameter that is equal to 1 if schedule k interrupts an attack strategy (j, t), zero otherwise. In the terminology of column generation, this LP is called the linear programming master problem (LPM). Note that each column in the LPM corresponds to a schedule. In general the set $\mathcal{K}$, may be exponentially large; however, the number of non-zero variables (the basic variables) in the LPM is equal to the number of constraints i.e. the total number of (j, t) pairs: $T|J|$. Therefore, even though the number of possible schedules $\mathcal{K}$ is large, only a small number of them is used in the Nash equilibrium. Column generation algorithm uses this idea to start with a subset $\mathcal{K}' \subset \mathcal{K}$ of columns and generate columns as needed. The starting subset $\mathcal{K}'$ could be any set of feasible schedules. Using the restricted set of schedules

$\mathcal{K}'$ we obtain the following LP:

$$\text{Minimize} \quad u \tag{6.1}$$

$$\text{subject to} \quad u \geq \sum_{k \in \mathcal{K}'} C_{jt}(1 - w_{jt}^k \delta_j) x_k, \quad \forall j, t, \tag{6.2}$$

$$\sum_{k \in \mathcal{K}'} x_k = 1, \tag{6.3}$$

$$x_k \geq 0, \quad \forall k \in \mathcal{K}'. \tag{6.4}$$

This problem is called Restricted LPM (RLPM). The dual of RLPM is:

$$\text{Maximize} \quad v$$

$$\text{subject to} \quad v \leq \sum_{j,t} C_{jt}(1 - w_{jt}^k d_j) q_{jt}, \quad \forall k \in \mathcal{K}',$$

$$\sum_{jt} q_{jt} = 1,$$

$$q_{jt} \geq 0, \quad \forall j, t.$$

where q_{jt} is the dual variable associated with constraint (j, t) in the RLPM. Next step is to find a column (schedule) in $\mathcal{K} \setminus \mathcal{K}'$ that could improve the current optimal solution of RLPM. Given the optimal dual solution q_{jt} of RLPM, the reduced cost of column $k \in \mathcal{K} \setminus \mathcal{K}'$ is $\sum_{j,t} C_{jt}(1 - w_{jt}^k \delta_j) q_{jt} - v$. Based on the concept of duality in linear programming, optimality of RLPM is equivalent to feasibility of the dual. Therefore, patrols that violate constraint $v \leq \sum_{j,t} C_{jt}(1 - w_{jt}^k \delta_j) q_{jt}$ can improve the current optimal solution. Therefore we should look for a column (schedule) k such that: $\sum_{j,t} C_{jt}(1 - w_{jt}^k \delta_j) q_{jt} - v < 0$. Note that q_{jt} are fixed, and the problem is to find a schedule k with w_{jt}^k such that: $\sum_{j,t} C_{jt}(1 - w_{jt}^k \delta_j) q_{jt} - v < 0$. This problem is called the pricing subproblem. The pricing subproblem involves finding a column, i.e. a schedule, with a negative reduced cost. Subsection 6.3.1 develops a mathematical program to solve the pricing subproblem. Subsection 6.3.2 presents the overall column generation algorithm and a lower bound on the value of the game.

6.3.1 Mathematical Formulation to Solve the Pricing Subproblem

In this section, we develop a mathematical formulation to solve the pricing subproblem. Here is a list of parameters and variables used to formulate the pricing subproblem:

- $a_{jj'}$: Binary parameter, is equal to 1 if it is feasible to visit stations j and j' consecutively; 0 otherwise.
- x_{sjt} : Binary variable, 1 if team s patrols station j at time t; zero otherwise.
- y_{sjt} : Binary variable, 1 if team s takes a break at station j at time t; zero otherwise.
- w_{jt} : Binary variable, 1 if a team patrols station j at time t, zero otherwise.

Using this notation, the pricing subproblem is formulated as follows:

$$\text{Maximize} \quad \sum_{j\in\mathcal{N}}\sum_{t\in T} C_{jt}w_{jt}q_{jt}\delta_j \tag{6.5}$$

$$\text{subject to} \quad \sum_{j}(x_{sjt}+y_{sjt}) = 1, \quad \forall s,t, \tag{6.6}$$

$$x_{sjt}+y_{sjt}+x_{sj',t+1}+y_{sj',t+1} \leq a_{jj'}+1, \quad \forall s,j,j',t, \tag{6.7}$$

$$\sum_{s} x_{sjt} \leq Mw_{jt}, \quad \forall j,t, \tag{6.8}$$

$$w_{jt} \leq \sum_{s} x_{sjt}, \quad \forall j,t, \tag{6.9}$$

$$\sum_{j,t} y_{sjt} = 2, \quad \forall s, \tag{6.10}$$

$$\sum_{j} y_{sjt} + \sum_{j} y_{sj,t+1} \leq 1, \quad \forall s,t, \tag{6.11}$$

$$\sum_{j} y_{sjt} = 0, \quad \forall s \in \mathcal{S}, t \in \{1,T\}, \tag{6.12}$$

$$x_{slt}, w_{jt} \in \{0,1\}. \tag{6.13}$$

In this formulation equation (6.5) is the objective function which is minimizing the reduced cost. Note that the reduced cost is equal to $\sum_{j,t} C_{jt}(1-w_{jt}\delta_j)q_{jt} - v$ which, after removing the fixed terms, is equivalent to maximizing $\sum_{j\in\mathcal{N}}\sum_{t\in T} C_{j,t}w_{jt}q_{jt}\delta_j$. Equation (6.6) ensures that each team at each time can be assigned to exactly one job. This job can be patrolling a station or taking a break. Equation (6.7) ensures for each team that the pairs of jobs undertaken consecutively are feasible, for example the team cannot consecutively patrol two stations that a far apart from each other, or they cannot take two consecutive breaks. Equation (6.8) ensures that if any team is patrolling station j at time t then w_{jt} is equal to 1. Equation (6.9) ensures that if no team is patrolling station j at time t then w_{jt} is equal to 0. Constraint (6.10) ensures that the number of breaks for each team is exactly equal to 2. Constraint (6.11) ensures that consecutive breaks do not happen. Constraint (6.12) ensures that the breaks are not scheduled at the beginning or the end of time horizon. Constraint (6.13) is the integrality constraint for variables w_{jt} and x_{ijt}.

6.3.2 Overall Column Generation Procedure and a Dual Bound

Algorithm 3 presents the pseudo-code for the overall column generation algorithm. As seen in this figure, the column generation algorithm starts with a randomly generated set of initial columns (schedules). The RLPM is solved using this set of initial columns and the vector of dual values is obtained. Dual values are then used in the pricing subproblem to generate a new column (schedule). If a new column with a negative reduced cost is obtained, it is added to the RLPM and the process is repeated; otherwise the procedure terminates. During column generation we have access to a

Algorithm 3: Pseudo-code for the overall column generation algorithm

```
1  Initialize set of schedules $\mathcal{K}'$.
2  Solve RLPM. Let $\overline{\mathbf{q}} = (q_{jt})$ and $v$ be the obtained optimal dual values and
     objective function value, respectively.
3  Solve the pricing subproblem using $\overline{\mathbf{q}}$ as dual values and let $\mathbf{w}^* = (w^*_{jt})$ be
     the obtained optimal solution.
4  if $\sum_{jt} C_{jt}(1 - w^*_{jt}\delta_j)q_{jt} - v < 0$ then
5      Add the new schedule $\mathbf{w}^*$ to $\mathcal{K}'$.
6      Go To Line 2.
7  else
8      Return $v$ as the value of the game.
9      Terminate the procedure.
10 end
```

dual bound on value of the game so that we can terminate the algorithm when a desired solution quality is reached. The following lemma offers a lower bound on the value of the game which can be computed in each iteration of the column generation algorithm.

Lemma 6.1. *(Dual bound) Let $v(RLPM)$ and $v(LPM)$ denote the optimum objective function value of the current RLPM and LPM, respectively. Also let $v(PP)$ be the minimum reduced cost obtained by solving the pricing subproblem to optimality. We have: $v(LPM) \geq v(RLPM) + v(PP)$*

Proof. General form of this result can be found in [97]. Because we know that $\sum_{k \in \mathcal{K}'} x_k = 1$ for an optimal solution of the MP, one cannot improve $v(RLPM)$ by more than 1 times the smallest reduced cost $v(PP)$, hence $v(LPM) \geq v(RLPM) + v(PP)$. □

Remark 6.1. *Note that in order for the bound in lemma 6.1 to be valid, the pricing subproblem should be solved to optimality. In general, the dual bound is not monotone over the iterations, this is called the yo-yo effect.*

6.4 A Heuristic Solution Approach for the Pricing Subproblem

In this section, we develop a dynamic programming based greedy algorithm to obtain an approximate solution to the pricing subproblem. This heuristic algorithm can be used inside the column generation procedure to obtain an approximate solution for the patrolling game. To solve the pricing subproblem we use a greedy algorithm to generate schedules. We define a patrol as a detailed course of action for one team that determines what to do at each time period t. Note that, a patrol is different from a schedule. While a schedule determines the complete course of action for all teams, a patrol does so for only one team. The definitions match if we have only one security team. For each security team the greedy algorithm assigns a patrol that maximizes $\sum_{j\in\mathcal{N}}\sum_{t\in T} C_{jt}q_{jt}\delta_j a^p_{jt}(1-w_{jt})$, where w_{jt} is a binary variable equal to 1 if station j at time t is already covered. To find such patrol we use a dynamic programming algorithm. In the following subsections, the details of the dynamic programming approach, the overall greedy algorithm, as well as some results on the quality of the proposed algorithm are presented.

6.4.1 Dynamic Programming Procedure to Find a Greedy Patrol

In this section, we develop a dynamic programming (DP) procedure to solve the pricing subproblem to find a greedy patrol. The aim is to find a patrol p with a^p_{jt} such that $\sum_{j\in\mathcal{N}}\sum_{t\in T} C_{jt}q_{jt}\delta_j a^p_{jt}(1-w_{jt})$ is maximized.

DP is a method for solving complex problems by breaking them down into smaller problems [81]. In order to solve a problem using DP, the problem must be divided into smaller problems called stages. The stages are often solved backward which is the case in the proposed DP procedure. Each stage has a number of states that are generally the information needed to solve the stage. The decision at a stage updates the state for the current stage to the state for the next stage. Given the current state, the optimal decision for the remaining stages is independent of the decisions made in the previous stages. This is the fundamental principle of optimality in DP. It means that the problem can be broken down into smaller problems which can be solved independently. Finally a recursive relationship between the values of decision at the current stage and the optimum decisions at previous stages must be identified. In other words the optimum decision uses the previously found optimum decision values. Elements of the proposed dynamic programming procedure are as follows:

- $t = 1, 2, \ldots, T$: Stage variable, each time shift is considered a stage.
- $x(t) = [b(t), l(t)]$: State variable at stage t, consists of two components: number of breaks until time shift $t : b(t)$ and location at time $t : l(t)$.
- $u(t)$: Decision at time t, take a break at an adjacent station j or patrol an adjacent station j.
- $r(x, t, u)$: Instant reward in state x at stage t if action u is taken. If the action is to patrol an adjacent station j then the obtained instant reward is $r(x, t, u) = C_{jt} q_{jt} \delta_j (1 - w_{jt})$. If the action is to take a break at an adjacent station j then the reward is $r(x, t, u) = 0$.
- $R(x, t)$: Optimum accumulated reward with state x at stage t.
- $u^*(x, t)$: Optimum action with state x at stage t.
- $F(x, u)$: Transition function, if action u is taken in state x, then the state in the next stage will be $F(x, u)$. If the action is to patrol an adjacent station j then the next state is $x(t+1) = [b(t+1), l(t+1)] = [b(t), j]$. If the action is to take a break at an adjacent station j then the next state is $x(t+1) = [b(t+1), l(t+1)] = [b(t)+1, j]$.

Now using these parameters, a recursive equation can be written for the optimum accumulated reward functions:

$$R(x, t) = \max_{u(x)} \left\{ r\left(x, t, u\right) + R\left(F(x, u), t+1\right) \right\}. \tag{6.14}$$

This equation describes an iterative relation for determining $R(x, t)$, for all feasible x and t, from the knowledge of $R(x, t+1)$ for all feasible x. $R(x, T)$ can be easily solved by using $R(x, T) = \max_{u(x)} \left\{ r\left(x, T, u\right) \right\}$ then $R(x, T-1)$ can be determined using equation (6.14). Continuing this backward procedure $R(x, 1)$ is determined.

6.4.2 Overall Heuristic Procedure

Algorithm 4 presents the pseudo code for the overall greedy approach. As seen in this pseudo code, the algorithm starts with initializing w_{jt} to indicate that no patrols have been assigned to any security team. Then the DP is used to obtain a patrol with maximum $\sum_{j \in \mathcal{N}} \sum_{t \in T} C_{jt} q_{jt} \delta_j a^p_{jt}(1 - w_{jt})$. Next we assign the obtained patrol to the security team and update w_{jt} so that is reflects the covered (j, t) pairs. This process is repeated for all available security teams. The following lemma shows that the greedy algorithm achieves an approximation ratio of $1 - (1 - \frac{1}{|\mathcal{S}|})^{|\mathcal{S}|}$.

Algorithm 4: Pseudo-code for the greedy algorithm

```
for $j \leftarrow 1$ to $|\mathcal{N}|$ do
    for $t \leftarrow 1$ to $|T|$ do
        $w_{jt} \leftarrow 0$
    end
end
for $s \leftarrow 1$ to $|\mathcal{S}|$ do
    Obtain a new patrol using DP and let $\mathbf{a^*} = (a^*_{jt})$ be the obtained optimal
      patrol.
    Assign patrol $\mathbf{a^*}$ to team $\mathcal{S}$.
    foreach $(j, t)$ with $a^*_{jt} = 1$ do
        $w_{jt} \leftarrow 1$
    end
end
```

Lemma 6.2. *For fixed values of q_{jt} Suppose w^*_{jt} and w^G_{jt} are the optimal and greedy values of w_{jt}. Then we have:* $\sum_{j \in \mathcal{N}} \sum_{t \in T} C_{jt} w^G_{jt} q_{jt} \delta_j \geq (1-(1-\frac{1}{|\mathcal{S}|})^{|\mathcal{S}|}) \sum_{j \in \mathcal{N}} \sum_{t \in T} C_{jt} w^*_{jt} q_{jt} \delta_j > (1 - \frac{1}{e}) \sum_{j \in \mathcal{N}} \sum_{t \in T} C_{jt} w^*_{jt} q_{jt} \delta_j$.

Proof. The pricing subproblem is in fact a weighted maximum coverage problem with $C_{jt}q_{jt}\delta_j$ acting as weights and $|\mathcal{S}|$ as the maximum number of sets to be selected. The result comes from the fact that for the weighted maximum coverage problem the greedy algorithm achieves an approximation ratio of $1-(1-\frac{1}{|\mathcal{S}|})^{|\mathcal{S}|}$ [106]. □

The next lemma shows the impact of using the greedy algorithm in the column generation procedure instead of solving the subproblem to optimality.

Lemma 6.3. *Let $v(RLPM)$ and $v(LPM)$ denote the optimum objective function value of the current $RLPM$ and LPM, respectively. Also let $v(PP^G)$ be the reduced cost obtained by solving the pricing subproblem using the greedy algorithm. We have: $v(LPM) \geq v(RLPM) + \frac{v(PP^G)}{1-(1-\frac{1}{|\mathcal{S}|})^{|\mathcal{S}|}} > v(RLPM) + \frac{v(PP^G)}{1-\frac{1}{e}}$.*

Proof. Let $v(PP)$ be the optimal solution of pricing subproblem. From lemma 6.2 we have $v(PP^G) \leq (1-(1-\frac{1}{|\mathcal{S}|})^{|\mathcal{S}|})v(PP) < (1-\frac{1}{e})v(PP)$. Thus:

$$v(PP) \geq \frac{v(PP^G)}{1-(1-\frac{1}{|\mathcal{S}|})^{|\mathcal{S}|}} > \frac{v(PP^G)}{1-\frac{1}{e}}. \tag{6.15}$$

Moreover from lemma 6.1 we have:

$$v(LPM) \geq v(RLPM) + v(PP). \tag{6.16}$$

The result follows from inequalities (6.15) and (6.16). □

Remark 6.2. *Note that when $|\mathcal{S}| = 1$, when there is only one security team, the bound in lemma 6.2 is tight and the greedy algorithm obtains the optimal solution.*

Remark 6.3. *Even-though we cannot prove a tighter bound for the greedy algorithm, the numerical experiments show that the solutions obtained using the greedy algorithm match the optimal solution in every instance. Therefore we conjecture that the greedy algorithm finds the optimal solutions for the pricing subproblem.*

6.5 Numerical Experiments

In this section, we perform computational experiments to investigate efficiency of the proposed algorithms and gain insight on some properties of the game. The algorithms are coded in C++ and CPLEX 12.6 solver has been used to solve the LPs and the pricing subproblems. The computational experiments are performed on a computer with 2.4 GH processor and 4 GB of RAM.

In our first experiment, we compare the performances of the proposed exact column generation approach (ECG) with the greedy column generation (GCG). Our base set of test instances consists of randomly generated instances with underlying general graphs. To generate general graphs, the expected edge density (measured as $\frac{|\mathcal{E}|}{|\mathcal{N}|(|\mathcal{N}|-1)}$, where we do not consider self-loop edges in calculating edge density) of 60% is used for the graph, and the number of stations, $|\mathcal{N}|$, ranges from 20 to 40. In generating general graphs, we first started with a random tree and added random edges until the edge density reaches 60%. We generated five instances for each problem size, with different values of $T \in \{10, 11, \dots, 15\}$ and $|\mathcal{S}| \in \{1, 2, \dots, 5\}$.

Our results show that the GCG always finds the same expected damage value as the ECG. This leads us to conjecture that the GCG always finds the optimal solution. We then compare the algorithms in terms of their run times. Tables 6.1 and 6.2 show the obtained run times in seconds. In these tables for each instance the smaller run time is highlighted in bold. As seen in these tables GCG performs better than ECG for all instances of the problem. Moreover, for both algorithms, the run time generally increases as the number of stations, i.e. $|\mathcal{N}|$, and the number of time periods, i.e. $|T|$, increases. Figure 6.2 shows the effect of increasing the number of security teams on the expected damage for different number of stations. As seen in this figure, the expected damage decreases as the number of security teams increase. However, the amount of decrease in expected damage also decreases as the number of security teams increases. This diminishing returns effect is visible for all values of $|\mathcal{N}|$.

Table 6.1: Comparison of exact column generation and greedy column generation run times (seconds)

	T=10		T=11		T=12	
$\mathcal{N}$	ECG	GCG	ECG	GCG	ECG	GCG
20	195.01	**64.18**	228.04	**62.37**	323.93	**119.62**
25	354.48	**115.77**	471.22	**153.19**	519.32	**208.09**
30	477.88	**146.73**	554.75	**229.61**	776.09	**236.04**
35	891.87	**276.07**	1015.90	**422.39**	1287.66	**503.86**
40	1147.89	**336.41**	1855.11	**719.51**	2830.80	**663.60**

Table 6.2: Comparison of exact column generation and greedy column generation run times (seconds)

	T=13		T=14		T=15	
$\mathcal{N}$	ECG	GCG	ECG	GCG	ECG	GCG
20	382.34	**146.01**	385.26	**199.01**	472.00	**201.62**
25	707.37	**383.70**	725.17	**272.52**	987.22	**430.63**
30	1272.09	**619.50**	1595.14	**829.14**	1945.62	**1187.35**
35	1384.21	**578.76**	2044.15	**932.16**	2554.24	**1352.95**
40	2746.47	**1297.65**	2897.79	**1432.55**	3736.70	**2273.71**

Figure 6.3 shows convergence of the lower and upper bounds of GCG over iterations for an instance of the problem with $|\mathcal{S}| = 1$, $T = 10$ and different values of $|\mathcal{N}|$. In each iteration the lower bound is computed using lemma 6.3 and the upper bound

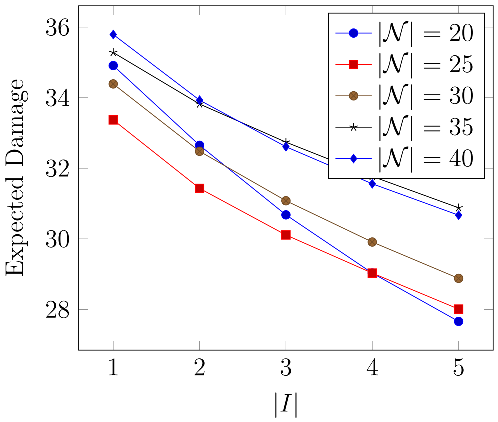


Figure 6.2: The effect of number of security teams on Expected Damage

is taken as the current objective function value. As seen in this figure, the lower bound is not monotone over the iterations and the yo-yo effect is visible. Moreover, for most cases, the upper bound value stabilizes way before the algorithm terminates. This means that after the upper bound values stabilize, we can terminate the column generation algorithm without undermining the solution quality drastically.

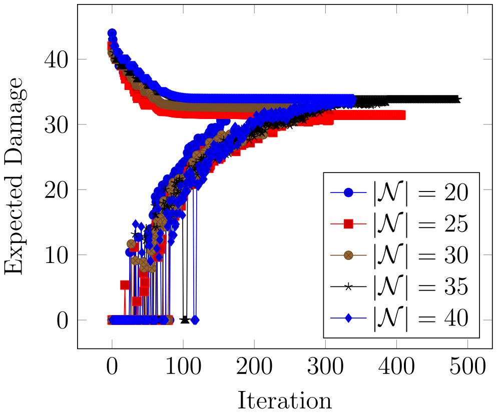


Figure 6.3: Convergence of GCG over iterations

Next, we analyze the effect of detection probabilities on the expected damage, here we assume that the detection probabilities equal to each other. Figure 6.4 shows the effect of detection probability on the expected damage for different values of number of security teams $|\mathcal{S}|$. As seen in this figure, the expected damage is smaller when there are more security teams. Moreover, as the detection probability increases, the expected damage, generally, decreases and this decrease, roughly speaking, behaves linearly for higher values of detection probability. Next, we consider a real case of an urban rail network with 51 stations. Figure 6.5 shows the network graph of this case. As seen in this figure, there are two main lines that connect different parts of the city together. There is one free interchange between stations 41 and 16. Occupancy levels in each station are collected based on ridership totals which are in turn based on turnstile entry and exclude free interchange ridership. A 15 hour planning time horizon, starting from 5:00AM and ending at 8:00pm, is considered for this problem (i.e. $T = 15$ is considered). Based on this information, we run the GCG algorithm to obtain the best patrolling strategy for $|\mathcal{S}| \in \{1, 2, \dots, 10\}$. We first study the effect of the defender's deviation from the Nash equilibrium on the expected damage. Specifically we consider the case that the defender, in deriving her strategy, mistakenly, thinks that the attack probabilities are the same and are uniformly distributed over stations. She then uses a probabilistic approach (PA) to obtain a single schedule. We compare the expected damage in this case with the expected damage in the Nash equilibrium (NE). Figure 6.6 shows the results of this comparison. As seen in this figure, Nash strategy results in smaller expected damage values for all instances. Moreover, as the number of teams increases, the expected

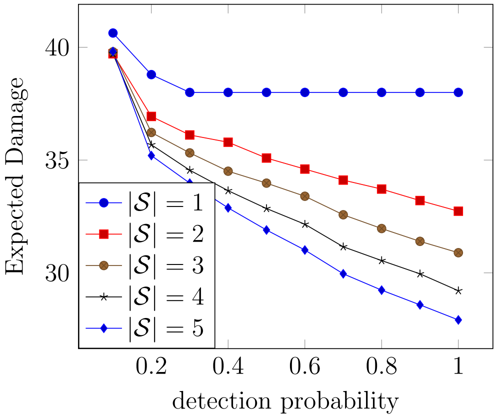


Figure 6.4: The effect of detection probability on Expected Damage

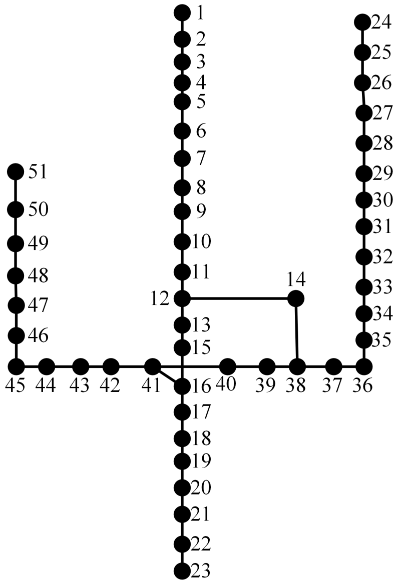


Figure 6.5: Case network

damage decreases for both NE and PA. The diminishing returns phenomenon is visible for NE, however, this effect does not exists for PA. We now study the distribution of the expected number of visits in important stations. Figure 6.7 shows the expected number of visits over a 30 day period for 5 most visited stations based on time of the day. As seen in this figure, majority of the stations have two peak visit times: one starts around 7:00 AM and end around 10:00 AM, the other one starts around 2:00 PM and end around 5:00 PM. Some stations also have peak visit times at the

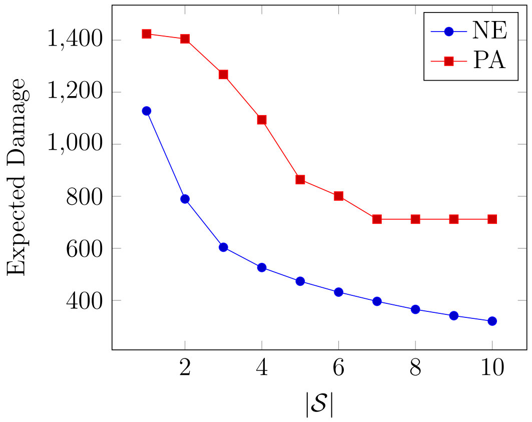


Figure 6.6: The effect of number of security teams on Expected Damage

start and end of times horizon. Next, we study the distribution of the expected

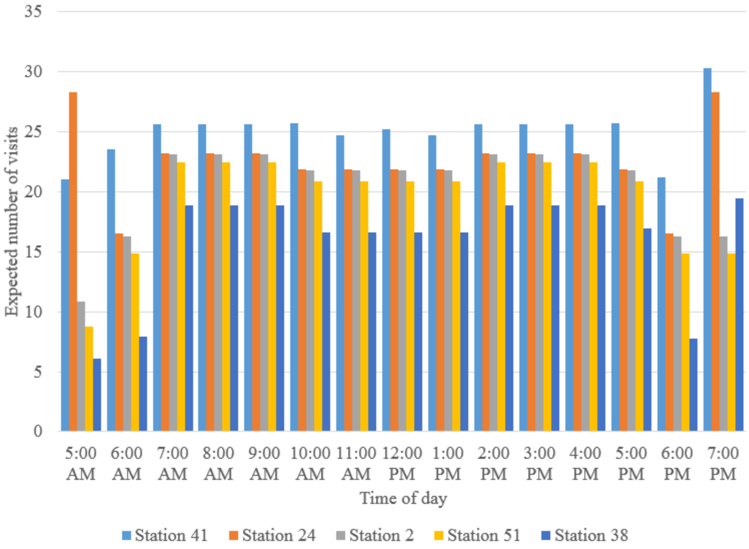


Figure 6.7: Expected number of visits for five most visited stations

damage among stations based on time of the day. Figure 6.8 shows the distribution of expected damage for 10 stations with highest expected damage values. As seen in this figure, for majority of the stations there are two peak times for expected damage: one starts around 7:00 AM and end around 10:00 AM, the other one starts around 1:00 PM and end around 5:00 PM.

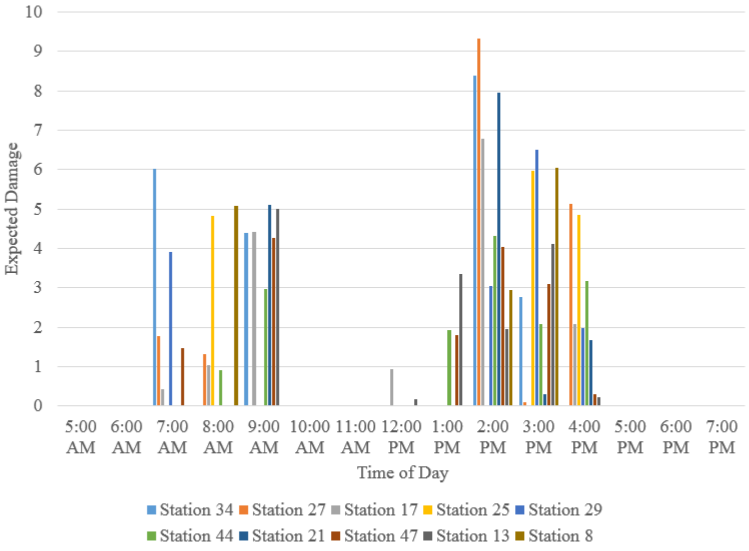


Figure 6.8: Distribution of expected damage for 10 stations with highest expected damage values

Chapter 7

Weighted Search Games with Multiple Hidden Objects, Multiple Search Teams and Dispersed Hiding Locations

7.1 Introduction and Literature Review

The search games were first introduced by Rufus Isaacs in 1965 [66]. His "simple search game" is defined on an arbitrary region $\mathbf{R}$. The Hider picks a point in $\mathbf{R}$ and the Searcher selects a unit speed trajectory in $\mathbf{R}$ to find the Hider. Payoff to the players is the search time, which is the time required for the Searcher's trajectory to meet the Hider for the first time. Gal [30] provides a more precise formulation of the search game defined on a network Q consisting of a finite set of connected arcs and a predetermined starting point O. The Hider picks a point in Q to hide and the Searcher selects a unit speed path starting from O. Since then, many variations of the network search games have been introduced [2, 3, 4, 6, 7, 8, 9, 15, 26, 31, 34, 147]. For example, Dagan and Gal [26] consider an arbitrary starting point for the Searcher, Alpern [3] studies a find-and-fetch search model. Other examples include search problems on networks with asymmetric travel times [2, 8], an expanding search paradigm [7], search games on a lattice [147], search games with searching costs at nodes [15], bi-modal search games to find a small object [9], and search games with combinatorial search paths [4]. For a more recent survey of search games see [32, 65]. For a background in search games see [6, 8, 31, 34].

In this chapter, we study a game played between a Hider and a Searcher. The Hider picks one or more locations on a network to hide some objects and the Searcher follows a path to find the hidden objects such that an objective function is optimized. The objective function is usually assumed to be the search time. While the Hider aims at maximizing the search time, the Searcher wants to minimize it. Therefore, this problem can be formulated as a zero-sum game. Majority of the papers in the literature of search games assume that the players do not have preference over different locations on the network and they only care about the search time. However, there are some cases in which the players differentiate the hiding places from each other. In these cases, the players may want to minimize/maximize a weighted search time with node weights representing the rate of damage. For example, in certain attacks (biological or chemical), casualty rate depends on factors such as population density, environment conditions etc. Therefore, different locations may have different casualty rates and the overall damage will be proportional to time and casualty rate. Another example is the problem of detecting an eavesdropping agent over communication channels [37]. Different channels may have different transmission capacities and the rate of damage to the network will be proportional to the detection time and the capacity of the channel. The only study that considers node weights in the search games is conducted by Zoroa *et al.* [148]. However, they only consider such games on lattices, not general graphs.

The problem of searching for a hidden object in discrete time and discrete space has been well studied in the literature. Blackwell studied the problem of finding a hidden object in a set of boxes [99]. Given a probability distribution over which box contains the hidden object and a search cost for each box, the objective is to minimize the expected cost of finding the object. Game-theoretic variants of this game have also been studied [20, 41, 95, 124, 125]. Less attention has been given to the search games with multiple hidden objects. Assaf and Zamir [11] and Sharlin [133] study a search game with several hidden objects with the objective of finding one of these objects with minimum expected cost. Lidbetter [91] investigates a game in which all of the hidden objects have to be found at minimum expected cost. Alpern *et al.* [5] introduce caching games in which a Searcher with a limited resource aims at maximizing the probability of finding a certain number of hidden objects. Lidbetter and Lin [92] introduce multi-look search problems in which the Searcher can find at most one hidden object each time a box is opened. In other words, to find all of the hidden objects in a box, the Searcher may have to open it multiple times.

We propose a new discrete search game in which different hiding locations have different weights and the payoff to the players is proportional to the search time and the location weight. In this setting, the location weights represent the rate of damage at that location. The players want to minimize/maximize the overall damage to the network, which is represented as a weighted search time. The game is played between a Searcher who controls a set of homogeneous search teams and a Hider who picks hiding locations to hide the objects. The hiding locations are dispersed on a network and the time it takes to visit a location depends on the previously visited location. We first consider a special case of this game and characterize the Nash equilibrium. Afterwards, we develop a column and row generation procedure to solve the general form of the game. The rest of this chapter is organized as follows. In section 7.2, the proposed weighted discrete search game is presented and some properties of this game are proven. In section 7.3, a solution approach based on column and row generation is developed to solve this game in its general form. Section 7.4 presents numerical experiments to investigate the efficiency of the proposed algorithms and gain insight on some properties of the game.

7.2 Proposed Model

A Searcher and a Hider play a zero-sum weighted search game. The Searcher controls a set of S search teams and the Hider controls a set of H objects to hide. The game is played on a complete graph $Q = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{0, 1, 2, \ldots, N\}$ is the set of nodes in the graph and $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$ is the set of edges. Using Gal's [30] approach, let node 0 represent the origin, which is a predetermined location where all search teams are initially located. Let $\mathcal{N}_h = \{1, 2, \ldots, N\}$ denote the set of N potential hiding locations. These locations are dispersed throughout a large area, and each location may differ in rate of damage, and difficulty to search. The Hider hides the objects in these potential locations. The Searcher uses a set of homogeneous search teams to find the hidden objects. It takes a search team v_i time units to inspect location $i \in \mathcal{N}_h$ and, for each edge $(i, j) \in \mathcal{E}$, the time required to travel from location i to location j is denoted by d_{ij}. If a location is inspected, the search team will find the hidden object, if any, i.e., false negative response is not possible. Each location i has a weight denoted by C_i. This weight represents the rate of damage to the network, if the Hider decides to hide an object at location i. Throughout the chapter, we assume, without loss of generality, that the locations are sorted in the order of decreasing weights, i.e., $C_1 > C_2 > \cdots > C_N$.

The objective of the Hider is to maximize the total damage to the network, while the Searcher wants to minimize the damage. A pure strategy for the Hider (also called a pure hiding strategy throughout the chapter) is to select a hiding location to hide each object. We assume that, hiding more than one object in a location does not increase the rate of damage in that location. Hence, it is not beneficial for the Hider to hide multiple objects in a single location. A pure strategy for the Searcher (also called a pure search strategy throughout the chapter) is to select a joint schedule for the search teams.

The players play a zero-sum matrix game with the Searcher playing as the row player, and the set of all possible pure search strategies constituting the rows of the matrix. The Hider plays as the column player, with the set of all possible pure hiding strategies constituting the columns of the game matrix. We use $\mathcal{K}$ to denote the set of all possible pure search strategies and k to index them. Let x_k be the probability of using search strategy k in the Searcher's mixed strategy. Hence $\mathbf{x} = (x_1, x_2, \dots, x_{|\mathcal{K}|})$ represents a mixed strategy of the Searcher, where $|\mathcal{K}|$ denotes the cardinality of set $\mathcal{K}$, $x_k \geq 0$, for all $k \in \mathcal{K}$ and $\sum_{k=1}^{|\mathcal{K}|} x_k = 1$. Similarly, we use $\mathcal{L}$ to denote the set of all possible pure hiding strategies and index them by l. Let y_l denote the probability of using hiding strategy l. Hence, a mixed strategy of the Hider is denoted as $\mathbf{y} = (y_1, y_2, \dots, y_{|\mathcal{L}|})$, $y_l \geq 0$, $\forall l \in \mathcal{L}$, and $\sum_{l=1}^{|\mathcal{L}|} y_l = 1$. We use parameter t_i^k to denote the time at which joint schedule k completes inspection in location i, and r_i^k to denote the order of visiting location i while using schedule k. We also define binary parameter z_i^l, which is equal to 1 if hiding strategy l involves hiding an object in location i, 0 otherwise. If the Searcher and the Hider use mixed strategies $\mathbf{x}$ and $\mathbf{y}$, respectively, then the expected total damage is $v(\mathbf{x}, \mathbf{y}) = \sum_{k \in \mathcal{K}} \sum_{l \in \mathcal{L}} w_i t_i^k z_i^l x_k y_l$. The Nash equilibrium (saddle point) of the game is $(\mathbf{x}^*, \mathbf{y}^*)$ at which the following inequalities hold:

$$v(\mathbf{x}^*, \mathbf{y}) \leq v(\mathbf{x}^*, \mathbf{y}^*) \leq v(\mathbf{x}, \mathbf{y}^*).$$

The following theorem describes the saddle-point equilibrium for a special case of the game.

Theorem 7.1. *If $d_{ij} = d, v_i = v, \forall (i,j) \in \mathcal{E}, i \in \mathcal{N}_h$ and $H = S = 1$, then the equilibrium is characterized by*

$$q_i = \begin{cases} \frac{\frac{1}{C_i}}{\sum_{j=1}^{m} \frac{1}{C_j}}, & i = 1, 2, \ldots, m, \\ 0, & i = m+1, \ldots, N, \end{cases} \tag{7.1}$$

$$E\left[\textit{search order of the ith node}\right] = \sum_{k \in \mathcal{K}} r_i^k x_k = \frac{m(m+1)}{2} \frac{\frac{1}{C_i}}{\sum_{j=1}^{m} \frac{1}{C_j}}, \quad i = 1, 2, \ldots, N, \tag{7.2}$$

where q_i is the probability of hiding the object at node i for the Hider, and m is the index s that leads to the maximum value of: $\frac{\frac{s(s+1)}{2}}{\sum_{j=1}^{s} \frac{1}{C_j}}$.
Value of the game is: $V^* = (d+v) \frac{\frac{m(m+1)}{2}}{\sum_{j=1}^{m} \frac{1}{C_j}}$.

Note that in this case, one can write t_i^k simply as $t_i^k = r_i^k (d+v)$.

Proof. Based on the definition of Nash Equilibrium, $(\mathbf{x}, \mathbf{q})$ is an equilibrium if and only if for some u:

$$(d+v) C_i \sum_{k \in \mathcal{K}} x_k r_i^k \begin{cases} = u, & q_i > 0, \\ \leq u, & q_i = 0, \end{cases} \quad \Rightarrow \quad C_i \sum_{k \in \mathcal{K}} x_k r_i^k \begin{cases} = \frac{u}{d+v} \equiv \bar{u}, & q_i > 0, \\ \leq \bar{u}, & q_i = 0. \end{cases} \tag{7.3}$$

Similarly,

$$\sum_{i=1}^{N} C_i q_i r_i^k \begin{cases} = \bar{u}, & x_k > 0, \\ \geq \bar{u}, & x_k = 0. \end{cases} \tag{7.4}$$

We define A as the set of active nodes in which the Hider hides the object with a positive probability, i.e., $A = \{i | q_i > 0\}$. Clearly, if $i, j \in A$ with $C_i > C_j$, the expected search orders will satisfy

$$E\left[\text{search order of the } i\text{th node}\right] = \sum_{k \in \mathcal{K}} x_k r_i^k = \frac{\bar{u}}{C_i} < \frac{\bar{u}}{C_j} = E\left[\text{search order of the } j\text{th node}\right]. \tag{7.5}$$

On the other hand, given the set of active nodes A, if $C_i q_i = c$, a constant for all $i \in A$, i.e., $q_i = c/C_i = \frac{\frac{1}{C_i}}{\sum_{j\in A}\frac{1}{C_j}}$ since $\sum_{i\in A} q_i = 1$, then by [80]

$$\sum_{i\in A} C_i q_i r_i^k = \frac{1}{\sum_{j\in A}\frac{1}{C_j}} \sum_{i\in A} r_i^k \geq \frac{|A|(|A|+1)}{2} \frac{1}{\sum_{j\in A}\frac{1}{C_j}}. \tag{7.6}$$

Moreover, the minimum of the left hand side of (7.6) is achieved only in the case that the order of visiting all nodes in A, follows the natural numbers up to $|A|$, that is $\sum_{i\in A} r_i^k = \sum_{r=1}^{|A|} r = \frac{|A|(|A|+1)}{2}$. Thus, the searcher should visit during the first $|A|$ instances, each one of the locations in A, regardless of the exact order, that is $r_i^k \in \{1, \ldots, |A|\}$ for all $i \in A$, and $k \in B = \{l : x_l > 0\}$.

Following the above discussion, given the set of active nodes A, we can show that the following is a solution to equations (7.3) and (7.4):

$$q_i = \begin{cases} \frac{\frac{1}{C_i}}{\sum_{j\in A}\frac{1}{C_j}}, & i \in A, \\ 0, & i \notin A. \end{cases} \tag{7.7}$$

$$E\left[\text{search order of the } i\text{th node}\right] = \sum_{k\in\mathcal{K}} x_k r_i^k \begin{cases} = \frac{|A|(|A|+1)}{2} \frac{\frac{1}{C_i}}{\sum_{j\in A}\frac{1}{C_j}}, & i \in A, \\ \leq \frac{|A|(|A|+1)}{2} \frac{\frac{1}{C_i}}{\sum_{j\in A}\frac{1}{C_j}}, & i \notin A. \end{cases} \tag{7.8}$$

Moreover, we can show that, based on this solution, the expected total damage is $ETD(A) = (d+v)\frac{\frac{|A|(|A|+1)}{2}}{\sum_{j\in A}\frac{1}{C_j}}$. Note that, using schedules that visit nodes in set A before other nodes, the quantity $\bar{u}$ can be obtained from (7.4), which can be used to derive (7.8) from (7.3).

Next step is to characterize the active set A. We show that, for any active set A with $i \in A, j \notin A, C_j > C_i$, the Hider can replace node i with node j to create an active set A' that leads to a higher ETD. This is because $ETD(A) = (d+v)\frac{\frac{|A|(|A|+1)}{2}}{\sum_{j\in A}\frac{1}{C_j}}$ is an increasing function of $C_i, i \in A$. Therefore, replacing C_i with C_j will lead to a higher ETD. Thus, knowing that C_i values are sorted, the active set with the highest

ETD is of the form $A = \{1, 2, \ldots, l\}$ and the cut-off index $l = m$ is, by assumption, the cut-off index that leads to the highest ETD. Therefore, the Hider cannot increase the ETD by changing the active set from $A = \{1, 2, \ldots, m\}$ to any other set.

Next, we show that there exists a search strategy characterized by equation (7.2). The space of possible vectors for the expected visiting orders is the convex hull of all permutations of the set $\{1, 2, \ldots, N\}$. This space is called permutahedron and is not full-dimensional [80].

$$Conv(R) = \left\{ \bar{r} \in \mathbb{R}^N, \sum_{i=1}^{N} \bar{r}_i = \frac{N(N+1)}{2}, \sum_{i \in Q} \bar{r}_i \geq \binom{|Q|+1}{2} \quad Q \subseteq \{1, 2, \ldots, N\} \right\}, \tag{7.9}$$

where $\bar{r}_i$ is the expected visiting order of node i, i.e., $\bar{r}_i = \sum_{k \in K} r_i^k x_k$. To show that there exists a strategy for the Searcher, we need to prove that the expected visiting orders, from equation (7.2), are in the permutahedron. Obviously, $\sum_{i=1}^{N} \bar{r}_i = \frac{N(N+1)}{2}$ is true. We proceed to show the other inequalities are also valid. We show that the inequality is valid for $|Q| = l$ with $1 \leq l < N$. For each l, it is enough to show the inequality for $Q = \{1, 2, \ldots, l\}$ (for other sets of size l, the inequality will follow due to ordered node weights). We need to show that:

$$\sum_{i=1}^{l} \frac{m(m+1)}{2} \frac{\frac{1}{C_i}}{\sum_{j=1}^{m} \frac{1}{C_l}} \geq \binom{l+1}{2} = \frac{(l+1)l}{2}.$$

In other words, we need to show that:

$$\frac{\frac{m(m+1)}{2}}{\sum_{j=1}^{m} \frac{1}{C_l}} \geq \frac{\frac{l(l+1)}{2}}{\sum_{i=1}^{l} \frac{1}{C_l}}.$$

But, this is true based on the assumption that index m is chosen so that this inequality holds. Therefore, there exists a search strategy that satisfies the equation (7.2). This completes the proof. □

Remark 7.1. *Using the expected search order values obtained in equation* (7.2)*, we can compute a complete search strategy for the Searcher in polynomial running time of* $O(N^2)$*. This can be done using the decomposition algorithm provided in [141].*

Remark 7.2. *Theorem 7.1 indicates that, when $d_{ij} = d, v_i = v$, $\forall (i,j) \in \mathcal{E}, i \in \mathcal{N}_h$ and $H = S = 1$, the Nash equilibrium is of threshold type. Meaning that, there exists a cut-off index, m, such that the Hider will only consider the first m locations and ignore the remaining ones. This theorem can be used to obtain an upper bound on the value of the game in general by taking $d = \max_{i,j} d_{ij}$ and $v = \max_i v_i$. Similarly, a lower bound can be computed by taking $d = \min_{i,j} d_{ij}$ and $v = \min_i v_i$.*

Remark 7.3. *The Nash equilibrium characterized in Theorem 7.1 prescribes lower hiding probabilities for locations with higher weights. The reason for this counter-intuitive outcome is that, the expected visiting order of the locations with higher weights is smaller in equilibrium. Therefore, even though the rate of damage is higher for locations with higher weights, the expected duration of damage is smaller for these locations. This observation is in line with the results obtained in the area of security games [16, 142].*

Lemma 7.1. *Given an upper bound $\overline{V}$ on the value of the game, any search strategy $k \in K$ that completes inspection at any node i at time t_i^k with $t_i^k > \frac{\overline{V}}{C_i}$ does not belong to a Nash equilibrium.*

Proof. We show that the Hider can improve his payoff by hiding an object in location i. The expected damage from hiding an object in location i is: $C_i t_i^k > \overline{V} \geq V^*$. Therefore, using search strategy k leads to an expected damage value that is higher than the expected damage in equilibrium. Therefore, strategy k does not belong to a Nash equilibrium. □

The game can be solved by generating all possible pure strategies for both players. However, this may not be efficient for games of larger size. In the next section, we develop an efficient algorithm to obtain a Nash equilibrium for this game.

7.3 Solution Approach for General Weighted Search Games

In this section, we develop a solution algorithm based on column and row generation [105, 122] to find a Nash equilibrium for the weighted search game introduced in the previous section. The proposed solution method can also be described as a modification of the algorithm proposed in [42, 43]. Because this is a zero-sum game, we can write the following linear program (LP) to obtain a Nash equilibrium for this game:

$$\begin{aligned} \text{LPM} \qquad\qquad & \text{Minimize} \quad u \\ & \text{subject to} \quad u \geq \sum_{k\in\mathcal{K}} x_k \sum_{i\in\mathcal{N}_h} C_i t_i^k z_i^l, \quad \forall l \in \mathcal{L}, \\ & \sum_{k\in\mathcal{K}} x_k = 1, \\ & x_k \geq 0, \quad \forall k \in \mathcal{K}. \end{aligned}$$

This LP is called the linear programming master problem (LPM). In this formulation, $\mathcal{K}$ is the set of all possible joint schedules, indexed by k. x_k is a decision variable representing the probability of using joint schedule $k \in \mathcal{K}$ in the Searcher's mixed strategy. In general, the sets $\mathcal{K}$ and $\mathcal{L}$ may be exponentially large; however, the number of strategies used is expected to be much smaller. Our proposed column and row generation algorithm uses this idea to start with small subsets $\mathcal{K}' \subset \mathcal{K}$ and $\mathcal{L}' \subset \mathcal{L}$ of search and hiding strategies and generates them as needed. The starting subsets $\mathcal{K}'$ and $\mathcal{L}'$ could be any set of feasible strategies. Using the restricted set of strategies $\mathcal{K}'$, we obtain the following LP:

LPM-RS

$$\text{Minimize} \quad u \tag{7.10}$$

$$\text{subject to} \quad u \geq \sum_{k \in \mathcal{K}'} x_k \sum_{i \in \mathcal{N}_h} C_i t_i^k z_i^l, \quad \forall l \in \mathcal{L}', \tag{7.11}$$

$$\sum_{k \in \mathcal{K}'} x_k = 1, \tag{7.12}$$

$$x_k \geq 0, \quad \forall k \in \mathcal{K}'. \tag{7.13}$$

This problem is called LPM with Restricted Strategies (LPM-RS). The dual of LPM-RS is

Dual LPM-RS

$$\text{Maximize} \quad v \tag{7.14}$$

$$\text{subject to} \quad v \leq \sum_{l \in \Lambda'} \sum_{i \in \mathcal{N}_h} C_i t_i^k z_i^l y_l, \quad \forall k \in \mathcal{K}', \tag{7.15}$$

$$\sum_{l \in \mathcal{L}'} y_l = 1, \tag{7.16}$$

$$y_l \geq 0, \quad \forall l \in \mathcal{L}'. \tag{7.17}$$

In this formulation, y_l is the dual variable corresponding to constraint (7.11) in LPM-RS. This variable represents the probability of using hiding strategy l in the Hider's mixed strategy. Moreover, v is the dual variable corresponding to constraint (7.12) which represents the minimum expected total damage. Next step is to find new strategies in $\mathcal{K} \setminus \mathcal{K}'$ and $\mathcal{L} \setminus \mathcal{L}'$ that could improve the current optimal solution for the corresponding players. Given the optimal dual solution y_l of LPM-RS, the reduced cost of joint schedule $k \in \mathcal{K} \setminus \mathcal{K}'$ is given by $\sum_{l \in \mathcal{L}'} \sum_{i \in \mathcal{N}_h} C_i t_i^k z_i^l y_l - v$. Based on the concept of duality in linear programming, optimality of LPM-RS is equivalent to the feasibility of its dual. Therefore, joint schedules that violate the constraint $v \leq \sum_{l \in \mathcal{L}'} \sum_{i \in \mathcal{N}_h} C_i t_i^k z_i^l y_l$, can improve the current optimal solution. Consequently, we need to look for a joint schedule k such that: $\sum_{l \in \mathcal{L}'} \sum_{i \in \mathcal{N}_h} C_i t_i^k z_i^l y_l - v < 0$. Note that y_l values are fixed, and the problem is to find a joint schedule k with t_i^k such that:

$\sum_{l\in\mathcal{L}'}\sum_{i\in\mathcal{N}_h} C_i t_i^k z_i^l y_l - v < 0$. Therefore, we are looking for a new joint schedule k that leads to a smaller expected total damage, $\sum_{l\in\mathcal{L}'}\sum_{i\in\mathcal{N}_h} C_i t_i^k z_i^l y_l$, than the current expected total damage, v. To obtain an improving strategy for the Hider, we consider the LPM-RS. Given the optimal solution x_k of LPM-RS, the current expected total damage is u. Therefore, the Hider should look for a new hiding strategy l with z_i^l such that the expected total damage, i.e., $\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}_h} C_i t_i^k z_i^l x_k$, is greater than the current expected total damage, i.e., u.

In the following subsections, we develop mathematical programs and solution algorithms for the Searcher's and Hider's subproblems. We also present the overall column and row generation algorithm and provide bounds on the value of the game.

7.3.1 The Searcher's Subproblem

In this section, a flow type formulation is developed for the Searcher's subproblem. Here is a list of parameters and variables used to formulate the Searcher's subproblem:

- x_{ij}: Binary variable, for $(i,j)\in\mathcal{E}$, it is equal to 1 if a search team visits location j immediately after visiting location i.
- t_i : Non-negative variable, time at which a search team completes inspection at location i.
- $\mathcal{N}^+(i)$: Set of immediate successors of node i in graph G, i.e., $\mathcal{N}^+(i) = \{j \in \mathcal{N} | (i,j) \in \mathcal{E}\}$.
- $\mathcal{N}^-(i)$: Set of immediate predecessors of node i in graph G, i.e., $N^-(i) = \{j \in \mathcal{N} | (j,i) \in \mathcal{E}\}$.
- M : A big number.

Using this notation, the Searcher's subproblem can be formulated as follows:

$$\text{Minimize} \quad \sum_{i \in \mathcal{N}_h} C_i \sum_{l \in \mathcal{L}'} z_i^l y_l t_i \tag{7.18}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{N}^+(i)} x_{ij} = 1 \quad \forall i \in \mathcal{N}_h, \tag{7.19}$$

$$\sum_{j \in \mathcal{N}^+(i)} x_{ij} - \sum_{j \in \mathcal{N}^-(i)} x_{ji} = 0, \quad \forall i \in \mathcal{N}_h, \tag{7.20}$$

$$\sum_{i \in \mathcal{N}^+(0)} x_{0i} = S, \tag{7.21}$$

$$t_i + v_j + d_{ij} - t_j \leq M(1 - x_{ij}), \quad \forall (i,j) \in \mathcal{E}, j \neq 0, \tag{7.22}$$

$$x_{ij} \in \{0,1\}, \quad \forall (i,j) \in \mathcal{E}, \tag{7.23}$$

$$t_i \geq 0, \quad \forall i \in \mathcal{N}. \tag{7.24}$$

In this formulation, the objective function (7.18) is minimizing the reduced cost. Constraint (7.19) ensures that each location is visited exactly once. Constraint (7.20) is the flow conservation constraint for the search teams. Constraint (7.21) ensures that each search team exits the origin exactly once. Constraint (7.22) computes the visit times for each location. This constraint also eliminates the infeasible subtours.

The Searcher's subproblem is a generalization of the multiple travelling repairman problem [98] with weighted delays. Because the travelling repairman problem is NP-hard, the Searcher's subproblem is also NP-hard. This indicates that the Searcher's subproblems are hard to solve. Therefore, in the next subsections, we develop efficient algorithms to solve the Searcher's subproblems.

Remark 7.4. *Even though the Searcher's subproblem is NP-hard in general, if $S = 1$ and $d_{ij} = d_j$, for all $(i,j) \in \mathcal{E}$, then this problem can be solved in polynomial time. Specifically, in this case, the Searcher's subproblem is equivalent to a single machine scheduling problem to minimize the weighted sum of completion times. This problem can be solved by using Smith's rule [135], that is visiting nodes by non-increasing* *order of* $\frac{C_i \sum_{l \in \mathcal{L}'} z_i^l y_l}{d_i + v_i}$.

A Simulated Annealing Search Strategy Generator for the Searcher's Subproblem

In this subsection, we develop a simulated annealing (SA) algorithm to rapidly obtain a high quality solution to the Searcher's subproblem. SA is a probabilistic search method to approximate the global optimal solution in a large search space [76]. Our proposed SA algorithm starts with an initial solution obtained from the strategies that are used with a positive probability in the current LPM-RS. We then randomly generate a neighborhood solution by applying one of the following operations: (1) randomly selecting two hiding locations and swapping their places in the search schedule. (2) randomly selecting a hiding location and randomly assigning it to another search team. If the newly generated solution leads to a better objective function than the current solution, then it replaces the current solution. The new solution may still replace the current solution even if it leads to a worse objective function value. This happens with probability $e^{-\Delta/T}$, where Δ is the amount of deterioration in the objective function if the new solution replaces the current solution and T is a parameter called temperature. The algorithm starts with a relatively high temperature and reduces the temperature as the algorithm proceeds. Therefore, in the initial iterations, the algorithm tends to explore more areas in the solution space and in the final phases, it tries to exploit the current area.

SA is an efficient algorithm in providing a fast high quality solution. However, it cannot prove optimality of the solution. Therefore, when SA fails to result in an improving solution, we will need an exact solution method to prove optimality. To this end, we develop a branch and price algorithm to solve the Searcher's subproblem to optimality.

A Branch and Price Algorithm to Solve the Searcher's Subproblem

In this section, we propose a branch and price (BP) algorithm to solve the Searcher's subproblem. To this end, we use Dantzig-Wolfe decomposition to reformulate the problem as a set covering model and develop a branch and price algorithm to solve it. Let Ω denote the set of all admissible search schedules for the search teams. The expected damage of search schedule $s \in \Omega$ is: $ED_s = \sum_{i \in \mathcal{N}_h} C_i \sum_{l \in \mathcal{L}'} z_i^l y_l t_{is}$, where t_{is} is the time at which schedule s completes inspection at location i. Let a_{is} denote a binary parameter indicating if search schedule s visits location i. In other words, a_{is} is equal to 1 if search schedule s visits location i, 0 otherwise. For each schedule s, binary variable θ_s is equal to 1 if the schedule s is assigned to a search team, 0 otherwise. Using this notation, the following set covering formulation can be written for the Searcher's subproblem:

$$\text{LMP} \qquad \text{Minimize} \quad \sum_{s \in \Omega} ED_s \theta_s \tag{7.25}$$

$$\text{subject to} \quad \sum_{s \in \Omega} a_{is} \theta_s \geq 1, \quad \forall i \in \mathcal{N}_h, \tag{7.26}$$

$$\sum_{s \in \Omega} \theta_s \leq S, \tag{7.27}$$

$$\theta_s \in \{0, 1\}, \quad \forall s \in \Omega. \tag{7.28}$$

The objective function (7.25) minimizes the expected total damage corresponding to the selected search schedules. Constraint (7.26) indicates that every location needs to be visited at least once. Constraint (7.27) ensures that the number of selected search schedules is limited by the number of search teams. Constraint (7.28) is the integrality constraint.

Column generation

In this subsection, we develop a column generation approach to solve the linear programming relaxation of the model (7.25)-(7.28), with the addition of appropriate branching decisions. We call the linear relaxation of the model (7.25)-(7.28) the Linear Master Problem (LMP). The optimal solution to LMP is a lower bound to the corresponding node in the branch and bound tree. Column generation algorithm starts with a small subset of columns $\Omega' \subset \Omega$ and generates new columns as needed. LMP with restricted columns is called restricted linear master problem (RLMP) and the problem of finding a new column is called the pricing subproblem. In other words, given the dual solution of the current RLMP, the goal of the pricing subproblem is to find columns in $\Omega \setminus \Omega'$ with negative reduced cost. If we are unable to find such a column, then the optimal solution of current RLMP is the optimal solution of LMP and we can terminate the procedure. Otherwise, we add the new columns to RLMP and repeat the process.

The pricing subproblem

At each branch and bound node, the column generation procedure is performed to get a lower bound for the Searcher's subproblem. The RLMP is developed as follows:

$$\begin{aligned}
\text{Minimize} \quad & \sum_{s\in\Omega'} ED_s\theta_s && (7.29)\\
\text{subject to} \quad & \sum_{s\in\Omega'} a_{is}\theta_s \geq 1, \quad \forall i \in \mathcal{N}_h, && (7.30)\\
& \sum_{s\in\Omega'} \theta_s \leq S, && (7.31)\\
& 0 \leq \theta_s \leq 1, \quad \forall s \in \Omega'. && (7.32)
\end{aligned}$$

To check if the optimal solution of the current RLMP is optimal for LMP, we solve the pricing subproblem. In other words, we look for a new column with a negative reduced cost. We use $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ and μ to denote the corresponding dual variables for constraints (7.30) and (7.31), respectively. We use $(\hat{\pi}, \hat{\mu})$ to denote the optimal dual solution for the current RLMP. Using this dual solution, the reduced cost of schedule s is $RC_s = ED_s - \sum_i \hat{\pi}_i a_{is} - \hat{\mu}$. Hence, we can formulate the pricing subproblem as follows:

$$\text{Minimize} \quad \sum_{i\in\mathcal{N}_h}(C_i\sum_{l\in\mathcal{L}'}z_i^l y_l t_i - \hat{\pi}_i\sum_{j\in\mathcal{N}^+(i)}x_{ij}) - \hat{\mu} \tag{7.33}$$

$$\text{subject to} \quad \sum_{j\in\mathcal{N}^+(0)}x_{0j} = 1, \tag{7.34}$$

$$\sum_{j\in\mathcal{N}^+(i)}x_{ij} = \sum_{j\in\mathcal{N}^-(i)}x_{ji}, \quad \forall i\in\mathcal{N}_h, \tag{7.35}$$

$$\sum_{j\in\mathcal{N}^+(i)}x_{ij} \leq 1, \quad \forall i\in\mathcal{N}_h, \tag{7.36}$$

$$t_i + v_j + d_{ij} - t_j \leq M(1-x_{ij}), \quad \forall (i,j)\in\mathcal{E}, j\neq 0, \tag{7.37}$$

$$x_{ij}\in\{0,1\}, \quad \forall (i,j)\in\mathcal{E}. \tag{7.38}$$

In this formulation, the objective function (7.33) is to minimize the reduced cost. Constraint (7.34) ensures that the search schedule starts from the origin. Constraint (7.35) is the flow conservation constraint. Constraint (7.36) ensures that each location is visited at most once. Constraint (7.37) indicates that, if location j is visited immediately after location i, then the visit time of location j is at least $t_i + v_j + d_{ij}$. Constraint (7.38) is the integrality constraint for variable x_{ij}.

A special case of this pricing subproblem has been shown to be NP-hard in [98]. Therefore, this pricing subproblem is also NP-hard. Thus, solving the formulation (7.33)-(7.38) directly maybe computationally expensive. To this end, we propose a branch, bound and remember (BBR) algorithm to solve the pricing subproblems. BBR is a branch and bound algorithm that uses memory to avoid revisiting partial solutions that have already been visited. In BBR, before branching on a partial solution, it is looked up in the memory to see if it has already been visited. This idea has been used in different fields of combinatorial optimization [70, 102, 127]. The details of the proposed BBR algorithm are as follows.

- Branching: The branch and bound algorithm explores partial solutions through an enumeration tree. We denote a partial solution as $\mathcal{P} = (U, l_1, l_2, \ldots, l_p)$, where U is the sets of unvisited locations; p is the number of nodes visited by the partial solution and l_i is the ith visited location for $i = 1, 2, \ldots, p$. Branching on a partial solution means extending it by removing one of the locations from U and adding it to the list of visited locations. This leads to new partial solutions that need to be evaluated by computing upper and lower bounds for their objective function.

- Lower Bound: To obtain a lower bound for a partial solution, we use the Lagrangian relaxation method. We relax the constraint that each location needs to be visited at most once. Therefore, we have the following Lagrangian problem:

$$\text{Minimize} \quad \sum_{i \in \mathcal{N}_h} (C_i \sum_{l \in \mathcal{L}'} z_i^l y_l t_i - \hat{\pi}_i \sum_{j \in \mathcal{N}^+(i)} x_{ij}) - \hat{\mu} + \sum_{i \in \mathcal{N}_h} \lambda_i (1 - \sum_{j \in \mathcal{N}^+(i)} x_{ij}) \tag{7.39}$$

$$\text{subject to} \quad \sum_{j \in \mathcal{N}^+(0)} x_{0j} = 1, \tag{7.40}$$

$$\sum_{j \in \mathcal{N}^+(i)} x_{ij} = \sum_{j \in \mathcal{N}^-(i)} x_{ji}, \tag{7.41}$$

$$t_i + v_j + d_{ij} - t_j \leq M(1 - x_{ij}), \quad \forall (i,j) \in \mathcal{E}, j \neq 0, \tag{7.42}$$

$$x_{ij} \in \{0, 1\}, \quad \forall (i,j) \in \mathcal{E}, \tag{7.43}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N)$, $\lambda_i \geq 0$ is a vector of Lagrangian multipliers. This problem can be solved in pseudo-polynomial time using a dynamic programming approach. It is well-known that, for any vector of Lagrangian multipliers $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N)$, $\lambda_i \geq 0$, the optimal solution to the Lagrangian problem (7.39)-(7.43) gives a lower bound to the original problem (7.33)-(7.38). Moreover, the optimal solution to the Lagrangian problem (7.39)-(7.43) is a concave function of λ. We use a subgradient algorithm to estimate the optimal Lagrangian multipliers.

- Use of memory: BBR memorizes already visited partial solutions. Before considering a partial solution $\mathcal{N} = (U, l_1, l_2, \ldots, l_p)$, BBR checks the memory to see if there is a partial solution with the same set of unvisited nodes, U, and the same current location, l_p, if there is such a partial solution $\mathcal{M} = (U' = U, l'_1, l'_2, \ldots, l'_p = l_p)$, the objective function computed so far by $\mathcal{N}$ is greater than or equal to the objective function computed so far by $\mathcal{M}$ and the time computed so far by $\mathcal{N}$ is greater than or equal to the time computed so far by $\mathcal{M}$, then $\mathcal{N}$ is dominated by $\mathcal{M}$ and $\mathcal{N}$ can be pruned. BBR uses a hash table to store already visited partial solutions along with their computed objective function and time values.

- Search strategy: Preliminary numerical tests revealed that the best first search (BFS) strategy, in which nodes with smaller lower bounds have higher priority of being selected, was unable to rapidly find the optimal solution for some problems. Using BFS, the search algorithm tends to spend a lot of time exploring the nodes in the middle of the enumeration tree rather than choosing the nodes that are deeper in the enumeration tree. Therefore, the algorithm generated very few complete solutions before exploring all of the nodes in the middle of the enumeration tree. To avoid this issue, we use the cyclic best first search (CBFS) strategy [71, 72, 128, 103] in our BB&R algorithm. CBFS systematically chooses the best nodes at all possible depths in the enumeration tree. Specifically, starting by choosing the best nodes at depth 1, CBFS continues to choose the best nodes at deeper levels in the enumeration tree until it reaches the deepest level. At this point, the algorithm goes back to depth 1 and repeats this process until all nodes are explored.

Branching strategy

At each node of the branch and bound tree, we use the column generation algorithm to obtain an optimal solution to the linear relaxation of the model (7.25)-(7.28). The resulting value is a lower bound for the corresponding node. If this lower bound is greater than or equal to the current upper bound, then the node is pruned. Otherwise, we must branch further. If the solution obtained at the current node is integral and the solution value is smaller than the current upper bound, then the upper bound is updated.

Branching on θ_s variables is not beneficial. Fixing θ_s variables at 0 leads to stopping BBR from generating a set of specific schedules, which complicates the solution of the pricing subproblem. Moreover, ruling out a specific schedule is not possible until almost all of the schedule has already been considered by the enumeration tree. Therefore, branching on θ_s variables is not helpful in ruling out candidate solutions early on in the enumeration tree. Thus, it is better to opt for other branching strategies that are more compatible with the pricing subproblem and BBR algorithm. In our proposed branch and price algorithm, we branch on the edges. We choose the edge (i, j) with fractional flow $\hat{x}_{ij}$ farthest to an integer value to branch on. The flow on each edge can be calculated as $\hat{x}_{ij} = \sum_{s \in \Omega'} \theta_s b_{ijs}$, where b_{ijs} is a binary parameter which is equal to 1 if edge (i, j) is used in schedule s, 0 otherwise. After selecting the edge (i, j) to branch on, two child nodes are created: by setting $x_{ij} = 0$ and $x_{ij} = 1$. Fixing $x_{ij} = 0$ implies that the edge (i, j) is forbidden. To enforce this constraint in the pricing subproblem, we delete edge (i, j) from E and, for all schedules s containing edge (i, j), we remove the corresponding variable θ_s from the RLMP. Setting $x_{ij} = 1$ implies that the edge (i, j) must be used. To enforce this constraint in the pricing subproblem, we eliminate all of the edges (i, j') and (i', j) with $i \neq i'$ and $j \neq j'$. Note that, branching on the edges only leads to changes in the underlying graph and it does not require modifying the BBR algorithm.

7.3.2 The Hider's Subproblem

In this section, we formulate the Hider's subproblem to obtain an improving hiding strategy. The Hider's subproblem is formulated as follows:

$$\text{Maximize} \quad \sum_{i \in \mathcal{N}_h} z_i C_i \sum_{k \in \mathcal{K}} t_i^k x_k \tag{7.44}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{N}_h} z_i \leq H, \tag{7.45}$$

$$z_i \in \{0,1\}, \quad \forall i \in \mathcal{N}_h. \tag{7.46}$$

In this formulation, z_i is a binary variable which is equal to 1 if the Hider hides an object in location i. Equation (7.44), the objective function, is the expected total damage. Equation (7.45) corresponds to the constraint on the number of hidden objects. Finally, constraint (7.46) is the integrality constraint for variable z_i.

The Hider's subproblem is a special case of 0-1 knapsack problem with unit item weights. This problem can be solved in polynomial time by sorting the hiding locations i in a non-increasing order of $C_i \sum_{k \in \mathcal{K}} t_i^k$ and choosing the first H hiding locations.

7.3.3 Overall Solution Procedure and Bounds

Algorithm 5 provides the pseudo-code for the overall solution procedure. The column and row generation algorithm begins by randomly generating a set of initial strategies. Then, using this set of strategies, the LPM-RS is solved to obtain a solution $\overline{\mathbf{x}}$ and a vector of dual values $\overline{\mathbf{y}}$. Dual values $\overline{\mathbf{y}}$ are then used in the Searcher's subproblem to generate a new search strategy. If a new search strategy with a smaller expected total damage is obtained, it is added to $\mathcal{K}'$. Then the Hider's subproblem is solved to generate a new hiding strategy. If a new hiding strategy with a greater expected total damage is obtained, it is added to $\mathcal{L}'$. If, during the last two steps, either $\mathcal{K}'$ or $\mathcal{L}'$

has been updated, then the process is repeated; otherwise the procedure terminates. Because the number of possible strategies for both players is finite, the algorithm terminates after a finite number of iterations. Moreover, when the algorithm terminates, no player can improve the expected total damage in their own favor by changing their strategies. Therefore, by definition, the algorithm returns a Nash equilibrium point upon termination. During solution procedure, we have access to lower and upper

Algorithm 5: Pseudo-code for the overall solution algorithm

```
1  Initialize sets $\mathcal{K}'$ and $\mathcal{L}'$.
2  Solve LPM-RS. Let $\overline{\mathbf{x}} = [x_k]$, $\overline{\mathbf{y}} = [y_l]$ and $u$ be the optimal primal solution,
     dual solution and objective function value, respectively.
3  Solve the Searcher's subproblem using $\overline{\mathbf{y}}$ as dual values and let $\mathbf{t}^* = [t_i^*]$
     denote the optimal solution.
4  if $v > \sum_{i \in \mathcal{N}_h} t_i^* C_i \sum_{l \in \mathcal{L}'} z_i^l y_l$ then
5      Add the new search strategy $\mathbf{t}^*$ to $\mathcal{K}'$.
6  end
7  Solve the Hider's subproblem using $\overline{\mathbf{x}}$ as primal values and let $\mathbf{z}^* = [z_i^*]$ be
     the optimal solution.
8  if $\sum_{i \in \mathcal{N}_h} z_i^* C_i \sum_{k \in \mathcal{K}'} t_i^k x_k > v$ then
9      Add the new hiding strategy $\mathbf{z}^*$ to $\mathcal{L}'$.
10 end
11 if $\mathcal{K}'$ or $\mathcal{L}'$ has been updated then
12     Go to Line 2.
13 else
14     Return $v$ as the value of the game.
15     Terminate the procedure.
16 end
```

bounds on the value of the game so that we can terminate the algorithm when a desired solution quality is reached. The following lemma offers solution bounds on the value of the game which can be computed in every iteration of the solution algorithm.

Lemma 7.2. *Optimal solution to the Searcher's (Hider's) subproblem yields a lower bound (an upper bound) to the expected total damage in equilibrium.*

Proof. Because the Hider's strategy set is restricted, i.e. $\mathcal{L}' \subseteq \mathcal{L}$, solving the Searcher's subproblem leads to a lower bound on the expected total damage in equilibrium. Similarly, because the Searcher's strategy space is restricted, solving the Hider's subproblem leads to an upper bound on the value of the game. □

Remark 7.5. *Note that, in order for the bound in Lemma 7.2 to be valid, the Searcher's subproblem should be solved to optimality. In general, this bound is not monotone over the iterations, this is called the yo-yo effect [97].*

7.4 Numerical Experiments

In this section, we perform computational experiments to investigate efficiency of the proposed algorithms and gain insight on some properties of the game. The algorithms are coded in C++ and CPLEX 12.8 solver has been used to solve the LPs and the pricing subproblems. The computational experiments are performed on a computer with 2.6 GH processor and 32 GB of RAM. We used a maximum running time of 2 hours (7200 seconds), for all of the algorithms involved in this section.

Our base set of test instances consists of randomly generated instances. The location of the potential hiding places are randomly generated on a hypothetical square with side of n units. Manhattan distance is used to compute the travel times, d_{ij}, between these locations. Location weights, C_i, are generated randomly from the range $[1, N]$. Similarly, the visit times, v_i, are generated randomly from the range $[1, N]$.

In our first experiment, we compare the performances of different methods for solving the Searcher's subproblem. Specifically, we consider two cases: the flow-type mathematical formulation (MF) of (7.18) to (7.24) and the branch and price algorithm (BP) developed in section 7.3.1. We consider 16 instances for each problem size, with various values of $S \in \{2, 3, 4, 5\}$ and $H \in \{1, 2, 3, 4\}$. Tables 7.1 and 7.2 compare the average run times of these algorithms for different values of the number of search teams and the number of hidden objects, respectively. Based on these tables, for all problem sizes, BP performs significantly better than MF. Moreover, for both MF and BP algorithms, the running time increases as n increases and it, generally, decreases as S increases. However, no clear pattern is observed on the effect of increasing H on the running time.

Tables 7.3 and 7.4 demonstrate the average optimality gap, in percent, obtained for different number of search teams and hidden objects, respectively. In these experiments, while both algorithms are able to find the optimal solution for problems of smaller size, BP performs significantly better than MF for larger-sized problems. Moreover, for both MF and BP algorithms, the average gap time increases as N increases and it decreases as S increases. However, no clear pattern is observed on the effect of increasing H on the average optimality gap.

Figure 7.1 shows the convergence of the solution for different problem sizes. The number of iterations needed for convergence increases as the problem size increases. Moreover, the bounds are not monotone over the iterations and the yo-yo effect is visible due to the non-monotonicity of the bound in Lemma 7.2. Another interesting observation is that, the upper bound values stabilize much earlier than the termination of the algorithm. This means that after the upper bound values stabilize, we can

Table 7.1: Average running times for different number of search teams (in seconds)

N	S=2		S=3		S=4		S=5	
	MF	BP	MF	BP	MF	BP	MF	BP
10	2.02	0.26	1.42	0.34	0.82	0.15	0.52	0.32
15	15.67	2.59	8.01	2.28	3.94	0.69	1.11	0.18
20	7200.00	83.82	838.34	27.42	60.04	40.62	13.83	2.71
25	7200.00	7200.00	7200.00	2696.27	7200.00	4055.85	4537.37	520.81
30	7200.00	7200.00	7200.00	7200.00	7200.00	5480.06	7200.00	1848.59

Table 7.2: Average running times for different number of hidden objects (in seconds)

N	H=1		H=2		H=3		H=4	
	MF	BP	MF	BP	MF	BP	MF	BP
10	1.34	0.53	1.08	0.18	1.14	0.18	1.19	0.19
15	4.00	0.71	6.83	0.80	9.37	1.32	8.51	2.91
20	1914.93	38.84	1913.25	21.70	2015.77	51.84	2268.26	42.18
25	5506.43	4214.13	6230.94	4296.92	7200.00	2787.20	7200.00	3174.74
30	7200.00	5192.93	7200.00	5420.28	7200.00	4901.70	7200.00	6214.18

Table 7.3: Average optimality gap for different number of search teams (%)

N	S=2		S=3		S=4		S=5	
	MF	BP	MF	BP	MF	BP	MF	BP
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	43.50	0.40	35.33	0.00	6.49	0.12	0.50	0.00
30	130.27	8.31	75.61	5.94	45.30	0.10	9.77	0.03

Table 7.4: Average optimality gap for different number of hidden objects (%)

N	H=1		H=2		H=3		H=4	
	MF	BP	MF	BP	MF	BP	MF	BP
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.32	0.00	0.23	0.00	0.08	0.00	0.10	0.00
25	22.78	0.12	18.94	0.23	21.11	0.13	23.01	0.05
30	61.12	4.60	61.36	2.72	53.90	2.41	84.56	4.65

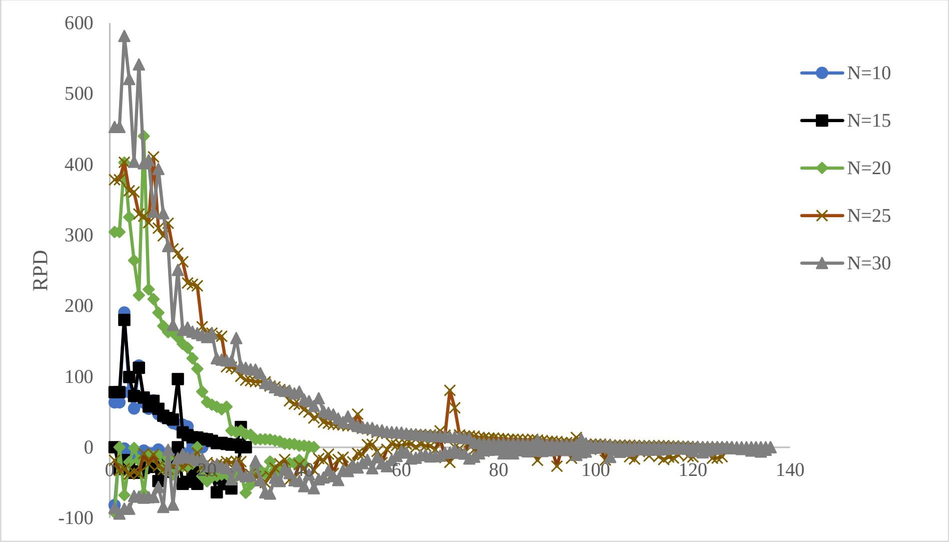


Figure 7.1: Convergence of the proposed column and row generation algorithm

terminate the solution algorithm without undermining the solution quality too much.

Next, we study the effect of the number of search teams and hiding objects on the expected total damage in equilibrium. An experiment is designed with $S \in \{2, 3, 4, 5\}$ and $H \in \{1, 2, 3, 4\}$. Figure 7.2 demonstrates that, as the number of search teams increases, the expected damage decreases. Moreover, a diminishing returns effect is visible in the reduction in expected total damage for each unit increment in S. Figure 7.3 shows that the expected total damage increases almost linearly with the number of hidden objects. Moreover, the rate of increase is greater for smaller number of search teams. In our next experiment, we study the effect of the Searcher's deviation from the equilibrium on the value of the game. Specifically, we consider the case that the Searcher, in deriving her strategy, mistakenly, thinks that the node weights are the same. We compare the weighted search time values in this case with the weighted search time values in equilibrium. Table 7.5 show the obtained results for different problem sizes. In this tables, the "ENWS" (Equal Node Weight Strategy) column shows the expected weighted search time when the Searcher deviates from the Nash

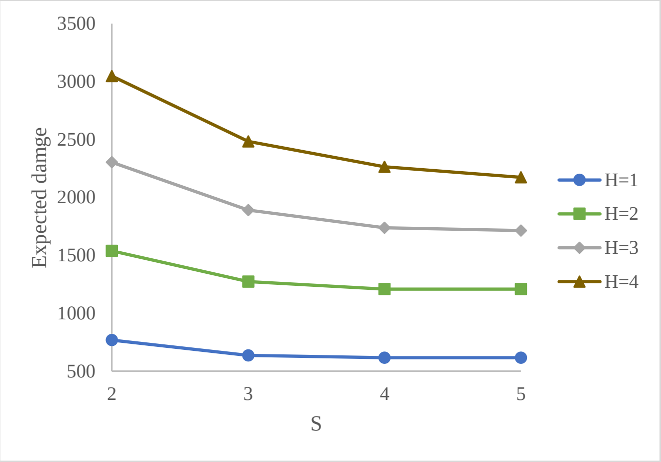


Figure 7.2: The effect of number of search teams on the expected total damage in equilibrium

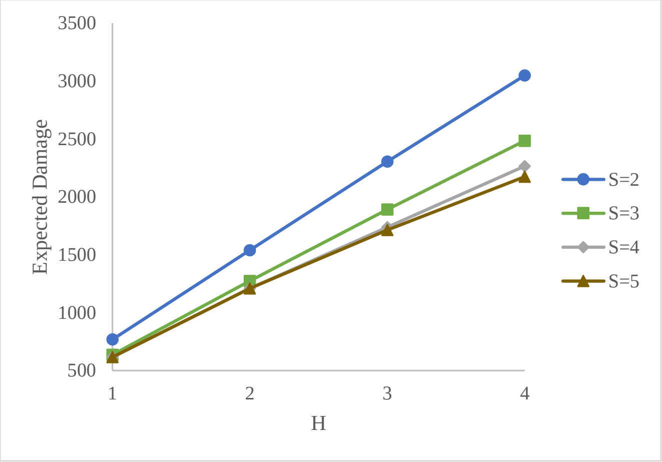


Figure 7.3: The effect of number of hidden objects on the expected total damage in equilibrium

strategy to use a strategy based on equal node weights assumption. “NE” column shows the expected weighted search time in the Nash Equilibrium and “Increase (%)” column shows the percentage of increase in the expected weighted search time as a result of the Searcher’s deviation from the equilibrium. As seen in these tables, deviation of the Searcher from the Nash equilibrium can lead to an increase of up to 63 percent in expected weighted search time. This increase is at least 9 percent across all problem instances. This highlights the importance of using the information about node weights.

Table 7.5: Comparison of NE with ENWS for moderate instances

N	NE	ENWS	Increase(%)
10	165.00	270.00	63.64
15	384.00	422.38	9.99
20	616.00	904.29	46.80
25	1118.00	1511.80	35.22
30	1856.00	2610.00	40.63

Chapter 8

Conclusions

This dissertation develops game-theoretic models to determine protection strategies for infrastructures including their users. The game models studied in this research include resource allocation models, as well as patrolling and search games. These models address the strategic decision making process of competing agents: defender vs attacker in resource allocation and patrolling games, and searcher vs hider in hide-and-seek games. Most of the existing resource allocation models assume that the parameters of the game are either deterministic or follow a known distribution. Whereas in reality, some parameters of the game may be uncertain with no known distribution or distributional information about them may be unreliable. To this end, we look at one-shot security games under uncertainty about site valuations. We propose a model where both players use a robust approach to contend with the uncertainty of site valuations. We then apply our model to a real case of assigning grant resources to 10 urban areas in the United States. Another limitation is the lack of models that address hierarchical decision making. Protecting infrastructures and their users against terrorism involves making both strategic and operational decisions in an organization's hierarchy. Although usually analyzed separately, these decisions influence each other. To address this issue, we develop a two-stage game model where in the first stage, the players make investment decisions and in the second stage, they decide which sites to defend/attack. We then characterize the existence and uniqueness conditions for the Nash equilibrium of this game. Afterwards, we apply the proposed model to a real case. Another limitation of the existing resource allocation models is that most of the models with overarching protection options only consider a single option that covers all targets. However, in reality, there maybe multiple overarching protection options and each option may cover only a subset of targets. To address this issue, we develop a new resource allocation model with generalized overarching protection options. Specifically, we assume that there are multiple overarching protection options and each option covers a subset of targets. We also develop an efficient decomposition algorithm to solve this problem.

In the second part of the dissertation, we study patrolling and search games. Most of the patrolling game models assume that the site values are either the same or that they do not change over time. However, this is not a realistic assumption. Particularly in the case of soft targets, these values may be different and may change over time. We propose new models with dynamically changing node values and node-based attack times. We solve these models numerically using algorithms like column generation, and column and row generation. We then apply these models to a real case of an urban rail network in a major US city. In the area of search games, most of the models assume that the hiding places are identical and the players' objective is to optimize the search time. However, there are some cases in which the players may differentiate the hiding places from each other and the objective is to optimize a weighted search time. To address this, we introduce a new discrete search game with consideration given to the weights at different locations. For a special case of the problem, we show that the game has a closed-form Nash equilibrium. For the general case, we develop an algorithm based on column and row generation. We show that the Searcher's subproblem is NP-hard and propose a branch and price algorithm to solve it. We also present a polynomial time algorithm for the Hider's subproblem. Numerical experiments investigate the performance of the approach and reveal insights on the properties of this game.

This dissertation addresses some of the gaps existing in the literature of resource allocation for security as well as patrolling and search games. However, the current models can be further extended to address other remaining issues. For example, in the two-stage invest/defend game we are interested in further investigating multi-period invest-defend games with multiple defenders. Another area of interest is the effect of investment transparency vs secrecy, and of side information on the defense policy. In the area of security resource allocation models, there are many ways to extend the current resource allocation models to include generalized overarching protection options. One way to extend this model is to consider all-hazard protection options. Some of the protection alternatives may protect against both terrorism and natural disasters; e.g., hardening a bridge. Using a similar argument to the proof of Lemma 4.1, we can show that the resource allocation model still remains a convex optimization problem after adding all-hazard protection options. Developing a decomposition approach for the problem with the addition of all-hazard protection options is recommended as a future research idea in this area. Another extension is to incorporate discrete decision variables into the model in case protection decisions are not continuous and it is more appropriate to model them as binary or integer variables. The addition of discrete decision variables into the model will make the problem a mixed integer nonlinear program (MINLP). We expect the resulting MINLP to be amenable to outer approximation approaches similar to the one proposed in [28].

In the area of patrolling and search games, accommodating mobile adversaries and imperfect detection is an interesting idea. Most of the models in the literature of patrolling games assume that once the patroller and the attacker are in the same location, the patroller will successfully capture the attacker. However, this assumption may not be realistic and patroller's detection and capturing capabilities may be imperfect. Moreover, most of the models assume that all of the targets are equally accessible to the attacker and attacker can jump on the intended target eliminating the need to traverse the graph and the risk of being caught in the process. However, in reality attacker may get caught in the process of moving towards the target site. Therefore, developing a model with entrance nodes and imperfect detection, is an interesting research topic in this area.

Published and Submitted Papers

The following is a list of published and submitted papers as a result of this dissertation:

1. Yolmeh, A., Baykal Gürsoy, M., ‘Weighted network search games with multiple hidden objects and multiple search teams” *European Journal of Operational Research*, 2020.

2. Yolmeh, A., Baykal Gürsoy, M., “Patrolling Games on General Graphs with Time-Dependent Node Values” *Military Operations Research*, 24(2), 17-30, 2019.

3. Yolmeh, A., Baykal Gürsoy, M., “Two-stage invest-defend game: balancing strategic and operational decisions” *Decision Analysis*, 16(1):46-66, 2019.

4. Yolmeh, A., Baykal Gürsoy, M., “Urban Rail Patrolling: A Game Theoretic Approach” *Journal of Transportation Security*, 11(1-2): 23-40, 2018.

5. Yolmeh, A., Baykal Gürsoy, M., “A robust approach to the infrastructure security games.” *Computers & Industrial Engineering*, 110: 515-526. 2017.

Bibliography

[1] Michele Aghassi and Dimitris Bertsimas. Robust game theory. *Mathematical Programming*, 107(1-2):231–273, 2006.

[2] Steve Alpern. Search games on trees with asymmetric travel times. *SIAM Journal on Control and Optimization*, 48(8):5547–5563, 2010.

[3] Steve Alpern. Find-and-fetch search on a tree. *Operations Research*, 59(5):1258–1268, 2011.

[4] Steve Alpern. Hide-and-seek games on a network, using combinatorial search paths. *Operations Research*, 65(5):1207–1214, 2017.

[5] Steve Alpern, Robbert Fokkink, Joram Op Den Kelder, and Tom Lidbetter. Disperse or unite? a mathematical model of coordinated attack. In *International Conference on Decision and Game Theory for Security*, pages 220–233. Springer, 2010.

[6] Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*, volume 55. Springer Science & Business Media, 2006.

[7] Steve Alpern and Thomas Lidbetter. Mining coal or finding terrorists: The expanding search paradigm. *Operations Research*, 61(2):265–279, 2013.

[8] Steve Alpern and Thomas Lidbetter. Searching a variable speed network. *Mathematics of Operations Research*, 39(3):697–711, 2013.

[9] Steve Alpern and Thomas Lidbetter. Optimal trade-off between speed and acuity when searching for a small object. *Operations Research*, 63(1):122–133, 2015.

[10] Steve Alpern, Alec Morton, and Katerina Papadaki. Patrolling games. *Operations Research*, 59(5):1246–1257, 2011.

[11] David Assaf and Shmuel Zamir. Continuous and discrete search for one of many objects. *Operations Research Letters*, 6(5):205–209, 1987.

[12] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

[13] Nicola Basilico, Giuseppe De Nittis, and Nicola Gatti. Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence*, 246:220–257, 2017.

[14] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence*, 184:78–123, 2012.

[15] Vic Baston and Kensaku Kikuta. Search games on a network with travelling and search costs. *International Journal of Game Theory*, 44(2):347–365, 2015.

[16] Melike Baykal-Gürsoy, Zhe Duan, H Vincent Poor, and Andrey Garnaev. Infrastructure security games. *European Journal of Operational Research*, 239(2):469–478, 2014.

[17] Vicki M Bier and Vinod Abhichandani. Optimal allocation of resources for defense of simple series and parallel systems from determined adversaries. In *Proceedings of the engineering foundation conference on risk-based decision making in water resources X, Santa Barbara, CA: American Society of Civil Engineers*, pages 59–76, 2002.

[18] Vicki M Bier, Naraphorn Haphuriwat, Jaime Menoyo, Rae Zimmerman, and Alison M Culpen. Optimal resource allocation for defense of targets based on differing measures of attractiveness. *Risk Analysis*, 28(3):763–770, 2008.

[19] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization.* Cambridge university press, 2004. Ex. 5.32.

[20] Joseph Bram. A 2-player n-region search game. Technical report, Center for Naval Analyses Alexandria Va Operations Evaluation Group, 1963.

[21] Gerald Brown, Matthew Carlyle, Javier Salmerón, and Kevin Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, 2006.

[22] Bureau of Counterterrorism. National consortium for the study of terrorism and responses to terrorism: Annex of statistical information, June 2016.

[23] Gerard P Cachon and Serguei Netessine. Game theory in supply chain analysis. In *Handbook of Quantitative Supply Chain Analysis*, pages 13–65. Springer, 2004.

[24] Jan M Chaiken and Peter Dormont. A patrol car allocation model: Capabilities and algorithms. *Management Science*, 24(12):1291–1300, 1978.

[25] Kenneth Chelst. An algorithm for deploying a crime directed (tactical) patrol force. *Management Science*, 24(12):1314–1327, 1978.

[26] Arnon Dagan and Shmuel Gal. Network search games, with arbitrary searcher starting point. *Networks*, 52(3):156–161, 2008.

[27] Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the National Academy of Sciences*, 38(10):886–893, 1952.

[28] Roger Fletcher and Sven Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical programming*, 66(1-3):327–349, 1994.

[29] Drew Fudenberg and Jean Tirole. Game theory, 1991. *Cambridge, Massachusetts*, 393(12):80, 1991.

[30] Shmuel Gal. Search games with mobile and immobile hider. *SIAM Journal on Control and Optimization*, 17(1):99–122, 1979.

[31] Shmuel Gal. Search games, volume 149 of Mathematics in Science and Engineering, 1980.

[32] Shmuel Gal. Search games: a review. In *Search Theory*, pages 3–15. Springer, 2013.

[33] Mary Lynn Garcia. *Vulnerability assessment of physical protection systems*. Butterworth-Heinemann, 2005.

[34] Andrey Garnaev. *Search games and other applications of game theory*, volume 485. Springer Science & Business Media, 2012.

[35] Andrey Garnaev, Melike Baykal-Gürsoy, and H Vincent Poor. Incorporating attack-type uncertainty into network protection. *IEEE Transactions on Information Forensics and Security*, 9(8):1278–1287, 2014.

[36] Andrey Garnaev, Melike Baykal-Gursoy, and H Vincent Poor. How to deal with an intelligent adversary. *Computers & Industrial Engineering*, 90:352–360, 2015.

[37] Andrey Garnaev, Melike Baykal-Gürsoy, and H Vincent Poor. A game theoretic analysis of secret and reliable communication with active and passive adversarial modes. *IEEE Transactions on Wireless Communications*, 15(3):2155–2163, 2016.

[38] Andrey Garnaev, Melike Baykal-Gürsoy, and H Vincent Poor. Security games with unknown adversarial strategies. 2016.

[39] B John Garrick, James E Hall, Max Kilger, John C McDonald, Tara O'Toole, Peter S Probst, Elizabeth Rindskopf Parker, Robert Rosenthal, Alvin W Trivelpiece, Lee A Van Arsdale, et al. Confronting the risks of terrorism: making the right decisions. *Reliability Engineering & System Safety*, 86(2):129–176, 2004.

[40] Michael R Gary and David S Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.

[41] JC Gittins and DM Roberts. The search for an intelligent evader concealed in one of an arbitrary number of regions. *Naval Research Logistics Quarterly*, 26(4):651–666, 1979.

[42] Pedro Godinho and Joana Dias. A two-player competitive discrete location model with simultaneous decisions. *European Journal of Operational Research*, 207(3):1419–1432, 2010.

[43] Pedro Godinho and Joana Dias. Two-player simultaneous location game: Preferential rights and overbidding. *European Journal of Operational Research*, 229(3):663–672, 2013.

[44] Mohsen Golalikhani and Jun Zhuang. Modeling arbitrary layers of continuous-level defenses in facing with strategic attackers. *Risk Analysis: An International Journal*, 31(4):533–547, 2011.

[45] Boaz Golany, Edward H Kaplan, Abraham Marmur, and Uriel G Rothblum. Nature plays with dice–terrorists do not: Allocating resources to counter strategic versus probabilistic risks. *European Journal of Operational Research*, 192(1):198–208, 2009.

[46] Meigu Guan. Graphic programming using odd and even points. *Chinese Math.*, 1:237–277, 1962.

[47] Peiqiu Guan, Meilin He, Jun Zhuang, and Stephen C Hora. Modeling a multitarget attacker–defender game with budget constraints. *Decision Analysis*, 14(2):87–107, 2017.

[48] Mengran Hao, Shilan Jin, and Jun Zhuang. Robustness of optimal defensive resource allocations in the face of less fully rational attacker. In *IIE Annual Conference. Proceedings*, page 886. Institute of Industrial and Systems Engineers (IISE), 2009.

[49] Naraphorn Haphuriwat and Vicki M Bier. Trade-offs between target hardening and overarching protection. *European Journal of Operational Research*, 213(1):320–328, 2011.

[50] Naraphorn Haphuriwat, Vicki M Bier, and Henry H Willis. Deterring the smuggling of nuclear weapons in container freight through detection and retaliation. *Decision Analysis*, 8(2):88–102, 2011.

[51] Bernard Harris. Mathematical methods in combatting terrorism. *Risk Analysis*, 24(4):985–988, 2004.

[52] John C Harsanyi. Games with incomplete information played by Bayesian players, i–iii: part i. the basic model&. *Management Science*, 50(12_supplement):1804–1817, 1967.

[53] John C Harsanyi. Games with incomplete information played by Bayesian players part II. Bayesian equilibrium points. *Management Science*, 14(5):320–334, 1968.

[54] John C Harsanyi. Games with incomplete information played by Bayesian players, part III. The basic probability distribution of the game. *Management Science*, 14(7):486–502, 1968.

[55] Kjell Hausken. Income, interdependence, and substitution effects affecting incentives for security investment. *Journal of Accounting and Public Policy*, 25(6):629–665, 2006.

[56] Kjell Hausken. Combined series and parallel systems subject to individual versus overarching defense and attack. *Asia-Pacific Journal of Operational Research*, 30(02):1250056, 2013.

[57] Kjell Hausken. Individual versus overarching protection and attack of assets. *Central European Journal of Operations Research*, 22(1):89–112, 2014.

[58] Kjell Hausken. Special versus general protection and attack of parallel and series components. *Reliability Engineering & System Safety*, 165:239–256, 2017.

[59] Kjell Hausken. Special versus general protection and attack of two assets. *Operations Research and Decisions*, 29(4):53–93, 2019.

[60] Kjell Hausken, Vicki M Bier, and Jun Zhuang. Defending against terrorism, natural disaster, and all hazards. In *Game theoretic risk analysis of security threats*, pages 65–97. Springer, 2009.

[61] Kjell Hausken and Fei He. On the effectiveness of security countermeasures for critical infrastructures. *Risk Analysis*, 36(4):711–726, 2016.

[62] Kjell Hausken and Gregory Levitin. Review of systems defense and attack models. *International Journal of Performability Engineering*, 8(4):355–366, 2012.

[63] Kjell Hausken and Jun Zhuang. Governments' and terrorists' defense and attack in a t-period game. *Decision Analysis*, 8(1):46–70, 2011.

[64] Dorit S Hochbaum, Cheng Lyu, and Fernando Ordóñez. Security routing games with multi-vehicle Chinese postman problem. *Networks*, 64(3):181–191, 2014.

[65] Ryusuke Hohzaki. Search games: Literature and survey. *Journal of the Operations Research Society of Japan*, 59(1):1–34, 2016.

[66] Rufus Isaacs. *Differential games*. Wiley, New York, 1965.

[67] Manish Jain, Erim Kardes, Christopher Kiekintveld, Fernando Ordónez, and Milind Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.

[68] Brian Michael Jenkins and Joseph Trella. Carnage interrupted: An analysis of fifteen terrorist plots against public surface transportation. Technical report, 2012.

[69] Victor Richmond R Jose and Jun Zhuang. Technology adoption, accumulation, and competition in multi-period attacker-defender games. *Military Operations Research*, 18(2):33–47, 2013.

[70] A Jouglet, Ph Baptiste, and J Carlier. Branch-and-bound algorithms for total weighted tardiness. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004.

[71] Gio K Kao, Edward C Sewell, and Sheldon H Jacobson. A branch, bound, and remember algorithm for the $1|r_i|\sum t_i$ scheduling problem. *Journal of Scheduling*, 12(2):163, 2009.

[72] Gio K Kao, Edward C Sewell, Sheldon H Jacobson, and Shane N Hall. New dominance rules and exploration strategies for the $1|r_i|\sum U_i$ scheduling problem. *Computational Optimization and Applications*, 51(3):1253–1274, 2012.

[73] Stanley Kaplan and B John Garrick. On the quantitative definition of risk. *Risk analysis*, 1(1):11–27, 1981.

[74] Erim Kardeş. On discounted stochastic games with incomplete information on payoffs and a security application. *Operations Research Letters*, 42(1):7–11, 2014.

[75] Christopher Kiekintveld, Towhidul Islam, and Vladik Kreinovich. Security games with interval uncertainty. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 231–238. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[76] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[77] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Security games with multiple attacker resources. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 273, 2011.

[78] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492, Berkeley, Calif., 1951. University of California Press.

[79] Felipe Lagos, Fernando Ordóñez, and Martine Labbé. A branch and price algorithm for a Stackelberg security game. *Computers & Industrial Engineering*, 111:216–227, 2017.

[80] Giuseppe Lancia and Paolo Serafini. *Compact extended linear programming models*. Springer, 2018.

[81] RE Larson and JL Casti. Principles of dynamic programming, part 1: Basic analytic and computational methods, 1978.

[82] Richard C Larson. *Urban police patrol analysis*, volume 28. MIT Press Cambridge, MA, 1972.

[83] Hoong Chuin Lau, Zhi Yuan, and Aldy Gunawan. Patrol scheduling in urban rail network. *Annals of Operations Research*, 239(1):317–342, 2016.

[84] Gregory Levitin. Optimal distribution of constrained resources in bi-contest detection-impact game. *International Journal of Performability Engineering*, 5(1):45–54, 2009.

[85] Gregory Levitin and Kjell Hausken. Defence and attack of systems with variable attacker system structure detection probability. *Journal of the Operational Research Society*, 61(1):124–133, 2010.

[86] Gregory Levitin and Kjell Hausken. Individual versus overarching protection against strategic attacks. *Journal of the Operational Research Society*, 63(7):969–981, 2012.

[87] Gregory Levitin and Kjell Hausken. Parallel systems under two sequential attacks with imperfect detection of the first attack outcome. *Journal of the Operational Research Society*, 63(11):1545–1555, 2012.

[88] Gregory Levitin and Kjell Hausken. Resource distribution in multiple attacks with imperfect detection of the attack outcome. *Risk Analysis*, 32(2):304–318, 2012.

[89] Gregory Levitin, Kjell Hausken, and Yuanshun Dai. Individual vs. overarching protection for minimizing the expected damage caused by an attack. *Reliability Engineering & System Safety*, 119:117–125, 2013.

[90] Gregory Levitin, Kjell Hausken, and Yuanshun Dai. Optimal defense with variable number of overarching and individual protections. *Reliability Engineering & System Safety*, 123:81–90, 2014.

[91] Thomas Lidbetter. Search games with multiple hidden objects. *SIAM Journal on Control and Optimization*, 51(4):3056–3074, 2013.

[92] Thomas Lidbetter and Kyle Lin. Searching for multiple objects in multiple locations. *arXiv preprint arXiv:1710.05332*, 2017.

[93] Kyle Y Lin, Michael P Atkinson, Timothy H Chung, and Kevin D Glazebrook. A graph patrol problem with random attack times. *Operations Research*, 61(3):694–710, 2013.

[94] Kyle Y Lin, Michael P Atkinson, and Kevin D Glazebrook. Optimal patrol to uncover threats in time when detection is imperfect. *Naval Research Logistics (NRL)*, 61(8):557–576, 2014.

[95] Kyle Y Lin and Dashi I Singham. Robust search policies against an intelligent evader. Technical report, Naval Postgraduate School Monterey United States, 2015.

[96] J. Lou, A. M. Smith, and Y. Vorobeychik. Multidefender security games. *IEEE Intelligent Systems*, 32(1):50–60, Jan 2017.

[97] Marco E Lübbecke. Column generation. *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[98] Zhixing Luo, Hu Qin, and Andrew Lim. Branch-and-price-and-cut for the multiple traveling repairman problem with distance constraints. *European Journal of Operational Research*, 234(1):49–60, 2014.

[99] David Matula. A periodic optimal search. *The American Mathematical Monthly*, 71(1):15–21, 1964.

[100] William L McGill, Bilal M Ayyub, and Mark Kaminskiy. Risk analysis for critical asset protection. *Risk Analysis*, 27(5):1265–1281, 2007.

[101] Richard G McGrath and Kyle Y Lin. Robust patrol strategies against attacks at dispersed heterogeneous locations. *International Journal of Operational Research*, 30(3):340–359, 2017.

[102] Thomas L Morin and Roy E Marsten. Branch-and-bound strategies for dynamic programming. *Operations Research*, 24(4):611–627, 1976.

[103] David R Morrison, Jason J Sauppe, Wenda Zhang, Sheldon H Jacobson, and Edward C Sewell. Cyclic best first search: Using contours to guide branch-and-bound algorithms. *Naval Research Logistics (NRL)*, 64(1):64–82, 2017.

[104] John Moteff. Risk management and critical infrastructure protection: Assessing, integrating, and managing threats, vulnerabilities and consequences. DTIC Document, 2005.

[105] Ibrahim Muter, Ş İlker Birbil, and Kerem Bülbül. Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. *Mathematical Programming*, 142(1-2):47–82, 2013.

[106] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming*, 14(1):265–294, 1978.

[107] Mohammad E Nikoofal and Jun Zhuang. Robust allocation of a defensive budget considering an attacker's private information. *Risk Analysis*, 32(5):930–943, 2012.

[108] Mohammad E Nikoofal and Jun Zhuang. On the value of exposure and secrecy of defense system: First-mover advantage vs. robustness. *European Journal of Operational Research*, 246(1):320–330, 2015.

[109] David G Olson and Gordon P Wright. Models for allocating police preventive patrol effort. *Journal of the Operational Research Society*, 26(4):703–715, 1975.

[110] JB Orlin, RK Ahuja, and TL Magnanti. *Network flows, theory, algorithms and applications.* Prentice Hall, 1993.

[111] Katerina Papadaki, Steve Alpern, Thomas Lidbetter, and Alec Morton. Patrolling a border. *Operations Research*, 64(6):1256–1269, 2016.

[112] Praveen Paruchuri, Jonathan P Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. 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-Volume 2*, pages 895–902. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[113] Praveen Paruchuri, Jonathan P Pearce, Milind Tambe, Fernando Ordonez, and Sarit Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 181. ACM, 2007.

[114] Praveen Paruchuri, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Security in multiagent systems by policy randomization. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 273–280. ACM, 2006.

[115] Elisabeth Pat-Cornell and Seth Guikema. Probabilistic modeling of terrorist threats: A systems analysis approach to setting priorities among countermeasures. *Military Operations Research*, 7(4):5–23, 2002.

[116] Elisabeth Paté-Cornell. Fusion of intelligence information: A Bayesian approach. *Risk Analysis*, 22(3):445–454, 2002.

[117] Rui Peng, Li Guo, Gregory Levitin, Huadong Mo, and Wenbin Wang. Maintenance versus individual and overarching protections for parallel systems. *Quality Technology & Quantitative Management*, 11(3):353–362, 2014.

[118] James Pita, Manish Jain, Janusz Marecki, Fernando Ordóñez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: Industrial Track*, pages 125–132. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[119] Robert Powell. Allocating defensive resources with private information about vulnerability. *American Political Science Review*, 101(4):799–809, 2007.

[120] Robert Powell. Defending against terrorist attacks with limited resources. *American Political Science Review*, 101(3):527–541, 2007.

[121] Yundi Qian, William B Haskell, and Milind Tambe. Robust strategy against unknown risk-averse attackers in security games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1341–1349. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[122] Sebastian Riedel, David Smith, and Andrew McCallum. Parse, price and cut: delayed column and row generation for graph based parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 732–743. Association for Computational Linguistics, 2012.

[123] David Rios Insua, Jesus Rios, and David Banks. Adversarial risk analysis. *Journal of the American Statistical Association*, 104(486):841–854, 2009.

[124] DM Roberts and JC Gittins. The search for an intelligent evader: Strategies for searcher and evader in the two-region problem. *Naval Research Logistics Quarterly*, 25(1):95–106, 1978.

[125] William H Ruckle. A discrete search game. In *Stochastic Games And Related Topics*, pages 29–43. Springer, 1991.

[126] Daniel Seaberg, Laura Devine, and Jun Zhuang. A review of game theory applications in natural disaster management research. *Natural Hazards*, 89(3):1461–1483, 2017.

[127] Edward C Sewell and Sheldon H Jacobson. A branch, bound, and remember algorithm for the simple assembly line balancing problem. *INFORMS Journal on Computing*, 24(3):433–442, 2012.

[128] Edward C Sewell, Jason J Sauppe, David R Morrison, Sheldon H Jacobson, and Gio K Kao. A bb&r algorithm for minimizing total tardiness on a single machine with sequence dependent setup times. *Journal of Global Optimization*, 54(4):791–812, 2012.

[129] Xiaojun Shan and Jun Zhuang. Cost of equity in homeland security resource allocation in the face of a strategic attacker. *Risk Analysis*, 33(6):1083–1099, 2013.

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

[131] Xiaojun Shan and Jun Zhuang. Modeling credible retaliation threats in deterring the smuggling of nuclear weapons using partial inspection—a three-stage game. *Decision Analysis*, 11(1):43–62, 2014.

[132] Xiaojun Shan and Jun Zhuang. Modeling cumulative defensive resource allocation against a strategic attacker in a multi-period multi-target sequential game. *Reliability Engineering & System Safety*, In Press, 2017.

[133] Ariela Sharlin. Optimal search for one of many objects hidden in two boxes. *European Journal of Operational Research*, 32(2):251–259, 1987.

[134] Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. Protect: A deployed game theoretic system to protect the ports of the

united states. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 13–20. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[135] Wayne E Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.

[136] Veronica Strandberg. Rail bound traffic—a prime target for contemporary terrorist attacks? *Journal of Transportation Security*, 6(3):271–286, 2013.

[137] Jason Tsai, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Shyamsunder Rathi. Iris-a tool for strategic security allocation in transportation networks. In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (Industry Track), May 2009.*, 2009.

[138] Chen Wang and Vicki M Bier. Target-hardening decisions based on uncertain multiattribute terrorist utility. *Decision Analysis*, 8(4):286–302, 2011.

[139] Henry H Willis, Andrew R Morral, Terrence K Kelly, and Jamison Jo Medby. *Estimating terrorism risk.* Rand Corporation, 2006.

[140] Jie Xu and Jun Zhuang. Modeling costly learning and counter-learning in a defender-attacker game with private defender information. *Annals of Operations Research*, 236(1):271–289, 2016.

[141] Shota Yasutake, Kohei Hatano, Shuji Kijima, Eiji Takimoto, and Masayuki Takeda. Online linear optimization over permutations. In *International Symposium on Algorithms and Computation*, pages 534–543. Springer, 2011.

[142] Abdolmajid Yolmeh and Melike Baykal-Gürsoy. A robust approach to infrastructure security games. *Computers & Industrial Engineering*, 110:515–526, 2017.

[143] Abdolmajid Yolmeh and Melike Baykal-Gürsoy. Urban rail patrolling: a game theoretic approach. *Journal of Transportation Security*, 11(1-2):23–40, 2018.

[144] Abdolmajid Yolmeh and Melike Baykal-Gürsoy. Two-stage invest–defend game: Balancing strategic and operational decisions. *Decision Analysis*, 16(1):46–66, 2019.

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

[146] Jun Zhuang, Vicki M Bier, and Oguzhan Alagoz. Modeling secrecy and deception in a multiple-period attacker–defender signaling game. *European Journal of Operational Research*, 203(2):409–418, 2010.

[147] Noemí Zoroa, María-José Fernández-Sáez, and Procopio Zoroa. Tools to manage search games on lattices. In *Search Theory*, pages 29–58. Springer, 2013.

[148] Noemí Zoroa, Procopio Zoroa, and M José Fernández-Sáez. Weighted search games. *European Journal of Operational Research*, 195(2):394–411, 2009.

Appendices

Appendix A

Proofs of Selected Theorems and Lemmas

A.1 Proof of Theorem 2.3

Proof. The proof follows similar steps as the one in [35]. By definition of Nash Equilibrium, $(x, (\mathbf{y^1}, \mathbf{y^2}))$ is an equilibrium if and only if for some v^1, v^2 and v:

$$(\overline{\delta}_i x_i - 1)\underline{C}_i \begin{cases} = v^1, & y_i^1 > 0, \\ > v^1, & y_i^1 = 0, \end{cases} \tag{A.1}$$

$$(\overline{\delta}_i x_i)\underline{C} \begin{cases} = v^2, & y_i^2 > 0, \\ > v^2, & y_i^2 = 0, \end{cases} \tag{A.2}$$

$$q\left(\underline{\delta}_i \overline{C}_i y_i^1 - \sum_{j=1}^{N} \overline{C}_j y_j^1\right) + (1-q)\left(\underline{\delta}_i \overline{C}_i y_i^2 - \sum_{j=1}^{N} \overline{C}_j y_j^2\right) \begin{cases} = v, & x_i > 0, \\ < v, & x_i = 0. \end{cases} \tag{A.3}$$

These conditions imply that $0 \le v^2 \le \underline{C}$ and $v^1 \le 0$. There are two possible cases: (A) $x_i > 0$ for every i. (B) $x_i = 0$ for some i. As will be shown shortly, these two cases are equivalent to the cases $m \leq k$ and $m > k$, respectively.

Case (A): $x_i > 0$ for all i

Then, Eq. (A.2) indicates that $v^2 > 0$. Moreover, from Eq. (A.3), for every i, one can deduce that only three cases are possible:

Case A1: $y_i^1 > 0, y_i^2 > 0$

By equations (A.1) and (A.2), we have that $(\overline{\delta}_i x_i - 1)\underline{C}_i = v^1$, and $\overline{\delta}_i x_i \underline{C} = v^2$. Hence

$$C_i = -\frac{v^1}{1 - \frac{v^2}{\underline{C}}}, \quad \mathit{and} \quad x_i = \frac{v^2}{\overline{\delta}_i \underline{C}} = \frac{1}{\overline{\delta}_i}\left(\frac{v^1}{\underline{C}_i} + 1\right), \quad \mathit{for} \quad y_i^1,\ y_i^2 > 0. \tag{A.4}$$

Case A2: $y_i^1 > 0, y_i^2 = 0$

Then, the appropriate equality and inequality from (A.1) and (A.2) become $(\overline{\delta}_i x_i - 1)\underline{C}_i = v^1$ and $\overline{\delta}_i x_i \underline{C} > v^2$, respectively, thus giving

$$\underline{C}_i > -\frac{v^1}{1 - \frac{v^2}{\underline{C}}}, \quad \mathit{and} \quad x_i = \frac{1}{\overline{\delta}_i}\left(\frac{v^1}{\underline{C}_i} + 1\right) \quad \mathit{for} \quad y_i^1 > 0, y_i^2 = 0. \tag{A.5}$$

Case A3: $y_i^1 = 0, y_i^2 > 0$

The appropriate inequality and equality from (A.1) and (A.2) are $(\overline{\delta}_i x_i - 1)\underline{C}_i \geq v^1$ and $\overline{\delta}_i x_i \underline{C} = v^2$, respectively, therefore

$$\underline{C}_i < -\frac{v^1}{1 - \frac{v^2}{\underline{C}}}, \quad \mathit{and} \quad x_i = \frac{v^2}{\overline{\delta}_i \underline{C}} \qquad y_i^1 = 0, y_i^2 > 0. \tag{A.6}$$

Now, since $\underline{C}_i$ values are sorted it is obvious that there exists an m such that

$$\underline{C}_m = -\frac{v^1}{1 - \frac{v^2}{\underline{C}}},$$

$$y_i^1 \begin{cases} > 0, & i \leq m - 1, \\ \geq 0, & i = m, \\ = 0, & i \geq m + 1, \end{cases} \tag{A.7}$$

$$y_i^2 \begin{cases} = 0, & i \leq m-1, \\ \geq 0, & i = m, \\ > 0, & i \geq m+1, \end{cases} \tag{A.8}$$

and:

$$x_i = \begin{cases} \frac{1}{\overline{\overline{\delta}}_i}\left(\frac{v^1}{\underline{C}_i}+1\right), & i \leq m-1, \\ \frac{v^2}{\overline{\overline{d}}_m \underline{C}} = \frac{1}{\overline{\overline{\delta}}_m}\left(\frac{v^1}{\underline{C}_m}+1\right), & i = m, \\ \frac{v^2}{\overline{\overline{\delta}}_i \underline{C}}, & i \geq m+1. \end{cases} \tag{A.9}$$

In turn, equations (A.3), (A.7) and (A.8) imply:

$$y_i^1 = \begin{cases} \frac{v+(1-q)\sum_{j=1}^N \overline{C}_j y_j^2 + q\sum_{j=1}^N \overline{C}_j y_j^1}{\underline{\delta}_i \overline{C}_i q}, & i \leq m-1, \\ y_m^1, & i = m, \\ 0, & i \geq m+1, \end{cases} \tag{A.10}$$

$$y_i^2 = \begin{cases} 0, & i \leq m-1, \\ y_m^2, & i = m, \\ \frac{v+(1-q)\sum_{j=1}^N \overline{C}_j y_j^2 + q\sum_{j=1}^N \overline{C}_j y_j^1}{\underline{\delta}_i \overline{C}_i (1-q)}, & i \geq m+1, \end{cases} \tag{A.11}$$

and:

$$q y_m^1 + (1-q) y_m^2 = \frac{1}{\underline{\delta}_m \overline{C}_m}\left(v + (1-q)\sum_{j=1}^N \overline{C}_j y_j^2 + q\sum_{j=1}^N \overline{C}_j y_j^1\right). \tag{A.12}$$

From equations (A.10) and (A.11) it can be easily seen that the right hand side of (A.12) can be written as

$$q y_m^1 + (1-q) y_m^2 = \frac{\underline{\delta}_i \overline{C}_i q y_i^1}{\underline{\delta}_m \overline{C}_m}, \quad \forall i \leq m-1, \tag{A.13}$$

$$qy^1_m + (1-q)y^2_m = \frac{\underline{\delta}_i\overline{C}_i(1-q)y^2_i}{\underline{\delta}_m\overline{C}_m}, \quad \forall i \geq m+1. \tag{A.14}$$

Because $\mathbf{y^1}$ and $\mathbf{y^2}$ are probability vectors, we have

$$\sum_{j=1}^{N} y^1_i = 1, \quad \textit{and} \quad \sum_{j=1}^{N} y^2_i = 1.$$

Hence, summing equations (A.13) for $1 \leq i \leq m-1$ yields $1 - y^1_m$, while summing equations (A.14) for $m+1 \leq i \leq N$ yields $1 - y^2_m$ and both provide the following equalities respectively.

$$qy^1_m \sum_{j=1}^{m} \frac{1}{\underline{\delta}_j\overline{C}_j} + (1-q)y^2_m \sum_{j=1}^{m-1} \frac{1}{\underline{\delta}_j\overline{C}_j} = \frac{q}{\underline{\delta}_m\overline{C}_m},$$

$$qy^1_m \sum_{j=m+1}^{N} \frac{1}{\underline{\delta}_j\overline{C}_j} + (1-q)y^2_m \sum_{j=m}^{N} \frac{1}{\underline{\delta}_j\overline{C}_j} = \frac{1-q}{\underline{\delta}_m\overline{C}_m}.$$

Finally, summing the above equations gives

$$qy^1_m + (1-q)y^2_m = \frac{1}{\underline{\delta}_m\overline{C}_m \sum_{j=1}^{N} \frac{1}{\underline{\delta}_j\overline{C}_j}},$$

in turn this leads to the unique solution in equations (2.6) and (2.7).

In order to compute m note that $\mathbf{y^1}$ and $\mathbf{y^2}$ are probability vectors, thus $y^2_m \geq 0, y^1_m \geq 0$ in equations (2.6) and (2.7), implying that $q \leq \psi_m$ and $q \geq \psi_{m-1}$, respectively.

The defender's strategy can be obtained from (A.1) and (A.2) that indicate

$$x_i = \frac{v^1 + \underline{C}_i}{\underline{C}_i\overline{\delta}_i}, \quad i \leq m, \tag{A.15}$$

$$x_i = \frac{v^2}{\underline{C}\overline{\delta}_i}, \quad i \geq m, \tag{A.16}$$

together with the normalization equation, $\sum_{i=1}^{N} x_i = 1$, yielding the equation (2.5).

To show that $m \leq k$, it is enough to show $\phi_m \leq 1$. From equation (A.15), we have $x_m = \frac{v^1 + \underline{C}_m}{\underline{C}_m\overline{\delta}_m}$, which leads to $v^1 \geq -\underline{C}_m$. Using this inequality, we have:

$$\phi_m = \sum_{j=1}^{m} \frac{\underline{C}_j - \underline{C}_m}{\overline{\delta}_j\underline{C}_j} \leq \sum_{j=1}^{m} \frac{\underline{C}_j + v^1}{\overline{\delta}_j\underline{C}_j} = \sum_{j=1}^{m} x_j \leq 1.$$

Case B:

Suppose there exists an i such that $x_i = 0$. Then by (A.2) $v^2 = 0$, therefore $y_i^2 = 0$ for $x_i > 0$. From equations (A.1) to (A.3), for every i, one can deduce that only three cases are possible:

Case B1: $y_i^1 > 0, y_i^2 > 0$

By equations (A.1) and (A.2), we have that $(\overline{\delta}_i x_i - 1)\underline{C}_i = v^1$, and $\overline{\delta}_i x_i \underline{C} = v^2$. Hence

$$C_i = -\frac{v^1}{1 - \frac{v^2}{\underline{C}}}, \quad and \quad x_i = \frac{v^2}{\overline{\delta}_i \underline{C}} = \frac{1}{\overline{\delta}_i}\left(\frac{v^1}{\underline{C}_i} + 1\right) = 0, \quad for \quad y_i^1,\ y_i^2 > 0. \qquad \text{(A.17)}$$

Case B2: $y_i^1 > 0, y_i^2 = 0$

Then, the appropriate equality and inequality from (A.1) and (A.2) become $(\overline{\delta}_i x_i - 1)\underline{C}_i = v^1$ and $\overline{\delta}_i x_i \underline{C} \geq v^2$, respectively, thus giving

$$\underline{C}_i > -\frac{v^1}{1 - \frac{v^2}{\underline{C}}}, \quad and \quad x_i = \frac{1}{\overline{\delta}_i}\left(\frac{v^1}{\underline{C}_i} + 1\right) \quad for \quad y_i^1 > 0, y_i^2 = 0. \qquad \text{(A.18)}$$

Case B3: $y_i^1 = 0, y_i^2 > 0$

The appropriate inequality and equality from (A.1) and (A.2) are $(\overline{\delta}_i x_i - 1)\underline{C}_i \geq v^1$ and $\overline{\delta}_i x_i \underline{C} = v^2$, respectively, therefore

$$\underline{C}_i < -\frac{v^1}{1 - \frac{v^2}{\underline{C}}}, \quad and \quad x_i = \frac{v^2}{\overline{\delta}_i \underline{C}} = 0 \qquad y_i^1 = 0, y_i^2 > 0. \qquad \text{(A.19)}$$

Now, since $\underline{C}_i$ values are sorted , there exists a k such that

$$x_i \begin{cases} \geq 0, & i \leq k, \\ = 0 & i > k, \end{cases} \qquad \text{(A.20)}$$

$$y_i^1 \begin{cases} \geq 0, & i \leq k, \\ = 0 & i > k, \end{cases} \qquad \text{(A.21)}$$

$$y_i^2 \begin{cases} = 0, & i \leq k, \\ \\ \geq 0 & i > k. \end{cases} \tag{A.22}$$

Moreover, by equation (A.1), we have:

$$x_i = \begin{cases} \frac{1}{\overline{\delta}_i} \left(\frac{v^1}{\underline{C}_i} + 1 \right), & y_i^1 > 0, \\ \\ 0, & y_i^1 = 0. \end{cases} \tag{A.23}$$

In turn, equations (A.3), (A.21), and (A.22) imply

$$y_i^1 = \begin{cases} \frac{v + (1-q) \sum_{j=1}^N \overline{C}_j y_j^2 + q \sum_{j=1}^N \overline{C}_j y_j^1}{\underline{\delta}_i \overline{C}_i q}, & i \leq k, \\ \\ 0, & i > k, \end{cases} \tag{A.24}$$

$$y_i^2 \begin{cases} = 0, & i \leq k, \\ \\ \leq \frac{v + q \sum_{j=1}^N \overline{C}_j y_j^1 + (1-q) \sum_{j=1}^N \overline{C}_j y_j^2}{\underline{\delta}_i \overline{C}_i (1-q)}, & i > k. \end{cases} \tag{A.25}$$

Moreover, we have:

$$\sum_{j=1}^{k} y_i^1 = 1,$$

$$\sum_{j=1}^{k} x_i = 1.$$

Solving these equations leads to the solution characterized in equations (2.8) to (2.10). To compute k, we have $x_k \geq 0$, this leads to $\phi_k \leq 1$. To show that $m > k$ it is enough to show $\psi_k < q$. From equation (2.10) we have:

$$y_i^{*2} < \frac{q}{(1-q)} \left(\frac{\frac{1}{\underline{\delta}_i \overline{C}_i}}{\sum\limits_{l=1}^{k} \frac{1}{\underline{\delta}_l \overline{C}_l}}, \right) \forall i > k, \quad \sum_{j=k+1}^{N} y_j^{*2} = 1.$$

Using these equations, we have:

$$1 = \sum_{j=k+1}^{N} y_j^{*2} < \frac{q}{1-q} \left(\frac{\sum_{j=k+1}^{N} \frac{1}{\underline{\delta}_j \overline{C}_j}}{\sum_{j=1}^{k} \frac{1}{\underline{\delta}_j \overline{C}_j}} \right).$$

Which leads to $\psi_k = \frac{\sum_{j=1}^{k} \frac{1}{\delta_j \overline{C}_j}}{\sum_{j=1}^{N} \frac{1}{\delta_j \overline{C}_j}} < q$. This completes the proof. □

A.2 Theorem 3.1

Proof. To prove this theorem we first establish some lemmas.

Lemma A.1. *The second stage matrix game has a pure Nash Equilibrium if and only if* $(1-\delta_1) C_1 - \delta_1 P \geq C_2$.

Proof. Suppose we have $(1-\delta_1) C_1 - \delta_1 P \geq C_2$. It is easy to check that $\mathbf{x} = (1, 0, 0, ..., 0)$, $\mathbf{y} = (1, 0, 0, ..., 0)$ is a pure Nash Equilibrium strategy pair. This establishes the sufficiency part. We prove the necessity part by contradiction, suppose that $(1-\delta_1) C_1 - \delta_1 P < C_2$. and the game has a pure Nash Equilibrium, this pure Nash Equilibrium is definitely not $\mathbf{x} = (1, 0, 0, ..., 0)$, $\mathbf{y} = (1, 0, 0, ..., 0)$, because at this strategy profile the adversary can strictly increase his payoff by attacking site 2. Moreover it has to be on the diagonal of the matrix i.e. $x_i = y_i = 1$ for some $i > 1$ however, this implies that $(1-\delta_i) C_i - \delta_i P \geq C_1$ which contradicts our assumption of sorted C_is, thus proving the necessity part. □

Lemma A.2 characterizes the conditions under which some strategies of the adversary are dominated by a linear combination of other strategies. This lemma helps us find a critical index to compute the Nash Equilibrium.

Lemma A.2. *If* $\sum\limits_{j=1}^{k} \frac{C_j - C_k}{\delta_j (C_j + P)} > 1$, *then the adversary's strategies* $l \geq k$ *are strictly dominated by a mixed strategy that is composed of pure strategies* j *for* $1 \leq j < k$, *i.e., there exist* $\lambda_i \geq 0$, $1 \leq i \leq k-1$ *with* $\sum\limits_{i=1}^{k-1} \lambda_i = 1$ *such that:*

$$\lambda_1 \begin{bmatrix} (1-\delta_1)\, C_1 - \delta_1 P \\ C_1 \\ C_1 \\ \vdots \\ C_1 \end{bmatrix} + \cdots + \lambda_{k-1} \begin{bmatrix} C_{k-1} \\ \vdots \\ (1-\delta_{k-1})\, C_{k-1} - \delta_{k-1} P \\ \vdots \\ C_{k-1} \end{bmatrix} > \begin{bmatrix} C_l \\ \vdots \\ \vdots \\ (1-\delta_l)\, C_l - \delta_l P \\ \vdots \\ C_l \end{bmatrix}.$$

Proof. The inequality holds for rows $r \geq k$ because C_is are sorted, i.e., $\sum_{j=1}^{k-1} \lambda_j C_j > C_k$ for all $\lambda_i \geq 0$, $1 \leq i \leq k-1$ with $\sum\limits_{i=1}^{k-1} \lambda_i = 1$

For rows $r < k$, consider the assumption $\sum\limits_{j=1}^{k} \frac{C_j - C_k}{\delta_j (C_j + P)} > 1$ After some algebraic manipulations this inequality can be rewritten as:

$$\frac{(1-\delta_r)\, C_r - \delta_r P}{\delta_r (C_r + P) \sum\limits_{m=1}^{k-1} \frac{1}{\delta_m (C_m + P)}} + \sum_{j=1, j \neq r}^{k-1} \frac{C_j}{\delta_j (C_j + P) \sum\limits_{m=1}^{k-1} \frac{1}{\delta_m (C_m + P)}} > C_k.$$

Setting $\lambda_j = \frac{1}{\delta_j (C_j + P) \sum\limits_{m=1}^{k-1} \frac{1}{\delta_m (C_m + P)}}$ gives the result as:

$$\lambda_r \,(1-\delta_r)\, C_r + \sum_{j=1, j \neq r}^{k-1} \lambda_j C_j > C_k \geq C_l.$$

□

Lemma A.3 complements Lemma A.2 in characterizing the sites that should be in the mixed Nash Equilibrium.

Lemma A.3. *If* $\sum\limits_{j=1}^{k} \frac{C_j - C_k}{\delta_j (C_j + P)} < 1$, *any strategy profile with* $x_k = 0$ *is not a Nash Equilibrium.*

Proof. By contradiction. Suppose the Nash Equilibrium holds with $x_k = 0$. If $y_k = 0$, consider a critical $k^* \geq k$ such that $\sum\limits_{j=1}^{k^*} \frac{C_j - C_k^*}{\delta_j (C_j + P)} < 1 < \sum\limits_{j=1}^{k^*+1} \frac{C_j - C_{k^*+1}}{\delta_j (C_j + P)}$. Using Lemma

A.1 we can conclude that both players are playing a mixed strategy. Moreover using Lemma A.2 we have: $x_j = 0, y_j = 0, \forall j > k^*$. Therefore the adversary is indifferent towards his choices $i = 1, ..., k^*, i \neq k$, in other words:

$$(1-\delta_1 x_1) C_1 - \delta_1 x_1 P = \cdots = (1-\delta_{k-1} x_{k-1}) C_{k-1} - \delta_{k-1} x_{k-1} P =$$
$$(1-\delta_{k+1} x_{k+1}) C_{k+1} - \delta_{k+1} x_{k+1} P = \cdots = (1-\delta_{k^*} x_{k^*}) C_{k^*} - \delta_{k^*} x_{k^*} P.$$

Solving these equations along with the equation $\sum\limits_{j=1, j\neq k}^{k^*} x_j = 1$ yields:

$$x_{k^*} = \frac{1 - \sum\limits_{j=1,j\neq k}^{k^*} \frac{C_j - C_{k^*}}{\delta_j (C_j+P)}}{\delta_{k^*}(C_{k^*}+P) \sum\limits_{j=1,j\neq k}^{k^*} \frac{1}{\delta_j (C_j+P)}}.$$

Since $\sum\limits_{j=1}^{k^*} \frac{C_j - C_{k^*}}{\delta_j (C_j+P)} < 1$ and $C_{k^*} \leq C_k$, the following inequality holds

$$\sum_{j=1,j\neq k}^{k^*} \frac{C_j - C_k}{\delta_j (C_j+P)} < 1,$$

which could be rewritten as:

$$\sum_{j=1,j\neq k}^{k^*} \frac{C_j - C_{k^*} + (C_{k^*} - C_k)}{\delta_j (C_j+P)} < 1.$$

This further simplifies to:

$$(C_{k^*} - C_k) < \frac{1 - \sum\limits_{j=1,j\neq k}^{k^*} \frac{C_j - C_{k^*}}{\delta_j (C_j+P)}}{\sum\limits_{j=1,j\neq k}^{k^*} \frac{1}{\delta_j (C_j+P)}} = \delta_{k^*}(C_{k^*}+P) x_{k^*},$$

giving $(1-\delta_{k^*} x_{k^*}) C_{k^*} - \delta_{k^*} x_{k^*} P < C_k$. Therefore the adversary can strictly improve his payoff by increasing y_k to 1. Hence $y_k = 1$ should hold. Now the defender can strictly increase her payoff by increasing x_k to 1. This is in contradiction with our assumption of $x_k = 0$ being in a Nash Equilibrium. □

We are now ready to prove the theorem. Consider a critical k^* such that $\sum\limits_{j=1}^{k^*} \frac{C_j - C_{k^*}}{\delta_j (C_j+P)} < 1 < \sum\limits_{j=1}^{k^*+1} \frac{C_j - C_{k^*+1}}{\delta_j (C_j+P)}$, if $k^* = 1$ then Lemma A.1 and Lemma A.2 imply that the game has a unique pure strategy Nash Equilibrium. If $k^* \geq 2$, then using lemma A.2 and lemma A.3, the mixed strategy Nash Equilibrium is determined by solving the following systems of equations:

System 1:

$$(1-\delta_1 x_1)\,C_1 - \delta_1 x_1 P = (1-\delta_2 x_2)\,C_2 - \delta_2 x_2 P = ... = (1-\delta_{k^*} x_{k^*})\,C_{k^*} - \delta_{k^*} x_{k^*} P,$$

$$\sum_{j=1}^{k^*} x_j = 1.$$

System 2:

$$-\left(1-\delta_1\right)C_1 y_1 - \sum_{j=1, j\neq 1}^{k^*} C_j y_j = -\left(1-\delta_2\right)C_2 y_2 - \sum_{j=1, j\neq 2}^{k^*} C_j y_j = ... = -\left(1-\delta_{k^*}\right)C_{k^*} y_{k^*} - \sum_{j=1, j\neq k^*}^{k^*} C_j y_j,$$

$$\sum_{j=1}^{k^*} y_i = 1.$$

Both systems have unique solutions. Solving these systems lead to the solution in equations (3.1) to (3.4). □

A.3 Lemma 3.1

Proof. Because the first stage payoffs, $u_1^d(\boldsymbol{\alpha})$ and $u_1^a(\boldsymbol{\beta})$, are linear in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, they are continuous in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Therefore, in order to prove that the total payoff functions are continuous, we only need to prove that the second stage payoffs are continuous. We first prove that the second stage payoff function for the defender is continuous in both players' strategies. Here is the payoff function:

$$u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}}.$$

For a fixed value of k, clearly the payoff function is continuous, therefore we only need to prove that it is also continuous when the value of k changes. If $C_i = C, \ \forall i =$

$1, \ldots, N$, then $k = N$ always holds and the result follows. We now focus on the case of $P = 0$. The value of k changes only when $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{k-1} \dfrac{C_j - C_k}{\delta_j(\alpha_j, \beta_j)(C_j)} = 1$ and a small change in either α or β results in $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) > 1$, hence causing the value of k decreased by one unit. At this point the expected damage is computed using the formula $k' = k - 1$ as the threshold value. We prove that the expected damage under both threshold indices k,or $k - 1$, lead to the same value:

$$\left| u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)|_{k'} - u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)|_k \right| = \left| \frac{1 - \sum_{j=1}^{k-1} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k-1} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}} - \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}} \right|$$

$$= \left| \frac{1}{\delta_k(\alpha_k, \beta_k) C_k} \frac{1 - \sum_{j=1}^{k-1} \dfrac{C_j - C_k}{\delta_j(\alpha_j, \beta_j) C_j}}{\sum_{j=1}^{k-1} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j} \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}} \right| = 0.$$

This establishes continuity of the defender's payoff. Same argument applies when proving continuity of the adversary's payoff function. □

A.4 Lemma 3.2

Proof. We have already established continuity of the payoff functions. In order to prove concavity we show that the second derivative is negative. The first derivative of $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$ is given as:

$$\frac{\partial u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})}{\partial \alpha_i} = -1 + \left(\frac{\dfrac{1}{\delta_i^2(\alpha_i, \beta_i) C_i} \left(1 - \sum_{j=1}^{k} \dfrac{C_j - C_i}{\delta_j(\alpha_j, \beta_j) C_j} \right)}{\left(\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j} \right)^2} \right) \frac{\partial \delta_i(\alpha_i, \beta_i)}{\partial \alpha_i}.$$

Then the second derivative satisfies:

$$\frac{\partial^2 u_t^d(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})}{\partial\alpha_i^2} =$$

$$\left(\frac{2}{\delta_i^3(\alpha_i,\beta_i)C_i}\left(1-\sum_{j=1}^{k}\frac{C_j-C_i}{\delta_j(\alpha_j,\beta_j)C_j}\right)\right)\frac{\frac{1}{\delta_i(\alpha_i,\beta_i)C_i}-\sum_{j=1}^{k}\frac{1}{\delta_j(\alpha_j,\beta_j)C_j}}{\sum_{j=1}^{k}\frac{1}{(\delta_j(\alpha_j,\beta_j)C_j)^3}}\ \frac{\partial\delta_i(\alpha_i,\beta_i)}{\partial\alpha_i}$$

$$+\frac{\frac{1}{\delta_i^2(\alpha_i,\beta_i)C_i}\left(1-\sum_{j=1}^{k}\frac{(C_j-C_i)}{(\delta_j(\alpha_j,\beta_j)C_j)}\right)}{(\sum_{j=1}^{k}\frac{1}{\delta_j(\alpha_j,\beta_j)C_j})^2}\ \frac{\partial^2\delta_i(\alpha_i,\beta_i)}{\partial\alpha_i^2} < 0.$$

The last inequality is valid because from the assumptions either $P = 0$ or $C_i = C, \ \forall i = 1, \ldots, N$, holds. Therefore $u_t^d(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})$ is strictly concave in α_is. One can similarly show that $u_t^a(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})$ is strictly concave in β_is. □

A.5 Lemma 3.3

Proof. To prove the lemma for $u_t^d(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x}^*,\boldsymbol{y}^*)$, we first prove that $u_2^d(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x}^*,\boldsymbol{y}^*)$ is concave in $(\delta_1(\alpha_1,\beta_1),\delta_2(\alpha_2,\beta_2),...,\delta_N(\alpha_N,\beta_N))$. Here is the Hessian matrix for

$u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$:

$$H = \frac{2C}{\left(\sum_{j=1}^N \frac{1}{\delta_j}\right)^3} \begin{bmatrix} \frac{\left(\frac{1}{\delta_1} - \sum_{j=1}^N \frac{1}{\delta_j}\right)}{\delta_1^3} & \frac{1}{\delta_1^2\delta_2^2} & \cdots & \frac{1}{\delta_1^2\delta_N^2} \\ \frac{1}{\delta_2^2\delta_1^2} & \frac{\left(\frac{1}{\delta_2} - \sum_{j=1}^N \frac{1}{\delta_j}\right)}{\delta_2^3} & \cdots & \frac{1}{\delta_2^2\delta_N^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\delta_N^2\delta_1^2} & \frac{1}{\delta_N^2\delta_2^2} & \cdots & \frac{\left(\frac{1}{\delta_N} - \sum_{j=1}^N \frac{1}{\delta_j}\right)}{\delta_N^3} \end{bmatrix}.$$

Let H_l be the submatrix of H obtained by taking the upper left hand corner $l \times l$ matrix of H. Furthermore let $|H_l|$, be the lth principal minor of H.

We need to show that the principal minors of H alternate in sign, starting with negative i.e. $(-1)^l |H_l| > 0$ for $l = 1, 2, \ldots, N-1$ and $|H| = 0$. Because we are only concerned about sign of the determinant of H, we can divide (or multiply) rows and columns of H with positive quantities. Therefore, we divide row i by $\frac{2C}{\delta_i\left(\sum_{j=1}^N \frac{1}{\delta_j}\right)^3}$ for $i = 1, 2, \ldots, N$, then we multiply column i by δ_i for $i = 1, 2, \ldots, N$. Here is the resulting matrix:

$$H' = \begin{bmatrix} \frac{1}{\delta_1}\left(\frac{1}{\delta_1} - \sum_{j=1}^N \frac{1}{\delta_j}\right) & \frac{1}{\delta_1\delta_2} & \cdots & \frac{1}{\delta_1\delta_N} \\ \frac{1}{\delta_2\delta_1} & \frac{1}{\delta_2}\left(\frac{1}{\delta_2} - \sum_{j=1}^N \frac{1}{\delta_j}\right) & \cdots & \frac{1}{\delta_2\delta_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\delta_N\delta_1} & \frac{1}{\delta_N\delta_2} & \cdots & \frac{1}{\delta_N}\left(\frac{1}{\delta_N} - \sum_{j=1}^N \frac{1}{\delta_j}\right) \end{bmatrix}.$$

H' is a symmetric diagonally dominant matrix because absolute value of each element on the diagonal is equal to sum of absolute values of all other elements in the same row. Therefore H' is a negative semi-definite matrix. Hence the leading principal minors of H' alternate in sign, starting with negative i.e. $(-1)^l\,|H'_l| > 0$ for $l = 1, 2, \ldots, N-1$ and $|H'| = 0$. Because we obtained H' by multiplying rows and columns of H with positive quantities, H' and H have the same determinant sign, this is also true for their leading principal minor signs. Therefore the leading principal minors of H alternate in sign, starting with negative i.e. $(-1)^l\,|H_l| > 0$ for $l = 1, 2, \ldots, N-1$ and $|H| = 0$. Hence $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ is concave in $(\delta_1(\alpha_1, \beta_1), \delta_2(\alpha_2, \beta_2), ..., \delta_N(\alpha_N, \beta_N))$. It follows that $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ is concave in $\boldsymbol{\alpha}$, because increasing concave function of a concave function is concave. It then follows that $u_t^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ is concave in $\boldsymbol{\alpha}$, because sum of two concave functions is concave. One can similarly prove the lemma for $u_t^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y})$. □

A.6 Lemma 3.4

Proof. To prove this lemma for $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$, we prove that all of its upper level sets are convex. Suppose for two points $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}'$ and some L we have:

$$u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha_j, \beta_j) C_j}} \geq L, \tag{A.26}$$

and

$$u_2^d(\boldsymbol{\alpha}', \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) = \frac{1 - \sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha'_j, \beta_j)}}{\sum_{j=1}^{k} \dfrac{1}{\delta_j(\alpha'_j, \beta_j) C_j}} \geq L. \tag{A.27}$$

We prove that for all λ with $0 \leq \lambda \leq 1$ we have:

$$u_2^d(\lambda\boldsymbol{\alpha} + (1-\lambda)\boldsymbol{\alpha}', \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) \geq L.$$

To prove this, from equation (A.26) we have:

$$\sum_{j=1}^{k} \frac{1 + \dfrac{L}{C_j}}{\delta_j(\alpha_j, \beta_j)} \leq 1.$$

Similarly from equation (A.27) we have:

$$\sum_{j=1}^{k} \frac{1 + \dfrac{L}{C_j}}{\delta_j(\alpha'_j, \beta_j)} \leq 1.$$

From these two equations we have:

$$\sum_{j=1}^{k}(1 + \frac{L}{C_j})(\frac{\lambda}{\delta_j(\alpha_j, \beta_j)} + \frac{1-\lambda}{\delta_j(\alpha'_j, \beta_j)}) \leq 1.$$

Now $u_2^d(\lambda\boldsymbol{\alpha} + (1-\lambda)\boldsymbol{\alpha}', \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*) \geq L$ follows from the convexity of $\frac{1}{\delta_j(\alpha_j,\beta_j)}$ for all j. Proof of quasi-concavity for $u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$ is similar to $u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}^*, \boldsymbol{y}^*)$. □

A.7 Theorem 3.3

Proof. We fix the critical index k and write down the optimization problem for both players. For the defender we have the following optimization problem:

$$max \quad u_2^d(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{A.28}$$

$$\sum_{j=1}^{k} \alpha_j = A, \tag{A.29}$$

$$\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq 1, \tag{A.30}$$

$$\alpha_j \geq 0, \tag{A.31}$$

where $\phi_i(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{i} \frac{C_j - C_i}{\delta_j(\alpha_j, \beta_j)(C_j + P)}$ for $i \in 1, \ldots, N$ and $\phi_{N+1}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \infty$. It is easy to see that constraints (A.29) and (A.31) lead to convex strategy space for the defender. We show that the strategy space characterized by constraint (A.30) is also convex and therefore the whole strategy space is convex (because intersection of convex sets is convex). Consider two points $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha'}$ with $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq 1$, and $\phi_k(\boldsymbol{\alpha'}, \boldsymbol{\beta}) \leq 1$. We show that any convex combination of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha'}$ also satisfies constraint (A.30). Note that $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a convex function of $\boldsymbol{\alpha}$ because it is sum of convex functions. We have: $\phi_k(\lambda\boldsymbol{\alpha} + (1-\lambda)\boldsymbol{\alpha'}, \boldsymbol{\beta}) \leq \lambda\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) + (1-\lambda)\phi_k(\boldsymbol{\alpha'}, \boldsymbol{\beta}) \leq 1$. The first inequality comes from convexity of $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and the second inequality comes from assumption of $\phi_k(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq 1$, and $\phi_k(\boldsymbol{\alpha'}, \boldsymbol{\beta}) \leq 1$. Therefore defender's strategy space is convex. It is easy to check that the strategy space is also compact.

The optimization problem for the adversary is as follows:

$$max \quad u_2^a(\boldsymbol{\alpha}, \boldsymbol{\beta}) \tag{A.32}$$

$$\sum_{j=1}^{k} \beta_j = B, \tag{A.33}$$

$$\phi_{k+1}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \geq 1, \tag{A.34}$$

$$\beta_j \geq 0. \tag{A.35}$$

The strategy space for the defender is also convex and compact (the proof is similar to the convexity proof of defender's strategy space). Therefore the strategy space for both players are convex and compact. In lemma 3.4 we established that the payoff functions for both players are quasi-concave with respect to their own strategy.

Moreover, because we fixed the critical index k, the payoff functions are continuous. We also know that, because $\phi_i(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is increasing in i, for each $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ there exists an index k such that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is feasible. Therefore, applying Debreu's existence theorem (see [27]), there exist at least one Nash Equilibrium.

To establish uniqueness, we use the index theory approach (see Theorem 7 in [23]). Because first derivatives are all positive, there is no point with $\frac{\partial u_2^d(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})}{\partial \alpha_i} = 0$ and $\frac{\partial u_2^a(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{x},\boldsymbol{y})}{\partial \beta_i} = 0$ for $i = 1, 2, ..., N$, therefore conditions of this theorem are, vacuously, satisfied. Therefore there exists at most one Nash Equilibrium. □

A.8 Proof of Theorem 5.1

Proof. Our proof is similar to the NP-hardness proof of the DET-STRAT problem (Theorem 4.4) in [14]. We first prove the result for the case with $|\mathcal{S}| = |\mathcal{A}| = 1$. Once the NP-hardness of this case is established, NP-hardness of the case with multiple patrollers and attackers follows due to the added complexity. To prove NP-hardness, we prove that the corresponding decision problem is NP-complete. Here is the corresponding decision problem:

Given a patrolling game $G = G(Q, T, \mathbf{m}, \mathbf{c})$ and dual values $[y_{i\tau}]$ (note that we assumed there is only one attacker and $y_{i\tau}$ is the probability of using attack pair (i, τ) by this single attacker) is there a patroll with $[w_{i\tau}]$ such that:

$$\sum_{i=1}^{N} \sum_{\tau=0}^{T-m_i} y_{i\tau} C_{i,\tau+m_i-1} w_{i\tau} \geq L?$$

For future references, we call this problem DP. It is easy to see that DP is in NP. A non-deterministic machine could guess a patrol with $w_{i\tau}$ and check if $\sum_{i=1}^{N} \sum_{\tau=0}^{T} y_{i\tau} C_{i,\tau+m_i-1} w_{i\tau} \geq L$ is true, in polynomial time. Its NP-completeness can be shown by reducing the Hamiltonian path (HP) problem to DP. HP is a well-known NP-complete problem [40]. It is the problem of determining if a Hamiltonian path, i.e., a path that visits each vertex exactly once, exists on a given graph. Following [14] let us consider a generic HP problem given by a graph $G^h = (V^h, A^h)$ where V^h is the set of vertices and A^h is the set of edges. From this graph, an instance of the DP problem with $G = G(Q, T, \mathbf{m}, \mathbf{c})$, L and $[y_{i\tau}]$ can be constructed in polynomial time by setting $Q = G^h$, $T = |V^h|$, $m_i = |V^h|$, $y_{i\tau} = 1, \forall i, \tau$, $C_{i\tau} = 1, \forall i, \tau$ and $L = |V^h|$. It is easy to see that a solution to DP with $G = G(Q, T, \mathbf{m}, \mathbf{c})$, L and $[y_{i\tau}]$, if exists, is a Hamiltonian path. To achieve $L = |V^h|$, every node should be fully covered by the patrol i.e. all attack pairs (i, τ) should be caught by the patrol. Note that, for every node i, the only admissible attack start time is $\tau = 0$. Therefore, there are $|V^h|$ attack pairs and to achieve $L = |V^h|$, all of these attack pairs should be interrupted. To interrupt each attack pair (i, τ), it is enough to visit node i during time interval $[0, T - 1]$. Because the length of time horizon is equal to the number of nodes, the patrol should visit each node exactly once, because otherwise at least one node will not be covered. Therefore, computing a solution for DP with $G = G(Q, T, \mathbf{m}, \mathbf{c})$, L and $[y_{i\tau}]$ provides a solution for the HP problem with G^h. Thus the HP problem can be reduced to DP. This proves that DP is NP-complete. □