Detector deployment under a small vessel attack threat to port security

Yan, Xihong

2015

Yan, X. (2015). Detector deployment under a small vessel attack threat to port security.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65462>

<https://doi.org/10.32657/10356/65462>

DETECTOR DEPLOYMENT UNDER A SMALL VESSEL ATTACK THREAT TO PORT SECURITY

YAN XIHONG

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

2015

“ Research is what I’m doing when I don’t know what I’m doing.”

Wernher Von Braun

“ After all, the ultimate goal of all research is not objectivity, but truth.”

Helene Deutsch

Acknowledgements

I would like to express my deepest gratitude to Assistant Professor Nie Xiaofeng, my supervisor, for the patient guidance and encouragement he has given me. He not only provided a direction for my research, but also led me to deeply understand the corresponding knowledge. He often gave me useful suggestions which helped me clarify my thoughts.

My sincere gratitude goes to my friends, Mr. Yong Li and Dr. Yi Xu, for their support and help. They were always willing to give me their best suggestions.

I also owe a great debt of gratitude to my beloved family for being there for me.

Last but not least, heartfelt thanks are extended to all students and faculty members in the Division of Systems and Engineering Management for creating an excellent atmosphere for doing research.

Contents

Acknowledgements	i
List of Figures	iv
List of Tables	v
Abstract	vi
1 Introduction	1
1.1 Background	1
1.1.1 Importance of Ports	1
1.1.2 Maritime Threats	4
1.1.3 Response Measures to Maritime Threats	10
1.2 Research Scope	14
1.3 Organization of the Thesis	16
2 Literature Review	17
2.1 Survey on Port Security	18
2.1.1 Qualitative Studies on Port Security	19
2.1.2 Quantitative Studies on Port Security	21
2.2 Sensor Deployment for Surveillance	26
2.2.1 Stationary Sensor Location Models	26
2.2.2 Mobile Sensor Deployment Models	28
2.3 Game Theory	31
2.3.1 Application Models for Stackelberg Games	32
2.3.2 Network Interdiction	34
2.4 Relevant Algorithms and Techniques for Nonlinear Binary Integer Programming	38
2.4.1 Standard Branch and Bound Algorithm	38
2.4.2 Linearization Techniques	40
2.5 Literature Summary and Research Gaps	46
3 Optimal Placement of Multiple Types of Detectors under a Small Vessel Attack Threat to Port Security	48

3.1	Introduction	48
3.2	Modeling	51
3.3	Properties	57
3.4	Algorithms	60
3.4.1	Standard Branch and Bound Algorithm	61
3.4.2	Linearized Branch and Bound Algorithm	61
3.4.3	Greedy Adding Heuristic	66
3.5	Computational Study	68
3.5.1	Illustrative Example	68
3.5.2	Computational Results	71
4	Game-theoretic Detector Deployment under a Small Vessel At-	
	ack Threat to Port Security	77
4.1	Introduction	77
4.2	Problem Modeling	80
4.2.1	Problem Description	80
4.2.2	Problem Formulation	86
4.3	Algorithms	88
4.3.1	Optimal Detector Deployment for Fixed Attacking Plans	91
4.3.2	Optimal Attacking Plan for Fixed Detector Locations	95
4.3.3	Algorithms for Our Defender-Attacker Model	103
4.4	Computational Study	105
4.4.1	Illustrative Case	105
4.4.2	Performance of Algorithms	109
5	Summary and Future work	113
5.1	Summary	113
5.2	Future Work	117
A	Appendix for Chapter 3	120
	Bibliography	129

List of Figures

1.1	The number of piracy attacks worldwide from 2009 through 2014 [94].	5
1.2	A small vessel attack on USS Cole ¹ .	9
1.3	Limburg oil tanker fire caused by a small boat attack ² .	10
3.1	An example and the associated shortest paths.	52
3.2	An example for N_{kj}^h and detection length l_{hikj} .	54
3.3	A counter example.	58
3.4	A polyhedral outer approximation for $z_{kj} = \ln(y_{kj})$ over $[L_{kj}, U_{kj}]$ [87].	64
3.5	An illustrative example.	70
4.1	An illustrative example for a threat area and the associated network.	81
4.2	An example for N_{kj} and l_{ikj} .	83
4.3	A concerned bay in the Port of New York and New Jersey.	106
4.4	The optimal detector placement and attacking routes.	107
A.1	Possible patterns and the associated distances.	121
A.2	Possible structures for d_i .	122

List of Tables

3.1	Parameter values	69
3.2	Sensitivity analysis with respect to the budget limit	71
3.3	Computational results for small-size problems: 12×12 and 13×13	73
3.4	Computational results for small-size problems: 14×14 and 15×15	74
3.5	Computational results for middle-size problems: 20×20 and $25 \times$ 25	75
3.6	Computational results for middle-size problems: 30×30 and $35 \times$ 35	76
4.1	Parameters	85
4.2	Sensitivity analysis with respect to different detector numbers . . .	108
4.3	Sensitivity analysis with respect to different small boat numbers . .	108
4.4	Sensitivity analysis with respect to different target values	108
4.5	Parameter values	109
4.6	Computational results for problems with same target values	111
4.7	Computational results for problems with different target values . . .	112

Abstract

The 9/11 events have highlighted awareness about threats presented by terrorism. Ports and waterways play a crucial role in the global trade and homeland security. Because of its strategic importance and characteristics, a port is easily exposed to terrorist attacks. Providing an effective and efficient port security system has become one of the primary tasks for homeland security.

There are various types of terrorist attack threats in a port. In our research, we focus on scenarios where terrorists use small boats to attack maritime targets (e.g., military vessels, cruise ships, and oil tankers). We consider how to protect these targets by deploying detectors which can sense whether a passing small vessel carries radiological, chemical, biological, or other suspicious materials. Detectors are not perfectly reliable and the probability of detection depends on both the detector's characteristics and how long the small vessel would stay in the detector's effective detection area.

At the first step, we study how to place multiple types of fixed-position detectors to protect maritime targets from such a small vessel attack. The probability of detection depends on the type of detectors. The resulting detector placement problem is formulated as a nonlinear binary integer program such that the expected damage cost caused by the small vessel attack is minimized on the condition that the cost of detectors is within a budget limit. Two exact algorithms (standard branch and bound and linearized branch and bound algorithms) and a greedy adding heuristic are proposed. We conduct a detailed computational study and

find that the linearized branch and bound algorithm is preferred to the standard one and the greedy adding heuristic performs well.

Besides, our study addresses a more challenging problem in which the strategic behavior of a government and a terrorist group is taken into account. More specifically, we study a defender-attacker Stackelberg game with two players: a government (a leader) and a terrorist group (a follower). The government first decides the detector deployment strategy to minimize the maximum expected damage while keeping the cost of detectors within a budget limit. The terrorist group, after obtaining the full information of detector locations via intelligence, decides the small boat attacking strategy (determines attack routes) to maximize the expected damage. A path-based model is established and the resulting problem is formulated as a min-max nonlinear integer program. Two exact cutting-plane algorithms and a heuristic are proposed and tested. Computational results show the plan suggested by our defender-attacker model for the port of New York and New Jersey and the performance of the proposed algorithms.

In our above models there are three basic assumptions (detectors work independently, detectors are stationary, and the terrorist group obtains full information) which make the models simple but not reasonable for some situations. We will relax these assumptions one by one in future research.

Chapter 1

Introduction

1.1 Background

Ports are a vital component of the global transportation infrastructure. But historically they have not been subject to comprehensive governmental regulation and security oversight. The terrorist attacks of September 11, 2001 have made port safety emerge as a top priority in homeland security. More and more attention is being paid to port security. In this section we provide some background information related to maritime security.

1.1.1 Importance of Ports

Economic Importance of Ports. Driven by the economic boom in emerging markets, global maritime trade has grown significantly over the years. Ships become the primary transportation tool for the world trade. According to the

Review of Maritime Transport 2013 of the United Nations, 80 percent of global merchandise trade by volume is carried by sea and is handled by ports worldwide and the 2012 international seaborne trade was estimated to be 9 billion tons of goods [98]. In the U.S., one of the world's leading maritime trading nations, more than 95% of the nation's non-North American trade by weight and 75% by value are carried by sea. The trade has a large share of U.S. gross domestic product. In Singapore, a major international trade center, there are about 130 shipping groups contributing about 7% to Singapore's gross domestic product and employing more than 170,000 personnel [62]. As gateways for the movement of such a huge amount of goods, disruptions to this trade can have immediate and significant economic impacts.

The container shipping system is designed for speed and efficiency. During the last three decades, global economy is significantly dependent on container cargo trade because of the growth of global maritime trade and the advent of standardized cargo containers. According to the World Shipping Council, the total value of goods carried by sea in the U.S. is estimated to be \$2.23 trillion, of which roughly 55% is transported by cargo containers. According to the Review of Maritime Transport 2013 of the United Nations, world container port throughput increased by 3.8 percent to 601.8 million twenty-foot equivalent units (TEUs) in 2012 [98]. As the world's busiest container seaport, Singapore port handles around one-fifth of global container transshipment throughput. The terminals handled more than 32.6 million TEUs in 2013 [62].

Transportation services impel many companies to use the global just-in-time operation model which may incur lower inventory costs. In fact, most U.S. industries choose offshore suppliers to produce some parts. The just-in-time operation dramatically reduces costs of warehouse and inventory. Transportation efficiencies minimize the requirements of warehouses. Then, lean inventories have significant impact on the commercial productivity. Since many manufacturing companies utilize a just-in-time distribution system to lower inventory holding costs, port shutdowns may have catastrophic economic impacts. For example, the closure of 29 U.S. West Coast ports for ten days in 2002 was estimated to cost the U.S. economy \$19 billion [33].

National Security Importance of Ports. The marine transportation system is not only significant to economy, but also vital for national security. In the U.S., 17 seaports, thirteen of which are commercial seaports, have been chosen by the Departments of Defense (DOD) as strategic ports because they will be exploited necessarily by the DOD for major military deployments. For instance, ports played a crucial role for the Desert Storm war. During the war, the U.S. utilized its 18 commercial and military ports to send 312 vessels to provide military equipment and supplies. It is reported by the Government Accountability Office (GAO) that, “If the strategic ports (or the ships carrying military supplies) were attacked, not only could massive civilian casualties be sustained, but DOD could also lose precious cargo and time and be forced to rely heavily on its overburdened airlift capabilities [104].” Maritime and port security is the heart of Singapore which is a small nation with few natural resources. An attack on its port would send fatal

blow to the nation.

Because of their strategic importance and criticalness to economy and national security, ports are extremely vulnerable to a variety of terrorist threats. Moreover, ports have inherent security vulnerabilities due to their characteristics – size, open accessibility by water and land, location in metropolitan areas, the amount of materials being transported, and the ready transportation links to many locations [103].

1.1.2 Maritime Threats

There are a variety of potential threats in a port which can be categorized into two groups: piracy and maritime terrorism. Piracy is defined by the International Maritime Bureau as “an act of boarding or attempting to board any ship with the apparent intent to commit theft or any other crime and with the apparent intent or capability to use force in furtherance of that act [20].” Based on this definition, “maritime piracy includes any illegal acts of violence, detention, or depredation committed for private ends by the crew or the passengers of a private ship that is directed on the high seas against another ship, or against people or property on board a ship [41].” While the definition of maritime terrorism given by the Council for Security Cooperation in the Asia Pacific is “the undertaking of terrorist acts and activities (1) within the maritime environment, (2) using or against vessels or fixed platforms at sea or in port, or against any one of their passengers or personnel, (3) against coastal facilities or settlements, including tourist resorts, port areas and port towns or cities [20].”

The history of piracy can be dated back to the epoch when there were commerce activities on the sea. Today, it remains a main threat in the maritime area. The numbers of pirate attacks worldwide from 2009 through 2014 are shown in Figure 1.1 provided by Statista, the world's largest statistics portal. The shocking number of pirate attacks in each year is depicted.

FIGURE 1.1: The number of piracy attacks worldwide from 2009 through 2014 [94].

Compared to piracy, more recent maritime attacks are from terrorism which can be traced back to the early 1960s. Maritime terrorism is a formidable threat which is committed to achieving certain political objectives. Terrorists could potentially take the advantage of the weaknesses of ports and add maritime capabilities to strategically launch attacks on some critical infrastructures to cause massive disruptions. According to the U.S. Department of State's Report on Terrorism 2009 [102], there are ten terrorist groups having demonstrated maritime capabilities:

- Al-Qaeda
- Abu Nidal Organization

- Basque Fatherland and Liberty
- Hamas
- Hizbollah
- Jemaah Islamiya
- Lashkar-e-Tayyiba
- Liberation Tigers of Tamil Eelam
- Palestine Liberation Front-Abu Abbas Faction.

Al-Qaeda is the most destructive group with the most far-reaching strategies. Six major terrorist attacks have been implemented by al-Qaeda. The most devastating attacks are the September 11, 2001 attacks which resulted in killing approximately 3,000 people.

Luft and Korin [58] state that attacks occurred in ports in recent years imply that terrorism is going towards the sea [58]. There are 46 incidents of maritime terrorism from 2004 to 2007 globally [67]. Some examples related to maritime terrorist attacks are listed as follows:

- October 1985

The cruise ship Achille Lauro was seized by terrorists in the Mediterranean.

- January 2000

Al-Qaeda steered a boat loaded with explosives to launch an attack on USS Sullivans in Yemen. But the boat could not reach the target because of overload with explosives.

- October 2000

Al-Qaeda carried out a suicide attack against U.S. destroyer USS Cole in Yemen by a speedboat.

- October 2001

An oil tanker off northern Sri Lanka was hit by Tamil Tiger separatists who utilized five vessels to form a coordinated attack [41].

- June 2002

Morocco's secret service arrested some members of al-Qaeda who attempted to attack against British and U.S. ships in the Strait of Gibraltar [84].

- October 2002

The French oil tanker Limburg off the coast of Yemen was attacked by a bomb-laden boat.

- February 2004

Abu Sayyaf carried out an explosion on a large ferry, resulting in over 100 people killed.

- August 2005

A Syrian-led al-Qaeda plot was busted by Turkish authorities where Israeli cruise ships in international water would be hit by explosive laden boats.

- July 2006

The Israeli Saar-5 class corvette Ahi Hanit was attacked by Hezbollah.

- July 2009

An attack against the Suez Canal and the adjacent oil pipeline was prevented by Egypt security authorities [84].

Compared to piracy, maritime terrorism accounts for a small percentage of overall attacks in the sea. However, from these scenarios which have already happened, we can see one minor maritime terrorist attack can have a significant impact on a sense of panic and insecurity among public awareness and the global economy. Moreover, these terrorists and well-organized terrorist groups conduct attacks mainly using the following four strategies:

- Suicide attacks on boats
- Hitting maritime targets such as oil tankers, liquefied natural gas carriers, and transportation infrastructure using hijacked ships
- Using a small aircraft loaded with explosive to attack large vessels
- Utilizing underwater demolition teams or suicide bombers to attack vessels.

These main strategies demonstrate that small boats are most likely employed by terrorists to launch attacks, because they “satisfy the overwhelming terrorist requirement for simplicity [77].” In fact, terrorists have shown a great interest in conducting attacks using small boats. This is exemplified by the suicide attack on

U.S. Navy destroyer USS Cole in Aden, Yemen in October 2000 as shown in Figure 1.2. The terrorists navigated an explosive-laden small boat and attacked the port side of the destroyer. The incident killed 17 U.S. Navy sailors. Another prominent small-boat attack is that a boat with bombs hit French oil tanker Limberg, killing one crewman and damaging the ship as shown in Figure 1.3. Recently, the small boat threat has regained a lot of media attention, see, for example, USA Today [31], CBS News [18], and GlobalPost [70]. In April 2008, the Department of Homeland Security (DHS) of the U.S. announced the Small Vessel Security Strategy. In the report, the DHS identified four serious concerns associated with using small vessels in terrorist-related activities: using small vessels as waterborne improvised explosive devices, to smuggle weapons, to smuggle terrorists, and as waterborne platforms to carry out stand-off attacks [101]. In such an environment, this thesis focuses on small boat attack threats.

FIGURE 1.2: A small vessel attack on USS Cole ¹.

¹From http://www.cbsnews.com/8301-503543_162-57429253-503543/who-was-fahd-al-quso/.

²From http://www.informationdissemination.net/2011_05_01_archive.html/.

FIGURE 1.3: Limburg oil tanker fire caused by a small boat attack ².

1.1.3 Response Measures to Maritime Threats

Confronted with such complex, diversified, and terrible maritime threats, governments must implement effective and efficient measures for ports taking into account issues in all domains such as differing resources, transportation, access control, staff, passengers, and cargo. The basic counter-terrorism measures are developed from the following three aspects.

Law and Regulation Enforcement. In the wake of the 9/11 attacks, governments have developed a wide range of laws and regulations to strengthen maritime security. I list some of the most influential ones. The International Ship and Port Facility Security (ISPS) Code “was adopted in December 2002 and entered into force worldwide on July 1, 2004 [45].” The ISPS Code not only is a comprehensive set of measures to enhance the security of ships and port facilities, but also gives a framework on how to manage and enhance port safety. The implementation of the ISPS Code was fulfilled through the Maritime Transportation Security Act

(MTSA) of 2002 (Public Law 107-295), which strongly stresses technology. As mentioned in [38], the scopes that the MTSA focuses on include: enhancing targeting and inspection, improving the detection of explosives, nuclear and biological agents, and other dangerous materials, improving the operations of containers like labeling, fastening, and tracking, mitigating the damage caused by terrorists, and exploiting the existing technologies of other areas which can be applied to the maritime domain. For the Coast Guard and Maritime Transportation Act of 2004 (Public Law 108-293), apart from several similar topics to those contained in the MTSA, there are other requirements as stated in [38], for example, tools to improve the awareness on threats in port areas and means to enhance the safety and efficiency of port facilities when facing events which pose threat to maritime transportation security. The requirements of the Security and Accountability for Every Port Act of 2006 (Public Law 109-347) are categorized into three groups [105]: coordinating among stakeholders, conducting security operations, developing security plans, and conducting exercises to test security procedures, programs related specifically to security at individual facilities, such as examining security measures and ensuring that only properly cleared individuals have access to port areas, and cargo container security, such as screening containers at ports both here and abroad and forming partnerships with the private sector. The National Strategy for Maritime Security has four strategic objectives [106]: preventing all illegal activities such as crime and terrorism, protecting peoples and infrastructures which are related with ports, minimizing destruction and enhancing the rate of recovery, and safeguarding all things belonging to the ocean.

Cargo Container Inspection. Many concerns have been put on threats to cargo containers in a port. Cargo containers could be exploited by terrorists to smuggle terrorists and nuclear, chemical, or biological weapons. After the attacks of 9/11, interest in port and maritime security has been highlighted. Cargo container inspection is an important component because of its potential damage. Many countries have taken significant steps to improve container security. Singapore Customs launch the Security Trade Partnership (STP) program on May 25, 2007. It is a voluntary certification program that “encourages companies to adopt robust security measures in their trading operations, thereby contributing to the improvement in the security of the global supply chain [92].” The benefits from joining the program are profiled as low risk companies, reduced likelihood that their containers will be examined, and shorter inspection time or expedited clearance. The U.S. government has taken significant steps to improve container security since September 11, 2001. For example, the Container Security Initiative was launched to “establish working bilateral partnerships with foreign authorities to identify high-risk cargo containers originating at ports throughout the world before they are loaded on vessels destined for the U.S. [99].”

Physical Protection and Surveillance. Physical protection, surveillance, and methods of restriction which are widely used to improve security in various areas could be certainly applied to the maritime area. They contain some fundamental strategies such as access control and barriers, protected zones, fences, and surveillance equipment for securing critical infrastructures. Some examples are provided to show these actions against a small vessel threat which is a critical threat to

port security. The Surface Warfare Mission Package is a precision attack weapon which is designed to combat dangerous small speedy boats. It is made up of remote detectors which are fixed to an unmanned aerial vehicle and can provide surveillance in and beyond the line of sight. This package can deal with small boat attacks in different situations because it contains 30mm guns for close range use and launching systems for targets beyond the line of sight. A maritime pilot program is currently put into force by the Domestic Nuclear Detection Office in the U.S.. In the program, radiation sensors are attached to buoys to detect radiation levels from commercial and recreational ships which arrive and depart. The aim is to control the risk of weapons of mass destruction which are loaded with excessive radiation materials. In a more simplified way, many private companies exploit a range of products to construct obstacles for small unauthorized vessels.

Although various counter-threat measures in safeguarding a maritime area have been developed, a number of new problems about maritime security begin to come to light. Because attackers consistently improve their strategies and add maritime capabilities, more effective approaches should be exploited to enhance port security, especially for small boat attacks. Globally, terrorists have shown an increasing interest in using small boats to launch attacks on maritime targets such as military and commercial ships and maritime facilities. This means that governments should be committed to developing a truly comprehensive and multi-layered approach to address small boat attack threats. Motivated by this trend, we present our research scope in the next section.

1.2 Research Scope

In view of the background described in the previous section, in this thesis, we focus on the threat scenario where terrorists would utilize small boats as waterborne improvised explosive devices to launch attacks on maritime targets. As mentioned, there are many incidents where terrorists exploit small boats as waterborne improvised explosive devices to attack targets in the sea. For example, terrorists carried out an attack against the French oil tanker using an explosive-laden small vessel off the coast of Yemen in October 2002 to cause a large oil spill, killing one person and injuring four persons. The Liberation Tigers of Tamil Eelam carried out numerous successful suicide attacks using small boats against the Sri Lankan government and its assets and destroyed or damaged several civilian and military vessels [24]. Some critical infrastructures, such as military vessels, cruise ships, passenger ferries, oil tankers, and liquefied natural gas carriers, could be potential targets. Many up-to-date technologies are available to detect the presence of small amounts of radiological, chemical, and biological materials, and other dangerous materials within hundreds even thousands of meters. These devices include fixed-position detectors, human-portable radiation detection equipment, and mobile sensors. A detailed description of how detectors sense whether a small boat loads illicit materials is provided by Hill [39]. These devices are small and technologically advanced enough to be utilized at ports. For example, the DHS has announced a West Coast Maritime pilot program which involves developing a detection architecture to reduce the risk of threats transported on small vessels

[100]. The program is currently underway in Washington’s Puget Sound and California’s San Diego areas. This thesis focuses on how to deploy detectors to protect maritime targets from terrorist attacks using small vessels.

In our first work (see Chapter 3), we study how to place multiple types of fixed-position detectors to protect maritime targets from such a small vessel attack. Detectors are not perfectly reliable and the probability of detection depends on both the detector type and how long the small vessel would stay in the detector’s effective detection area. The resulting detector placement problem is formulated as a nonlinear binary integer program such that the expected damage cost caused by the small vessel attack is minimized while the cost of detectors is within a budget limit. Two exact algorithms (standard branch and bound and linearized branch and bound algorithms) and a greedy adding heuristic are proposed.

The second study (see Chapter 4) addresses a defender-attacker Stackelberg game with two players: a government (a leader) and a terrorist group (a follower). The government first decides the detector deployment strategy to minimize the maximum expected damage while keeping the cost of detectors within a budget limit. The terrorist group, after obtaining the full information of detector locations via intelligence, decides the small boat attacking strategy (determines attack routes) to maximize the expected damage. A path-based model is established and the resulting problem is formulated as a min-max nonlinear integer program. Two exact cutting-plane algorithms and a heuristic are proposed and tested.

1.3 Organization of the Thesis

The rest of this thesis is broken down as follows. Chapter 2 gives a review of some related literature and provides research gaps. Chapter 3 presents our first work which deals with how to deploy multiple types of fixed-position detectors to protect maritime targets from a small vessel attack threat. A defender-attacker Stackelberg game with two players is presented in Chapter 4. Finally, conclusions and several future research directions are stated in Chapter 5.

Chapter 2

Literature Review

The objective of this chapter is to deeply review previous studies related to our research. There are four main streams. The first stream is the context of port security from which we identify our research focus. The other three streams are methodology oriented which provide related theories and applications on how to solve our problems. An extensive literature review is broken into four categories. In Section 2.1, we first survey the studies on port security. Then, Section 2.2 and Section 2.3 offer overviews of the significant literature on sensor deployment for surveillance and game theory, respectively. Some algorithms and techniques for nonlinear binary integer programming are reviewed in Section 2.4. Finally, we summarize the review and identify some research gaps which open doors for our research in Section 2.5.

2.1 Survey on Port Security

There are many drivers, such as domestic legislation, international conventions, policy documents, and government strategies, that have set the direction for research on port security in recent years. As we mentioned in Chapter 1, there is a wide range of security requirements and programs proposed by governments to protect maritime systems from terrorist attacks. These requirements and programs provide a framework for port security research. Moreover, security experts also give instructive comments and insights on maritime security research.

Dr. Charles Massey, manager of the International Borders and Maritime Security Program at Sandia National Laboratories, identifies the need to exploit certain abilities and technologies for maritime security [38]. He points out specific research required in the context of weapons of mass destruction, drugs, and explosives, which contains:

- Stand-off sensing ability of small vessels
- Sensing capability of the presence of biological and chemical weapons
- Tracking skills of small boats
- Cargo security.

Dr. Stephen Flynn, maritime and national security expert at the Council on Foreign Relations, points out that the effect of security plans on the vulnerability of ports must be involved in the most needed critical research [38].

Based on the framework proposed by codes and laws and the insights provided by experts, there are many qualitative and quantitative studies on port security containing many aspects, such as surveillance, access control, and response.

2.1.1 Qualitative Studies on Port Security

There are a large body of qualitative studies on port security in the literature. They mainly analyze the existing situations and developments about issues related to maritime security [16, 29, 41, 52, 71].

Kopacz et al. [52] state thoughts on today's concepts and main components of maritime safety systems. An overall review of issues, such as institutions, regulations, procedures, and recent advances in radio-communication, related to maritime security is provided. Ferrier et al. [29] identify existing potential maritime security terrorism and propose possible solutions to counter the existing threats using commercial off-the-shelf technologies. It is pointed out that utilizing proven technologies will reduce regulatory oversight costs for shipping companies as well as create incentives for all stakeholders. Carafano and Kochems [16] analyze some likely major developments in the maritime realm for the next 20 years and put up some measures to develop a system-of-system maritime architecture. Hong and Ng [41] analyze the contributions and limitations of four major international legal instruments (the United Nations Convention on the Law of the Sea, the Convention on the Suppression of Unlawful Acts against the Safety of Maritime Navigation, the Proliferation Security Initiative, and the ISPS Code) which counter maritime

threats. According to the limitations of these instruments, five improved recommendations are provided: extend the regulations to tackle with more situations, clarify the definitions of different attacks to lead to more efficient approaches, establish a cooperative system, strengthen the jurisdiction over piracy, and ally the local government and harmonize the laws. Nelson [71] analyzes the characteristics of piracy and terrorism. He not only identifies the differences between piracy and terrorism from aims, attack targets, and methods but also points out the ambiguities between them. The analysis on the characteristics of maritime threats enables governments to make effective counter-measures.

Among the extensive qualitative research on port security, there is some work that concentrates on vessel attacks [21, 24, 39] which are more relevant to our research scope. Chang [21] investigates some combinations of technologies and systems to track and monitor different types of vessels and finds that an integrated system, including Inmarsat-C, automatic identification systems, and very high frequency digital selective calling, is most feasible for boat monitoring with automatic location communicators. Hill [39] determines that a small boat would be a most cost-effective and successful attack tool by analyzing the historical and current trends of maritime terrorism. He recommends that Coast Guard members and local U.S. maritime community members should have good corporation and highlights the use of up-to-date technologies and the importance of areas which are vulnerable to small vessel attacks. A report from the DHS addresses that small vessel attacks are modern devastating maritime threats which are exemplified by many successful incidents where terrorists employed small boats as waterborne

improvised explosive devices [24]. The report points out the existing laws are largely limited when facing this new challenge, and calls for coordinated efforts to enhance the small boat security environment.

2.1.2 Quantitative Studies on Port Security

Qualitative research on port security just provides some ideas and recommendations in theory. For implementation and practice, quantitative studies are clearly needed. For quantitative approaches, there are two main streams related to our work: container inspection and design of surveillance systems. We review work on these two topics separately.

Container Inspection. Most port security literature concentrates on container inspection. A container inspection system consists of a sequence of inspection sensors with different capabilities of detecting the presence of biological agents, drugs, nuclear materials, and other illegal shipments. Many up-to-date technologies are provided for cargo containers inspection, including radiation sensors, radiographic examination using either gamma rays or X-rays, and neutron-based inspection techniques [36, 37, 57, 76, 93]. Given that the sensors are technologically advanced enough to be used, the problem is to design the best inspection strategy.

Stroud and Saeger [95] investigate optimal plans for inspecting containers. The study includes the sensor model, inspection strategies represented as decision trees, and the performance measures and cost of the inspection process. The resulting model is a large-size linear program and an approach is developed based on a

hierarchical optimization process which is as follows: The first step enumerates all feasible binary inspection trees generated and the second step lists all feasible placements of sensors with optimal thresholds that are obtained by a grid optimization technique. However, the approach only provides a solution for a system in which the number of detectors cannot be larger than five, because its enumeration process is an exponential function of the number of detectors. This study is extended by Anand et al. [4] and Madigan et al. [61]. Anand et al. [4] study some variations of the method presented by Stroud and Saeger [95] in the context of different parameters which cover the costs of false negatives and false positives, the thresholds of various sensors, and the fraction of containers containing contraband. The study demonstrates the robustness of the optimal inspection strategy. But, they do not explain these conclusions about robustness theoretically and do not deeply investigate the relations among the different optimal trees. The study of Stroud and Saeger [95] is further improved by Madigan et al. [61] who exploit strategies to effectively search binary decision trees and find optimal thresholds, respectively.

Elsayed et al. [26] focus on determining the threshold levels of inspection sensors and their corresponding sequence simultaneously at a port-of-entry. The objective is to minimize the overall system cost which includes both inspection costs and misclassification costs. Moreover, they use several optimization approaches to deal with this problem. However, the cost of delay incurred in the system is not considered. Young et al. [123] extend the model to incorporate the time required to complete inspection as another objective. They formulate the problem as a

multi-objective optimization problem that attempts to minimize the total cost, as well as the delay time of container inspection.

Boros et al. [13] consider finding container inspection strategies (represented as decision trees) such that the expected per-container-inspection cost is minimized. They develop a large-scale linear programming model based on the polyhedral description of the decision trees. They argue that in the long term a combination of optimization plans is considered to be the first choice, because an attacker easily gets knowledge of the operation and pattern of a simple plan. Several evolutionary algorithms (a probabilistic solution discovery algorithm, a holistic evolutionary algorithm, and nested genetic algorithms) are proposed by Ramirez-Marquez et al [22, 83, 107].

McLay et al. [63] propose a risk-based framework to investigate how to define a system alarm. They develop a container reliability knapsack problem which decides what fractions of high-risk and low-risk containers undergo secondary screening. They conclude that effective screening has high dependence on accurate prescreening intelligence, especially for situations where sensors are highly dependent and the risk resulted from low prescreening intelligence can be mitigated by screening tests with high true alarm rates.

Merrick and McLay [66] utilize multiple-objective decision analysis to examine whether screening cargo containers for smuggled nuclear threats is worthwhile. They conclude that the result relies on some key inputs.

Maritime Surveillance Systems. Surveillance is another important concern

for port security which requires an effective and efficient system to safeguard nations from threats such as drugs smuggling, illegal immigration, trafficking of illicit materials, and other dangerous activities. An important issue for designing a surveillance system is how to place sensors in a threat field. Various technologies (for example, electro-optical, electro-magnetic, radar, and seismic) are involved and different types of detectors have their own detection characteristics such as monitoring ranges, detection capabilities, and costs. Since there are not enough resources available to cover everywhere, developing effective sensor placement strategies with an objective of either minimizing the total cost while within a required level of protection or maximizing detection while within a required level of cost is critical to a surveillance system.

There are few prior quantitative studies that focus on designing a surveillance system in a maritime area. The associated problems on this topic are addressed by qualitative methodologies in most previous studies in this area [21, 50]. Recently, Ben-Zvi and Nickerson [9] propose a method to place detectors for detecting threats in the underwater. However, the limitation is that it considers little about the behavior of attackers and the effects from the environment.

Pashko et al. [79] consider detection algorithms and sensor locations for single-period and multi-period detection models for underwater threats. Based on the classical Neyman-Pearson test and the likelihood ratio applied in sequential testing, they exploit detection algorithms for single-period and multi-period models. For the optimal sensor placement, an approximate algorithm is developed on the basis of continuous relaxations.

Wilhelm and Gokce [118] introduce an integer linear program to formulate a strategy for the surveillance of a maritime area which covers three issues: the types of detectors, the number of each type, and the optimal placement of each detector. Some practical factors, such as the environmental conditions, the shapes of areas, surveillance points, and sensor capabilities, are considered. The objective is to minimize the total cost while surveillance requirements are satisfied. Finally, they develop a branch-and-price decomposition method for the integer linear program.

Brown et al. [14] propose a bi-level (a defender and an attacker) integer linear program to position patrol vessels for port radar surveillance. The objective is to minimize the maximum evasion probability of the speedboats which are navigated by intelligent adversaries to mount attacks. The model provides the optimal positions for the defender. To solve large scale problems, a decomposition scheme is developed.

Molyboha and Zabaranin [68] deal with optimal hydrophone deployment for diver detection in an estuary or urban harbor with dynamically changing environment noise. Firstly, they introduce a diver detection scheme based on a signal processing algorithm and a statistical test. Given this scheme, the optimal hydrophone deployment problem is formulated as a two-stage stochastic program under various underwater noise scenarios. The objective is to maximize the level of detection. A deterministic energy-based method is used to solve this optimization problem.

Fang et al. [28] investigate allocating multiple mobile resources, such as fast patrol boats with installed detectors, to protect moving ferries from attacks. They apply a Stackelberg security game to formulate this problem as a linear dynamic program

where the defender seeks to minimize the maximum expected utility of attacks. A heuristic algorithm is developed for this deployment problem.

2.2 Sensor Deployment for Surveillance

Optimal sensor placement for general monitoring environments is valuable for our research since detector deployment is a major subject of this thesis. Firstly, we review the literature concentrating on stationary sensors.

2.2.1 Stationary Sensor Location Models

There exists a large body of work on static sensor location problems in some monitoring environments. We divide them into two categories on the basis of objectives: One deals with sensor location problems with the objective of minimizing cost or lost [19, 48, 73, 113] while the aim of the other is to maximize the detection ability of a system [17, 55, 78].

Chakrabarty et al. [19] consider determining the placement and type of sensors in a surveillance region such that the cost is minimized while achieving the desired coverage. An integer linear programming model is formulated and a divide-and-conquer approach is proposed for solving large problem instances.

Kaplan and Kress [48] study the operational effectiveness of suicide-bomber-detector schemes in two urban environments: a grid model and a plaza model. They compare the numbers of casualties caused by two scenarios: without surveillance and

timely detection. They conclude that even though suicide bomber sensors could play an important role in the defense of known targets, they may not be effective in random pedestrian attacks.

Wang and Zhong [113] formulate a sensor deployment problem in a 3-D area as a binary linear program where the objective is to minimize cost and one constraint is that every point of the field is covered by at least a specified number of detectors. A polynomial-time approximation algorithm is developed and the approximation ratio is provided.

Nie et al. [73] consider the optimal placement of suicide bomber detectors in a threat area where the locations of the potential targets are known. The objective is to minimize the expected number of casualties. They formulate the problem as a nonlinear integer program. A greedy adding heuristic and a branch and bound algorithm are proposed.

Park et al. [78] present an algorithm that combines global and local optimization to solve sensor location problems in some surveillance environments taking account of geographical features. The objective is to maximize detection coverage while detector redundancy is kept at a desired level.

Lin et al. [55] formulate a sensor deployment problem in a grid-based placement scenario as a combinatorial optimization problem with the objective of minimizing the maximum error of discrimination while keeping complete coverage and the required total cost constraints. A robust and scalable algorithm is proposed based on a simulated annealing approach.

Cavalier et al. [17] use a continuous approach to study the location problem of finite identical sensors in a convex polygon region. They establish a non-convex program with the objective of minimizing the maximum probability of non-detection and develop a fast heuristic based on Voronoi polygons.

2.2.2 Mobile Sensor Deployment Models

Now, we turn to mobile sensors. Mobile sensors are widely used for monitoring environments because of mobility. For sensor networks, mobility is a significant feature which makes sensors change locations to achieve detection requirements and acquire information needed, especially for sudden incidents. Mobility makes a sensor network adapt its configuration based on real situations and automatically reorganize itself even though some nodes are broken. In a word, mobility provides a kind of flexibility for the whole sensor network to efficiently complete a specific task. However, mobility also brings the sensor network some challenges like energy optimization, self-organizing ability, adaptive control, and distributed cooperative control.

We review some previous work related to mobile sensor deployment problems. Howard [43, 44] considers a mobile sensor deployment problem for the area of robotics. He uses different methods, an incremental deployment algorithm for the first one and a potential field approach for the second one, to achieve the same goal, which is maximizing the coverage of sensing.

Zou and Chakrabarty [124, 125] develop a virtual force algorithm to deploy detectors whose initial placement is generated randomly for a cluster-based distributed sensor network. The motion paths and speed of sensors are determined by a force-directed approach. The aim is to maximize coverage with a given number of sensors. They conclude that the proposed practical method has low computational costs.

Jourdan et al. [47] explore a multi-objective genetic algorithm to examine the optimal layout of a wireless sensor network. The considered objectives are the total sensing coverage and the lifetime of the entire network. Moreover, they analyze the impact of the ratio between the range of sensing to that of communication on the problem.

Wang et al. [112] propose a sensor relocation problem. In order to tackle with the problem, they develop a two-phase approach. In the first step, redundant sensors are quickly found by a grid-quorum-based method. The second phase is to relocate the identified redundant sensors by the proposed efficient heuristics while the original topology of the network is kept as much as possible.

Wu et al. [121] apply particle swarm optimization to solve a sensor deployment problem in a mobile sensor network. The process of the algorithm is as follows: Firstly, they use a probabilistic sensor model to deploy sensors uniformly to maximize coverage. Secondly, they optimize the locations of cluster heads to minimize energy consumption. They conclude that the exploited method outperforms the genetic algorithm for this problem.

Bai et al. [7] develop an improved particle swarm optimization method to deploy mobile sensors efficiently for a randomly deployed sensor network such that every point in the concerned area must be detected by at least k sensors. In their method, they allow partial sensors to move in a limited mobility manner to save energy and lower cost.

Focusing on a border surveillance system, Szechtman et al. [96] investigate the patrol policies of moving sensors with the objective of minimizing the infiltration rate under the assumption that the arrival of infiltrators is a Poisson process. They consider two types of sensor trajectories: leap-to-origin and back-and-forth.

Hochbaum and Fishbain [40] address the issue of deploying multiple mobile sensors to identify and locate nuclear threats in an urban environment. They propose the concentrated alert problem and show that the problem is solvable in polynomial time.

Luo et al. [59] first use a particle filter algorithm to decide a mobile sensor node from which highly accurate location information can be obtained. Then, they develop an easily implementable and efficient deployment method called a grid deployment method to achieve a high level of coverage. Moreover, they develop an asynchronous power management technique to deal with the power consumption problem for the wireless sensor network taking account of many factors such as battery status, communication situations, coverage issues, and probability of event generation.

2.3 Game Theory

In this thesis, we also consider the strategy of a terrorist group when designing an efficient surveillance system because the terrorist is very intelligent as exemplified by the tragic events of September 11, 2001. To deal with the strategic behavior of two adversary players, a natural approach is through game theory. Game theory is defined as a mathematical tool for modeling strategic decision making problems where two or more intelligent rational individuals decide their choices which have an impact on others' rewards. As such, game theory is widely applied to many fields, such as political science, computer science, and security, and can offer general mathematical skills and insights to analyze decision situations. In a game, there are a set of players and each player selects a pure or mixed strategy from his own set of strategies with the objective of maximizing his payoff. A strategy is defined as "a rule for action so complete and detailed that a player need not actually be present once his strategy is known [114]." Based on whether players select their strategies simultaneously, games can be divided into two groups: Cournot games (simultaneous games) and Stackelberg games (sequential games). In a Courant game, there are two scenarios. One is that all players reveal their choices simultaneously, and the other is that the players who make decisions later have no knowledge of the actions of the players who make decisions earlier. In a Stackelberg game, some players select strategies first, then other players make decisions based on some even very little information about the strategies of the earlier players. Many homeland security issues have been considered as Stackelberg

games where the earlier player is a government and the later player is a terrorist. We concentrate on Stackelberg games which are appropriate for modeling our problem.

2.3.1 Application Models for Stackelberg Games

Stackelberg game theory, a branch of game theory, is concerned with a two-level hierarchical structure in which in the upper level a leader makes his decision first, while in the lower level a follower makes his own decision after observing the leader's decision. The two decision makers have different and conflicting objectives. Many problems in security fields, which include air marshals problems [75, 97], infrastructure security [11, 85], robot patrolling problems [2, 8, 32], and network security [3, 60, 72], are cast as Stackelberg games. We focus on target or infrastructure protection problems which are closer to our problem [10, 11, 15, 80, 81]. The games in the infrastructure security area have the following common characteristics: (1) There are two opposing forces (a defender and an attacker), (2) the defender has a limited security resource to provide protection for everything all the time, (3) the attacker can know some pattern of defense, and (4) both players try to find an equilibrium as a solution.

Brown et al. [15] establish bi-level and tri-level optimization programs to model defender-attacker, attacker-defender, and defender-attacker-defender security Stackelberg games. The problem in each level is presented as a linear program. In these games, an intelligent attacker launches attacks on some critical assets to maximize the harm, while a defender wants to minimize the total cost by designing optimal

defense plans. They apply these mathematical models to many real situations, which verifies the practicality of the models.

Berman and Gaviols [10] investigate a Stackelberg game between a state and a terrorist where the state wants to minimize the damage caused by the terrorist by locating facilities with resources, while the terrorist determines attacking targets to maximize his utility under the assumption that he completely knows the actions of the state. They formulate this problem as a nonlinear integer program and provide an exact optimal solution for one facility case.

Bier et al. [11] use a Stackelberg game perspective to analyze a security problem where a defender seeks to find optimal resource allocation for two assets, while an attacker makes a decision on which site to attack under the assumption that the attacker knows the values of the two assets. The objective of the defender is to minimize the total cost including the expected loss caused by the attacker and the defense cost, while the aim of the attacker is to maximize the expected damage. A pure-strategy sequential equilibrium is provided to analyze the behavior and preferences of the two players.

Powell [81] applies a Stackelberg game to solve a resource-allocation problem. In the model, a defender tries to optimize resource allocation to defend multiple potential sites, while a strategic adversary chooses one target to attack. He analyzes the problem with two types of games: a zero-sum game and a general game. In the zero-sum game, the loss of the defender at each target equals to the gain of the attacker at that site, while in the non-zero-sum game, the valuations of the sites are different between the two opposite players. The goal of the defender is

to minimize the maximum gain of the attacker. He proposes a greedy-based algorithm to get a solution in which the defender sequentially allocates resources to the target with the highest expected payoff given that the previously chosen sites have already been protected.

Pita et al. [80] focus on protecting over 400 airports for the U.S.. They use a Stackelberg game framework to model the comprehensive and complicated security problem which contains a number of defender activities and a variety of threats. In the model, the defender is the Transportation Security Administration of the U.S. which can utilize some defending activities with given resources to strengthen the safety of the airports, while the attacker attempts to choose targets based on the information of the defender's allocation of resources. They use compact representation to present the defender's actions to obtain an improved model which can be solved efficiently by some existing solvers. Finally, they develop a practical software package called Game-theoretic Unpredictable and Randomly Deployed Security.

2.3.2 Network Interdiction

The network interdiction problem also involves two opposing forces: an interdictor and an adversary. The interdictor destroys certain arcs with an objective of either minimizing the adversary's maximum flow [120] or maximizing the adversary's shortest path [46]. Strategies can be provided by interdiction models for the interdictor to block the activities of the adversary, such as delaying the delivery process of a supply, detecting a stealth traverse, and decreasing the adversary's

profit. The study of network interdiction problems begins in the sixties of the last century. Wollmer [119] investigates an interdiction problem of removing some arcs to maximize the reduction of maximum flow between a fixed source and destination pair. Later, this problem is applied to military operations. McMasters and Mustin [64] and Ghare et al. [34] develop deterministic mathematical programs to disrupt an enemy's logistics. Furthermore, there are many researchers who study drug interdiction problems such as Wood [120] and Washburn and Wood [115]. Here, we focus on network interdiction problems.

Washburn and Wood [115] consider a deterministic network interdiction problem where a single interdictor seeks to find an optimal arc to install inspection equipments in the network to minimize the evasion probability, while a single adversary attempts to select a path with the minimum detection probability between a fixed entrance and destination pair. Each arc is associated with a known detection probability. They formulate the two-person game as a max-min model and transform the problem to a simple minimum-cut problem which can be solved in polynomial time. Finally, they extend the basic model to several variants by considering multiple interdictors and adversaries, multiple sources and sinks, and other real generalizations.

Morton et al. [69] propose two stochastic variants for nuclear smuggling interdiction problems based on two assumptions. One is that a smuggler obtains full knowledge of the locations of detectors, and the other is that a smuggler obtains only partial information. In both cases, a defender locates sensors on a transportation network to minimize the attacker's maximum probability of evading detection.

The two models are stochastic due to the probability distribution of the smuggler's selected entrance and destination pairs. They build mixed integer programs for both cases. Finally, they discuss a special case where detectors are only deployed at the border crossing points of a single country.

In a series of papers, Wein et al. consider designing radiation detection-interdiction systems to protect cities from nuclear terrorist attacks. They focus on a threat scenario where a terrorist attempts to drive a nuclear or radiological weapon toward a target in a city center. They formulate three models: a sensor model [117], an optimal stopping problem [5], and a spatial interdiction model [6]. Specifically, the sensor model determines detection and false positive probabilities in terms of sensor threshold levels. In the optimal stopping model, the terrorist determines whether to proceed or to detonate the bomb at each node along his route. The interdiction model proposes a spatial queueing model to determine the mean damage inflicted by a terrorist in terms of the number of interdiction vehicles. Wein and Atkinson [116] embed these three models and formulate a Stackelberg game with two players: a government (a leader) and a terrorist (a follower). The government chooses the number of sensors, sensor thresholds, and the number of interdiction vehicles to minimize the expected damage caused by the terrorist. The terrorist then decides whether to move forward or to detonate the bomb with the objective of maximizing the expected damage.

Yates et al. [122] present a new formulation for a network detection-interception problem in which a defender places sensors in an important geographic area to minimize the maximum network vulnerability, while an attacker seeks to pass

through a path without being detected. This problem is considered as a variant of shortest-path network interdiction problem. They formulate this problem as a bi-level mixed binary integer program and use a decomposition method to get an optimal solution.

Schmidt et al. [86] and Vaněk et al. [109] develop efficient detection-interdiction systems for computer network security. The former one studies a non-cooperative game between an attacker and an intrusion detection system in a network whose configuration is set as a finite-state Markov chain. In the model, the malware generated by the attacker seeks to intrude into the target system by choosing an optimal route, while the interdictor decides the optimal locations of detectors to detect the attempt. Furthermore, the attacker is intelligent enough to get full knowledge of the current state of the network. Finally, they analyze some special cases and propose the corresponding algorithms to solve them. The later one provides a nonlinear program to model a game of packet selection and inspection. In the game, a defender tries to minimize the expected loss caused by an attacker using his best strategy. The strategy of the defender is to allocate the limited resource to sense malicious packets sent by the attacker whose strategy is to choose entrances and targets to result in maximum harm. They develop an approximation algorithm to obtain error-bounded solutions for realistic-size problems.

2.4 Relevant Algorithms and Techniques for Non-linear Binary Integer Programming

In this section, we review some algorithms related to our research. Since our first problem is formulated as a nonlinear binary integer program and our second work is modeled as a min-max program where the upper-level problem is also a nonlinear binary integer program, we compile a review of the literature on algorithms for solving nonlinear binary integer programs.

A wide range of practical problems arising from engineering and scientific fields can be represented as nonlinear binary integer programs. However, nonlinear binary integer programming is NP-hard since it has two challenging characteristics: integrality and nonlinearity. Hence, there is considerable research focusing on finding the optimal solutions of nonlinear binary integer programs. It is a daunting task to provide a comprehensive literature review on this topic. Here, we just review some of them which are most relevant to our research in two ways: how to deal with integrality and how to handle nonlinearity.

2.4.1 Standard Branch and Bound Algorithm

The branch and bound algorithm is widely regarded as an effective method to obtain an integer optimal solution since proposed by Land and Doig [53] and Dakin [23], and extended by Falk and Soland [27] and Horst and Tuy [42]. When solving a minimization problem, the idea of the branch and bound algorithm is

to improve the lower bound successively by solving relaxation problems which are selected based on some branching rule. The procedure of the standard branch and bound algorithm for a nonlinear binary minimization problem is as follows. For the ease of exposition, we use a vector $\mathbf{X} = (x_1, x_2, \dots, x_n)$ to represent a feasible solution.

Standard Branch and Bound Algorithm

Step 0. (Initialization) Set stage $s = 0$, active node set $O_s = \{0\}$, current node $n(s) = 0$, and $n = 0$. Let upper bound UB be the objective function value of the incumbent solution where all decision variables are equal to zero. Let ε be some optimality tolerance. Solve the relaxation problem constructed by relaxing the binary constraints and obtain a solution $\mathbf{X}^0$. If all values of $\mathbf{X}^0$ happen to be integer, we stop the procedure and $\mathbf{X}^0$ is an optimal solution. Otherwise, go to Step 1.

Step 1. (Division) Split node $n(s)$ into two child nodes (labeled by $n+1$ and $n+2$, respectively) through the following branching rule. Let $\mathbf{X}^{n(s)}$ be the solution of node $n(s)$. Find index i' such that $\mathbf{X}^{n(s)}(i')$ is the first non-integer element. Add a constraint $\mathbf{X}(i') = 0$ to the problem for node $n+1$ and $\mathbf{X}(i') = 1$ to the problem for node $n+2$. Then, we update O_s by $O_s = O_s \cup \{n+1, n+2\} \setminus \{n(s)\}$.

Step 2. (Bound updating and fathoming) Solve the relaxation programs of nodes $n+1$ and $n+2$. Update the incumbent solution and its objective function value UB. If the relaxation program is infeasible, or if the objective function value is larger

than $UB(1 - \varepsilon)$, we fathom the node. Update O_{s+1} by removing all fathomed nodes from O_s and let $n = n + 2$ and $s = s + 1$.

Step 3. (Termination and node selection) If O_s is an empty set, stop the procedure and return the incumbent solution. Otherwise, select a node $n(s)$ from O_s with the smallest objective function value and go to Step 1.

2.4.2 Linearization Techniques

Though the branch and bound algorithm obtains a solution within optimality tolerance ε , the relaxation problems repeatedly solved during the procedure are nonlinear programs. In the literature, many linearization techniques have been developed to handle nonlinearity for some specially structured nonlinear programs. Among them, a reformulation-linearization technique (RLT), which is used to solve discrete and continuous nonconvex programs, is one of the most noteworthy methodologies. For some minimization problems, the RLT can provide tighter linear relaxations.

Adams and Sherali [1] use an algorithm based on the RLT strategy to solve binary quadratic programming problems with linear constraints. The binary quadratic problem is presented as the following program:

$$\min \sum_{i=1}^m c_i x_i + \sum_{i=1}^{m-1} \sum_{j=i+1}^m d_{ij} x_i x_j \quad (2.1)$$

$$\text{s.t.} \quad \sum_{i=1}^m a_{ki} x_i = b_k, \quad k = 1, \dots, K, \quad (2.2)$$

$$\sum_{i=1}^m G_{li} x_i \geq g_l, \quad l = 1, \dots, L, \quad (2.3)$$

$$0 \leq x_i \leq 1, \quad i = 1, \dots, m, \quad (2.4)$$

$$x_i \in \{0, 1\}, \quad i = 1, \dots, m. \quad (2.5)$$

They multiple equations (2.2) and inequalities (2.3) by each $x_j, j = 1, \dots, m$, inequalities (2.3) by each $(1 - x_j), j = 1, \dots, m$, $x_i \geq 0, i = 1, \dots, m$, by each $(1 - x_j), j = 1, \dots, m, j \neq i$, and $x_i \leq 1, i = 1, \dots, m - 1$, by each $(1 - x_j), j = i + 1, \dots, m$, to form more constraints. Then, they use a substitution strategy $y_{ij} = x_i x_j \geq 0, \forall (i, j), i < j$, and equations $x_i^2 = x_i, \forall i$, to get an equivalent mixed binary linear program as follows:

$$\begin{aligned}
\min \quad & \sum_{i=1}^m c_i x_i + \sum_{i=1}^{m-1} \sum_{j=i+1}^m d_{ij} y_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^m a_{ki} x_i = b_k, & k = 1, \dots, K, \\
& \sum_{i=1}^m G_{li} x_i \geq g_l, & l = 1, \dots, L, \\
& 0 \leq x_i \leq 1, & i = 1, \dots, m, \\
& (a_{kj} - b_k)x_j + \sum_{i < j} a_{ki} y_{ij} + \sum_{i > j} a_{ki} y_{ji} = 0, & \forall (j, k), \\
& (G_{lj} - g_l)x_j + \sum_{i < j} G_{li} y_{ij} + \sum_{i > j} G_{li} y_{ji} \geq 0, & \forall (j, l), \\
& x_i - y_{ij} \geq 0, & \forall (i, j), i < j, \\
& x_j - y_{ij} \geq 0, & \forall (i, j), i < j, \\
& -x_i - x_j + y_{ij} \geq -1, & \forall (i, j), i < j, \\
& y_{ij} \geq 0, & \forall (i, j), i < j, \\
& x_i \in \{0, 1\}, & i = 1, \dots, m.
\end{aligned}$$

From the above description, we can see that the operation of the RLT is divided into two phases. In the first phase (a reformulation phase), more polynomial constraints are added into the problem by multiplication. In the second phase (a linearization phase), the resulting problem in the first phase is linearized by introducing new variables to perform proper variable substitutions. The RLT approach is extended to solve general quadratic and polynomial programming problems [89–91].

Polyhedral outer approximation is another useful and powerful linearization technique to deal with nonlinearity. This technique has been developed to solve a variety of optimization problems since the pioneering works of Kelley [49] and Veinott [110]. The basic idea is to relax a feasible region by constructing a sequence of polyhedral outer approximations. The process is as follows: A feasible region S is relaxed to a polyhedron S_1 and the resulting relaxation problem is solved to get a solution. If the solution happens to be in S , the optimal solution is obtained. Otherwise, an improved polyhedron S_2 is yielded to replace S_1 by adding a linear constraint to cut an appropriate portion of $S_1 \setminus S$ and the procedure is repeated.

In order to illustrate how to develop algorithms based on polyhedral outer approximation to solve problems, we consider the work [87] which provides a basic idea for our way of solving nonlinear integer programs. In the work, they use the technique to deal with a nonlinear continuous program which is defined as follows:

$$\min \sum_{j \in J} \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk} + \mu_D \sum_{j \in J} \eta_j + \mu_R \eta \quad (2.6)$$

$$\text{s.t.} \quad \sum_{j \in J} \sum_{k \in K_j} x_{ijk} \leq b_i, \quad \forall i \in I, \quad (2.7)$$

$$\eta_j \geq \gamma_j - \bar{\gamma}, \quad \forall j \in J, \quad (2.8)$$

$$\eta_j \geq \bar{\gamma} - \gamma_j, \quad \forall j \in J, \quad (2.9)$$

$$\eta \geq \gamma_j - \bar{\gamma}, \quad \forall j \in J, \quad (2.10)$$

$$\gamma_j \left(\sum_{k \in K_j} r_{jk} \alpha_{jk} \right) = \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk}, \quad \forall j \in J, \quad (2.11)$$

$$|J| \bar{\gamma} = \sum_{j \in J} \gamma_j, \quad (2.12)$$

$$z_{jk} + \sum_{i \in I} \beta_{ik} x_{ijk} = 0, \quad \forall j \in J, k \in K_j, \quad (2.13)$$

$$z_{jk} = \ln(y_{jk}), \quad \forall j \in J, k \in K_j, \quad (2.14)$$

$$x_{ijk} \geq L_{ijk}, \quad \forall i \in I, j \in J, k \in K_j, \quad (2.15)$$

$$l_{jk} \leq y_{jk} \leq u_{jk}, \quad \forall j \in J, k \in K_j, \quad (2.16)$$

where $r_{jk}, \alpha_{jk}, \beta_{ik}, b_i, \mu_D, \mu_R, L_{ijk}, l_{jk}$, and u_{jk} are parameters.

Notice that the above model is a linear program except that constraints (2.14) are nonlinear. Hence, they use polyhedral outer approximation to relax the constraints (2.14). The approximation consists of a convex envelope and tangential supports at some points. The corresponding relaxation problem is as follows:

$$\begin{aligned}
\min \quad & \sum_{j \in J} \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk} + \mu_D \sum_{j \in J} \eta_j + \mu_R \eta \\
\text{s.t.} \quad & \sum_{j \in J} \sum_{k \in K_j} x_{ijk} \leq b_i, & \forall i \in I, \\
& \eta_j \geq \gamma_j - \bar{\gamma}, & \forall j \in J, \\
& \eta_j \geq \bar{\gamma} - \gamma_j, & \forall j \in J, \\
& \eta \geq \gamma_j - \bar{\gamma}, & \forall j \in J, \\
& \gamma_j \left(\sum_{k \in K_j} r_{jk} \alpha_{jk} \right) = \sum_{k \in K_j} r_{jk} \alpha_{jk} y_{jk}, & \forall j \in J, \\
& |J| \bar{\gamma} = \sum_{j \in J} \gamma_j, \\
& z_{jk} + \sum_{i \in I} \beta_{ik} x_{ijk} = 0, & \forall j \in J, k \in K_j, \\
& z_{jk} \geq \ln(l_{jk}) + \frac{\ln(u_{jk}) - \ln(l_{jk})}{u_{jk} - l_{jk}} (y_{jk} - l_{jk}), & \forall j \in J, k \in K_j, \\
& z_{jk} \leq \ln(\bar{y}_{jk}) + \frac{y_{jk} - \bar{y}_{jk}}{\bar{y}_{jk}}, & \forall \bar{y}_{jk} \in \{l_{jk} + \frac{t}{n_{jk} - 1} (u_{jk} - l_{jk}), \\
& & t = 0, \dots, n_{jk} - 1\}, \forall j \in J, k \in K_j, \\
& x_{ijk} \geq L_{ijk}, & \forall i \in I, j \in J, k \in K_j, \\
& l_{jk} \leq y_{jk} \leq u_{jk}, & \forall j \in J, k \in K_j.
\end{aligned}$$

The above relaxation problem is a linear program which can be efficiently solved by existing software. Based on the relaxation scheme, they develop a branch-and-bound algorithm for solving the initial nonlinear program. The technique has been utilized in several application domains, for example, event tree optimization [88] and wireless network [51]. For our nonlinear binary programs, the polyhedral outer approximation is used to relax the nonlinear binary programs into mixed integer linear programs.

2.5 Literature Summary and Research Gaps

For port security, there are two aspects, perimeter security and access control, from which security plans are designed to counter potential threats. Our main focus is to design an effective surveillance system for perimeter security by optimally locating sensors. There are numerous threats in a port area. We are interested in a small vessel attack threat because it is a most likely attacking tool as shown in Chapter 1. Moreover, from the literature review, we observe that there are few quantitative studies addressing the small vessel attack threat in port security.

Our research is from the perspective of deploying detectors. The early work on this topic has an assumption that detectors are perfectly reliable, for example, [19, 48]. In reality, the operations of detectors have natural uncertainty which should be measured by probability. Some later work [14, 78, 118] relaxes this assumption. The objectives of these studies are divided into two categories: the cost of placing sensors and the ability of sensing. With these two kinds of objectives, easier mathematical programs can be established. For example, since the objectives of Wilhelm and Gokce [118] and Brown et al. [14] involve only the probabilities of evasion, both models can be formulated as linear integer programs (a linear integer program and a bi-level linear integer program, respectively). Recently, Nie et al. [73] not only measure the ability of sensing in probability, but also have an objective function of minimizing the expected damage cost which is closer to actual need. The resulting optimization model is a nonlinear binary integer program, which cannot be linearized through simple transformations. This work has various limitations which open doors for our research. We recognize the following topics:

- The first limitation is that they just consider how to place one type of stationary detector. In fact, there are many advanced technologies which could be used for port security. We plan to extend their work to incorporate multiple types of stationary detectors in a maritime domain.
- Secondly, they assume that terrorists know nothing about the locations of detectors. It can be seen from the historical terrorist attacks that terrorists are highly intelligent. Game theory will be applied to deploy detectors.
- We note that they consider the situations where detectors work independently. In many scenarios, there are dependencies among detectors. Joint detection probability should be studied.
- The last limitation we plan to exploit is that only stationary detectors are used. We know that the mobility of detectors provides more efficiency for surveillance systems. We plan to deploy mobile or both mobile and stationary detectors.

Based on these identified research gaps, the aim of this thesis is to construct models, develop algorithms, and provide insights. In the next chapter (Chapter 3), we focus on our first topic, that is, deploying multiple types of stationary detectors to protect a port from a small vessel attack threat. In Chapter 4, we fill the second research gap, that is, designing port surveillance plans from a Stackelberg game perspective.

Chapter 3

Optimal Placement of Multiple Types of Detectors under a Small Vessel Attack Threat to Port Security

3.1 Introduction

As stated in Chapter 1, ports are vulnerable to a variety of threats. A small vessel attack threat is one of top concerns by the DHS for port security. Especially, it is a key element of consideration for making future maritime security policy [65]. Small vessels are increasingly exploited to launch attacks by terrorists since they can cause stunning destructions with low investment. For instance, an al-Qaeda

leader Khallad bin Attash plotted to use suicide terrorists to crash a small boat into U.S. naval vessels at Port Klang, Malaysia in January 2000. The best known small vessel attack is that terrorists navigated an explosive-laden small boat to attack U.S. Navy destroyer USS Cole in Aden, Yemen in October 2000. More recently, a small boat attacked Japanese tanker M. Star in the Strait of Hormuz in August 2010. Advanced technologies have been explored to weaken the small boat attack threat. For example, many up-to-date technologies are available to detect the presence of radiological, chemical, and biological materials within hundreds of meters, and the abnormal states of a boat within hundreds even thousands of meters.

In this chapter, we concentrate on the threat scenario where terrorists would utilize small vessels as waterborne improvised explosive devices to attack maritime targets. Potential targets could be military vessels, cruise ships, passenger ferries, oil tankers, liquefied natural gas carriers, and other critical infrastructures. Since different technologies (for example, electro-optical, electro-magnetic, radar, and seismic) are involved, each type of detectors has its own detection characteristic and cost. Clearly, the costs of sensors increase as their detection abilities rise. However, the use of expensive, advanced sensors may lead to high total placement costs. On the other hand, if sensors that are cheap and have poor performance are deployed, a large number of these sensors are necessary for effective surveillance. Hence, this chapter focuses on how to place multiple types of fixed-position detectors to protect maritime targets from a small vessel attack threat.

We assume that detectors are not perfectly reliable and the probability of detection

depends not only on the type of the detector but also on how long the small vessel would stay in the detector's effective detection area. The expected damage cost caused by the small vessel attack is minimized while keeping the cost of detectors within a budget limit. The resulting optimization problem is formulated as a nonlinear binary integer program. Moreover, two exact algorithms and a greedy heuristic are proposed to solve the corresponding nonlinear binary integer program.

This study has three contributions. First, incorporating different characteristics of detectors (for example, detection rates, effective detection radiuses, and costs), we consider how to place multiple types of detectors to protect maritime targets. Second, we introduce two dominance definitions (strong dominance and weak dominance), investigate their relationship, and derive a property which can reduce the feasible regions of our optimization models. Third, we propose another problem formulation which is amenable to linear outer approximation, and design a linearized branch and bound algorithm which computationally proves to be more efficient than a standard branch and bound algorithm.

The chapter is organized as follows. The modeling is presented in Section 3.2. Section 3.3 concentrates on some properties. Two branch and bound based exact algorithms and one greedy adding heuristic are proposed in Section 3.4. A computational study is provided in Section 3.5.

3.2 Modeling

In a port area, there is a collection of valuable assets (for example, military vessels, cruise ships, and oil tankers) anchoring in the sea. These assets could be potential targets for a terrorist attack utilizing a small vessel as a waterborne improvised explosive device. Possible entrances for the small vessel are docks and marinas. In order to protect maritime targets away from such an attack, placing stationary detectors that can sense whether a passing small vessel carries explosive materials is considered. We assume that the detectors are perfectly concealed (for example, attached to buoys). Different types of detectors may own different characteristics (for example, detection rates, effective detection radiuses, and costs). We investigate how to place multiple types of detectors in a port area such that the expected damage cost is minimized.

Without loss of generality, we assume that the port area is rectangular. For the simplification of modeling, the rectangle is discretized into square grids. To model possible obstructions in the port area, we categorize grids into two types: blocked grids (a small vessel can not travel through) and unblocked grids. Let D be the set of all blocked grids. There is a set of entrances, denoted by E , along the boundary of the rectangle. Within the rectangle, there exists a set of potential targets, denoted by S . All entrances and targets are located in the centers of some unblocked grids. An illustrative example is provided in Figure 3.1 where a shaded region represents a blocked grid, a triangle represents an entrance, and a star represents a target. In this example, $D = \{13, 14, 18, 39, 40, 52\}$, $E =$

$\{32, 33, 58, 62\}$, and $S = \{12, 29\}$. In this chapter, we use grid and grid center interchangeably.

FIGURE 3.1: An example and the associated shortest paths.

The terrorist selects one entrance and steers the small vessel toward the chosen target. Traveling from a grid center to another grid center is allowed as long as the small vessel does not intersect any blocked grid. We assume that the small vessel tries to travel on the straight line path for the selected entrance and target pair. If this straight line path happens to cross one or more blocked grids, the small vessel chooses a shortest path that is composed of a sequence of straight line segments

not intersecting any block grid. The shortest path can be constructed through some shortest path algorithm, for example, Dijkstra's algorithm [25]. Continuing the example shown in Figure 3.1, the shortest path from entrance 32 to target 12 consists of two straight line segments: $32 \rightarrow 20$ and $20 \rightarrow 12$. For the simplicity of exposition, we assume that the shortest path from entrance k to target j is unique and is labeled as P_{kj} . Let γ_{kj} be the probability that the terrorist utilizes path P_{kj} . Hence, $\gamma_{kj} \geq 0, \forall k \in E$ and $j \in S$ and $\sum_{k \in E} \sum_{j \in S} \gamma_{kj} = 1$.

We assume that the centers of grids (blocked and unblocked grids) are potential locations for detectors. Let H be the set of all detector types, indexed by h . For a type h detector, the corresponding effective detection radius, instantaneous detection rate, and unit purchasing cost are τ_h , η_h , and ρ_h , respectively. In order to allow the interdiction team to have enough time to take action after detection, we define timely detection as detection such that there are at least λ seconds remaining before the small vessel reaches the target. Let the driving speed of the small vessel be ψ m/s. A time period of λ seconds converts to a distance of $\lambda\psi$ meters away from the target.

Define N_{kj}^h as the collection of all grids from which a detector of type h can timely detect the small vessel traveling on path P_{kj} . The set N_{kj}^h can be obtained geometrically. First, we construct a timely detection zone which includes all points in the rectangle from which a type h detector can timely detect the small vessel on path P_{kj} (we need to check if the distance from one point to its nearest point on path P_{kj} is less than τ_h and if the first point of detection is at least $\lambda\psi$ meters away from target j). Then, we check if a grid center is within the timely detection

zone. If yes, we include it into N_{kj}^h . Figure 3.2 provides an example where the grid size is $200 \text{ m} \times 200 \text{ m}$, $\tau_2 = 200 \text{ m}$, $\lambda = 10 \text{ s}$, and $\psi = 20 \text{ m/s}$. In this example, $N_{32,12}^2 = \{20, 21, 22, 23, 24, 28, 29, 30, 31, 32\}$.

FIGURE 3.2: An example for N_{kj}^h and detection length l_{hikj} .

We assume that all types of detectors are not perfectly reliable and the probability of timely detecting the small vessel depends on how long the small vessel stays within a specific type of detector's effective detection radius. For every $i \in N_{kj}^h$, we define p_{hikj} as the probability that a type h detector placed on grid i timely detects the small vessel traveling on path P_{kj} . The procedure of calculating p_{hikj}

is as follows: First, we construct a sub-path of path P_{kj} , labeled as $\bar{P}_{kj}$, which excludes the portion within $\lambda\psi$ meters from target j . Then, we obtain the length of the part of $\bar{P}_{kj}$ that is within τ_h meters from grid i and define this length as l_{hikj} . If entrance k happens to be inside of the circle with radius τ_h centered at grid i , we stretch the first segment of $\bar{P}_{kj}$ to intersect the circle. Finally, based on [82], we calculate p_{hikj} by using the following formula:

$$p_{hikj} = 1 - e^{-\eta_h l_{hikj}},$$

reminding that η_h is the instantaneous detection rate for a type h detector. Following the example shown in Figure 3.2, $l_{2,22,32,12}$ equals to 349.790 meters. Given that $\eta_2 = 0.005$, we obtain that $p_{2,22,32,12} = 1 - e^{-0.005 \times 349.790} = 0.826$.

Define $T^h = \cup_{kj} N_{kj}^h$, which includes all possible locations from which a type h detector can timely detect the small vessel. Moreover, defining $T = \cup_h T^h$, we can confine the detector locations to belong to T . We assume that we can place at most one detector in one grid and there is a budget limit M for procuring detectors. For each h and each $i \in T^h$, we define a binary decision variable x_{hi} , which equals to 1 if a detector of type h is placed in grid i and equals to 0 otherwise.

We assume that if the small vessel is timely detected, the intervention team foils the attack with probability θ . There are two possible scenarios in which the attack via path P_{kj} is successful. One scenario is that the small vessel is not detected with probability

$$\prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}}.$$

The other one is that the small vessel is detected but the team fails to stop the attack with probability

$$(1 - \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}})(1 - \theta).$$

For each target $j \in S$, let C_j be the resulting damage cost if the attack is successful.

Since the small vessel uses path P_{kj} with probability γ_{kj} , we obtain the expected damage cost as follows:

$$\sum_{k \in E} \sum_{j \in S} \left[\prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}} \gamma_{kj} C_j + (1 - \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}})(1 - \theta) \gamma_{kj} C_j \right]. \quad (3.1)$$

Rearranging (3.1), we obtain the following expression:

$$\sum_{k \in E} \sum_{j \in S} \left[(1 - \theta) \gamma_{kj} C_j + \theta \gamma_{kj} C_j \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}} \right]. \quad (3.2)$$

Inside the square bracket of (3.2), the first term $(1 - \theta) \gamma_{kj} C_j$ is constant and hence it can be removed without affecting the optimal solution. Moreover, for convenience, we define $W_{kj} = \theta \gamma_{kj} C_j$. The resulting detector placement problem is formulated as the following nonlinear binary integer program:

$$[\text{DP}] \quad \min \quad \sum_{k \in E} \sum_{j \in S} W_{kj} \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}} \quad (3.3)$$

$$\text{s.t.} \quad \sum_{h: i \in T^h} x_{hi} \leq 1, \quad \forall i \in T, \quad (3.4)$$

$$\sum_{h \in H} \sum_{i \in T^h} \rho_h x_{hi} \leq M, \quad (3.5)$$

$$x_{hi} \in \{0, 1\}, \quad \forall h, i \in T^h. \quad (3.6)$$

The objective function (3.3) is to minimize the expected damage cost of the small vessel attack. Constraints (3.4) ensure that there is at most one detector in one grid. Constraint (3.5) ensures that the cost of placing detectors is within the budget limit. Constraints (3.6) are binary constraints.

3.3 Properties

Dominance has been frequently exploited to reduce the feasible region of an optimization problem in the network and discrete location literature. In this section, we first provide two definitions for dominance and then explore some properties based on the definitions.

Definition 3.1. (Strong dominance) Consider two grids u and v . If $p_{hukj} \geq p_{hvkj}$ for every path P_{kj} and every detector type h , and moreover there exists at least one path P_{kj} and one detector type h such that $p_{hukj} > p_{hvkj}$, we say that grid u strongly dominates grid v .

Definition 3.2. (Weak dominance for detector type a) Consider two grids u and v and one specific detector type a . If $p_{aukj} \geq p_{avkj}$ for every path P_{kj} , and moreover there is at least one path P_{kj} such that $p_{aukj} > p_{avkj}$, we say that grid u weakly dominates grid v for detector type a .

From the two dominance definitions, we observe that if grid u strongly dominates grid v , grid u weakly dominates grid v for all types. But weak dominance for a specific type does not imply strong dominance. Moreover, even though grid

u weakly dominates grid v for the type with the largest radius, grid u may not weakly dominate v for other types. Hence, grid u may not strongly dominate grid v . A counter example is provided in Figure 3.3.

FIGURE 3.3: A counter example.

Example 3.1. Consider two detector types. Type b has detection radius $\tau_b = 200$ m and instantaneous detection rate $\eta_b = 0.005$, while type c has $\tau_c = 180$ m and $\eta_c = 0.004$. As shown in Figure 3.3, we consider two grids u and v and one unique path $P_{k'j'}$ which has one line segment within a circle centered at grid u with radius 200 m and two line segments within a circle centered at grid v with the same radius. We obtain that $l_{buk'j'} = 360.000$, $l_{cuk'j'} = 314.960$, $l_{bvk'j'} = 2 \times 179.003 = 358.006$, and $l_{cvk'j'} = 2 \times 158.643 = 317.268$. Hence, we have that

$$p_{buk'j'} = 0.835, p_{bvk'j'} = 0.833, p_{cuk'j'} = 0.716, \text{ and } p_{cvk'j'} = 0.719.$$

Based on these detection probabilities, we observe that grid u weakly dominates grid v for type b , but grid v weakly dominates grid u for type c .

Let type h_1 be the detector type with the largest radius. Even though in general weak dominance for type h_1 does not imply strong dominance, the following

theorem provides sufficient conditions under which the implication holds.

Theorem 3.1. Consider the situations where for each (k, j) pair the portions of path $\overline{P}_{kj}$ within the circles centered at grids u and v with radius τ_{h_1} are at most two line segments. Moreover, for one line segment and two line segments with their intersection point not on the circle boundary cases, there are either two or four intersections with the circle. If the intersection point of two segments is on the circle boundary, there are three intersections with the circle (including the intersection point of the two segments). Suppose that grid u weakly dominates grid v for type h_1 . For each (k, j) pair, if one of the following two conditions is satisfied:

- (i) There is no portion of path $\overline{P}_{kj}$ within the circle centered at grid v with radius τ_{h_1} ;
 - (ii) The longer distance from grid u to the lines passing through the two segments (if there is only one segment, the longer distance is the distance itself) is less than or equal to the shorter distance from grid v to the corresponding lines,
- we have that grid u strongly dominates grid v .

Based on the definition of strong dominance, we have the following theorem.

Theorem 3.2. If grid u strongly dominates grid v , $\sum_{h:u \in T^h} x_{hu}^* \geq \sum_{h:v \in T^h} x_{hv}^*$ in an optimal solution.

According to Theorem 3.2, we obtain the following corollaries without proof.

Corollary 3.1. Suppose that the budget can afford to buy at most m detectors of the type with the cheapest unit price. If grid v is strongly dominated by at least m grids, $x_{hv}^* = 0$ for every $h \in \{h : v \in T^h\}$ in an optimal solution.

Corollary 3.2. If there is a grid which strongly dominates all other grids, and moreover the budget allows us to place at least one detector, we will place one detector in that grid in an optimal solution.

Let's consider a special case of Corollary 3.2. Suppose that the port area has only one entrance and several maritime targets, and moreover the entrance grid strongly dominates other grids. Then, we will place one detector in the entrance grid.

Corollary 3.3. Suppose that there is only one detector type b , that is, $|H| = 1$. If grid u strongly dominates grid v , $x_{bu}^* \geq x_{bv}^*$ in an optimal solution.

3.4 Algorithms

Since the proposed [DP] in Section 3.2 is a nonlinear binary integer program, we provide two branch and bound based algorithms (a standard branch and bound algorithm and a linearized branch and bound algorithm) to obtain an optimal solution for a small- or middle-scale problem. For large-scale problems, it is implausible to obtain optimal solutions. So, we propose a greedy adding heuristic which can obtain good solutions in a short time period.

3.4.1 Standard Branch and Bound Algorithm

When we relax the binary constraints for all x_{hi} in [DP], the resulting relaxation problem is as follows:

$$[\text{DP-R}] \quad \min \sum_{k \in E} \sum_{j \in S} W_{kj} \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}} \quad (3.7)$$

$$\text{s.t.} \quad \sum_{h: i \in T^h} x_{hi} \leq 1, \quad \forall i \in T, \quad (3.8)$$

$$\sum_{h \in H} \sum_{i \in T^h} \rho_h x_{hi} \leq M, \quad (3.9)$$

$$0 \leq x_{hi} \leq 1, \quad \forall h, i \in T^h. \quad (3.10)$$

It is clear that [DP-R] is a convex nonlinear program. Hence, we can use a standard branch and bound algorithm to obtain an exact optimal solution for [DP]. The procedure of the standard branch and bound algorithm is described in Chapter 2.

3.4.2 Linearized Branch and Bound Algorithm

For the [DP] formulation, though the standard branch and bound algorithm obtains a solution within optimality tolerance ε , the relaxation problems repeatedly solved during the procedure are nonlinear programs. In this subsection, we propose another detector placement problem formulation which is amenable to linear outer approximation. The idea is borrowed from [87] where they deal with a non-convex continuous program and the relaxation problems are linear programs. The details are given in Chapter 2. Here, we apply the idea to a nonlinear binary

integer program and the associated relaxation problems are mixed integer linear programs.

For each (k, j) pair, we introduce two decision variables y_{kj} and z_{kj} as follows:

$$y_{kj} = \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}} \text{ and } z_{kj} = \ln(y_{kj}) = \sum_{h \in H} \sum_{i \in N_{kj}^h} \ln(1 - p_{hikj}) x_{hi}.$$

For the purpose of algorithm implementation, we need both lower and upper bounds for each y_{kj} , that is, $L_{kj} \leq y_{kj} \leq U_{kj}$. These bounds will be modified during algorithm execution. Define $\mathbf{L}$ and $\mathbf{U}$ as vectors that include all lower bounds L_{kj} and upper bounds U_{kj} , respectively. According to the expression of y_{kj} , the initial bounds are given by:

$$L_{kj}^0 = \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj}) \text{ and } U_{kj}^0 = 1, \quad \forall k \in E, j \in S. \quad (3.11)$$

Based on the introduced new decision variables y_{kj} and z_{kj} , an equivalent formulation for [DP] is stated as follows:

$$[\text{DP}(\mathbf{L}, \mathbf{U})] \quad \min \quad \sum_{k \in E} \sum_{j \in S} W_{kj} y_{kj} \quad (3.12)$$

$$\text{s.t.} \quad \sum_{h: i \in T^h} x_{hi} \leq 1, \quad \forall i \in T, \quad (3.13)$$

$$\sum_{h \in H} \sum_{i \in T^h} \rho_h x_{hi} \leq M, \quad (3.14)$$

$$z_{kj} = \sum_{h \in H} \sum_{i \in N_{kj}^h} \ln(1 - p_{hikj}) x_{hi}, \quad \forall k \in E, j \in S, \quad (3.15)$$

$$z_{kj} = \ln(y_{kj}), \quad \forall k \in E, j \in S, \quad (3.16)$$

$$L_{kj} \leq y_{kj} \leq U_{kj}, \quad \forall k \in E, j \in S, \quad (3.17)$$

$$x_{hi} \in \{0, 1\}, \quad \forall h, i \in T^h. \quad (3.18)$$

We notice that $[\text{DP}(\mathbf{L}, \mathbf{U})]$ is a mixed integer linear program except that constraints (3.16) are nonlinear. To find an effective method to solve $[\text{DP}(\mathbf{L}, \mathbf{U})]$, we relax constraints (3.16) through constructing a polyhedral outer approximation which consists of a convex envelope and three tangential supports. As suggested by [87], we adopt three support points: L_{kj} , $\bar{y}_{kj} = \frac{U_{kj} - L_{kj}}{\ln(U_{kj}) - \ln(L_{kj})}$, and U_{kj} . The middle point $\bar{y}_{kj}$ is the one that minimizes the maximum approximation error. Figure 3.4 shows such a polyhedral outer approximation.

FIGURE 3.4: A polyhedral outer approximation for $z_{kj} = \ln(y_{kj})$ over $[L_{kj}, U_{kj}]$ [87].

The corresponding relaxation problem of $[\text{DP}(\mathbf{L}, \mathbf{U})]$ is as follows:

$$\begin{aligned}
 [\text{DP}(\mathbf{L}, \mathbf{U})\text{-R}] \quad & \min && \sum_{k \in E} \sum_{j \in S} W_{kj} y_{kj} \\
 \text{s.t.} &&& \sum_{h: i \in T^h} x_{hi} \leq 1, && \forall i \in T, \\
 &&& \sum_{h \in H} \sum_{i \in T^h} \rho_h x_{hi} \leq M, \\
 &&& z_{kj} = \sum_{h \in H} \sum_{i \in N_{kj}^h} \ln(1 - p_{hikj}) x_{hi}, && \forall k \in E, j \in S, \\
 &&& z_{kj} \geq \ln(L_{kj}) + \frac{\ln(U_{kj}) - \ln(L_{kj})}{U_{kj} - L_{kj}} (y_{kj} - L_{kj}), && \forall k \in E, j \in S, \\
 &&& z_{kj} \leq \ln(L_{kj}) + \frac{y_{kj} - L_{kj}}{L_{kj}}, && \forall k \in E, j \in S, \\
 &&& z_{kj} \leq \ln(\bar{y}_{kj}) + \frac{y_{kj} - \bar{y}_{kj}}{\bar{y}_{kj}}, && \forall k \in E, j \in S, \\
 &&& z_{kj} \leq \ln(U_{kj}) + \frac{y_{kj} - U_{kj}}{U_{kj}}, && \forall k \in E, j \in S, \\
 &&& L_{kj} \leq y_{kj} \leq U_{kj}, && \forall k \in E, j \in S, \\
 &&& x_{hi} \in \{0, 1\}, && \forall h, i \in T^h.
 \end{aligned}$$

We observe that $[\text{DP}(\mathbf{L}, \mathbf{U})\text{-R}]$ is a mixed integer linear program. Now, we present a linearized branch and bound algorithm to solve $[\text{DP}]$ through repeatedly solving relaxation problems $[\text{DP}(\mathbf{L}, \mathbf{U})\text{-R}]$. While doing so, $\mathbf{L}$ and $\mathbf{U}$ will be updated correspondingly. We define $\mathbf{Y}$ and $\mathbf{Z}$ as vectors that include all elements y_{kj} and z_{kj} , respectively. The procedure of the linearized branch and bound algorithm is as follows:

Algorithm 3.1. Linearized Branch and Bound Algorithm

Step 0. (Initialization) Set stage $s = 0$, active node set $O_s = \{0\}$, current node $n(s) = 0$, and $n = 0$. Let the initial bounds be $\mathbf{L}^0$ and $\mathbf{U}^0$ with elements derived through (4.45). Solve $[\text{DP}(\mathbf{L}^0, \mathbf{U}^0)\text{-R}]$ and obtain an optimal solution $\mathbf{\Gamma}^0 = (\mathbf{X}^0, \mathbf{Y}^0, \mathbf{Z}^0)$ and the corresponding objective function value $V[\text{DP}(\mathbf{L}^0, \mathbf{U}^0)\text{-R}]$. We observe that $\mathbf{X}^0$ is feasible to $[\text{DP}]$. So, we let upper bound UB be the objective function value of the incumbent $\mathbf{X}^0$ evaluated through (3.3). Let ε be some optimality tolerance. If $V[\text{DP}(\mathbf{L}^0, \mathbf{U}^0)\text{-R}] \geq \text{UB}(1 - \varepsilon)$, stop the procedure. Otherwise, go to Step 1.

Step 1. (Division) Split node $n(s)$ into two child nodes $n+1$ and $n+2$ through the following branching rule. From an optimal solution $\mathbf{\Gamma}^{n(s)} = (\mathbf{X}^{n(s)}, \mathbf{Y}^{n(s)}, \mathbf{Z}^{n(s)})$ of $[\text{DP}(\mathbf{L}^{n(s)}, \mathbf{U}^{n(s)})\text{-R}]$, we find indices k' and j' such that

$$|z_{k'j'}^{n(s)} - \ln(y_{k'j'}^{n(s)})| = \max_{k \in E, j \in S} |z_{kj}^{n(s)} - \ln(y_{kj}^{n(s)})|.$$

Then, we split the bound interval $[L_{k'j'}^{n(s)}, U_{k'j'}^{n(s)}]$ into two subintervals: $[L_{k'j'}^{n(s)}, y_{k'j'}^{n(s)}]$ and $[y_{k'j'}^{n(s)}, U_{k'j'}^{n(s)}]$, and add a subinterval for each child node. Update active node

set by $O_s = O_s \cup \{n+1, n+2\} \setminus \{n(s)\}$ and denote the bounds for two child nodes as $(\mathbf{L}^{n+1}, \mathbf{U}^{n+1})$ and $(\mathbf{L}^{n+2}, \mathbf{U}^{n+2})$, respectively.

Step 2. (Bound updating and fathoming) Solve the two relaxation programs: $[\text{DP}(\mathbf{L}^{n+1}, \mathbf{U}^{n+1})\text{-R}]$ and $[\text{DP}(\mathbf{L}^{n+2}, \mathbf{U}^{n+2})\text{-R}]$. Evaluate the $\mathbf{X}$ parts of the solutions through (3.3) and update the incumbent solution and its objective function value UB. If the relaxation program is infeasible, or if the objective function value of the relaxation program is larger than $\text{UB}(1 - \varepsilon)$, we fathom the node. Update O_{s+1} by removing all fathomed nodes from O_s and let $n = n + 2$ and $s = s + 1$.

Step 3. (Termination and node selection) If O_s is an empty set, stop the procedure and return the incumbent. Otherwise, select a node $n(s)$ from O_s with the smallest relaxation objective function value and go to Step 1.

The linearized branch and bound algorithm with $\varepsilon = 0$ possesses a global convergence property. The convergence result is stated in the following theorem.

Theorem 3.3. Consider any infinite branch whose nodes are labeled as $\{n(s)\}$ where s is in some index set N . Let sequence $\{\mathbf{X}^{n(s)}\}$ be the $\mathbf{X}$ -variable parts of the corresponding sequential solutions generated by the linearized branch and bound algorithm with $\varepsilon = 0$. Then, $\{\mathbf{X}^{n(s)}\}$ converges to a global minimizer of $[\text{DP}]$.

3.4.3 Greedy Adding Heuristic

In this subsection, we introduce a greedy adding heuristic to quickly obtain solutions for large-scale problems. The idea is to sequentially choose pairs of (h, i)

such that the total expected damage cost is minimized given that the previously chosen pairs are fixed. The procedure of the greedy adding heuristic is as follows:

Algorithm 3.2. Greedy Adding Heuristic

Step 0. (Initialization) Let Q be the detector placement set. Initially, we set $Q = \emptyset$.

Step 1. (Termination checking) Check if $M < \min_{h \in H} \rho_h$. If yes, stop the procedure. Otherwise, go to Step 2.

Step 2. (Selection) Let $\tilde{H} \subseteq H$ be the set of detector types whose unit purchasing prices are less than or equal to M . Define $\tilde{Q} = \{(h, i) : h \in \tilde{H}, i \in T^h\}$. For each $(h, i) \in \tilde{Q}$, we calculate the corresponding objective function value given that the placement set is $Q \cup \{(h, i)\}$. Then, we determine pair (h', i') such that the calculated objective function value is minimal, that is,

$$(h', i') \in \arg \min_{(h, i) \in \tilde{Q}} \{\text{the objective function value of adding } (h, i) \text{ to } Q\}.$$

Step 3. (Updating) Let $Q = Q \cup \{(h', i')\}$, $T^h = T^h \setminus \{i'\}$ for every h such that $i' \in T^h$, and $M = M - \rho_{h'}$. Then, go to Step 1.

There are special situations where the greedy adding heuristic obtains optimal solutions. For example, if there is only one detector type and none of the sets N_{kj} intersect (that is, $\cap_{kj} N_{kj} = \emptyset$), as shown in [73], the greedy adding heuristic yields optimal solutions.

3.5 Computational Study

In this section, we give a detailed computational study. During the implementation of both exact algorithms (the standard and linearized branch and bound algorithms), we incorporate the strong dominance result in Theorem 3.2 to tighten our formulations. All algorithms are coded in Matlab and run on a PC with Intel Core 2 Duo 3.10GHz CPU and 4GB of RAM. To solve continuous nonlinear programs for the standard branch and bound algorithm, we use Matlab's *fmincon* function with a sequential quadratic programming approach. To solve mixed integer linear programs for the linearized branch and bound algorithm, we run IBM ILOG CPLEX V12.4 in a Matlab environment. The optimality tolerances for both branch and bound based algorithms are set to be 10^{-3} .

The structure of this section is as follows: First, we provide an illustrative example and its associated sensitivity analysis with respect to the budget limit. Second, computational results are provided to investigate the performance of the exact and heuristic algorithms proposed in Section 3.4.

3.5.1 Illustrative Example

In our illustrative example, we focus on a square threat area which is divided into 12×12 grids with each grid size $200 \text{ m} \times 200 \text{ m}$. The sets of entrances, targets, and blocked grids are $\{1, 4, 9, 48, 49, 84, 85, 135, 141\}$, $\{77, 105\}$, and $\{16, 21, 50, 90, 95, 112\}$, respectively. The associated shortest paths are shown in Figure 3.5. We assume

that the terrorist chooses every pair of entrance and target with the same probability, that is, $\gamma_{kj} = \frac{1}{|E||S|}, \forall k, j$. The damage cost for every target is the same with the value of 35 million, that is, $C_j = 35 \text{ million}, \forall j$. The probability θ that the intervention team foils the attack is 0.6. The timely detection time λ is 10 seconds and the speed of the small vessel ψ is 20 m/s. The total budget M for all types of detectors is 200 thousand. We consider three types of detectors where $(\rho_1, \tau_1, \eta_1) = (50, 220, 0.006), (\rho_2, \tau_2, \eta_2) = (40, 200, 0.005)$, and $(\rho_3, \tau_3, \eta_3) = (30, 180, 0.004)$. The unit of ρ_h is in thousands and the unit of τ_h is in meters. All parameter values are summarized in Table 3.1.

TABLE 3.1: Parameter values

Parameter	Description	Value
E	Set of entrances	$\{1, 4, 9, 48, 49, 84, 85, 135, 141\}$
S	Set of targets	$\{77, 105\}$
D	Set of blocked grids	$\{16, 21, 50, 90, 95, 112\}$
γ_{kj}	Probability of choosing path P_{kj}	$\frac{1}{ E S }$
C_j	Damage cost at target j	35 million
θ	Probability of successful intervention	0.6
λ	Timely detection time	10 s
ψ	Speed of the small vessel	20 m/s
M	Total budget	200 thousand
ρ_h	Unit purchasing price for type h	$\rho_1 = 50, \rho_2 = 40, \rho_3 = 30$ thousand
τ_h	Effective detection radius for type h	$\tau_1 = 220, \tau_2 = 200, \tau_3 = 180$ m
η_h	Instantaneous detection rate for type h	$\eta_1 = 0.006, \eta_2 = 0.005, \eta_3 = 0.004$

For the illustrative example, we find 131 pairs of grids such that one strongly dominates the other, for example, grid 1 strongly dominates grids 13, 14, 15, 26, 27, 38, and 51. The optimal solution is $x_{1,65} = x_{1,68} = x_{1,83} = x_{1,88} = 1$ (that is, type 1 detectors are placed in grids 65, 68, 83, and 88) and the corresponding objective function value is 25.980 million. The running CPU time for the standard branch and bound algorithm is 7867.074 seconds, while the CPU time for the linearized one is 1317.041 seconds. The example illustrates that the linearized branch and bound algorithm takes much less time to obtain the optimal solution.

FIGURE 3.5: An illustrative example.

Now, we turn to the sensitivity analysis with respect to the budget limit. When we increase the budget from 100 thousand to 300 thousand gradually, the corresponding optimal objective function values and optimal solutions are shown in Table 3.2. It is intuitive that when the budget increases, the optimal expected damage cost decreases. Moreover, we observe that due to their cost-effectiveness, type 1 detectors have a higher priority to be selected than the other two types.

TABLE 3.2: Sensitivity analysis with respect to the budget limit

Budget	Value	Optimal solution
100.000	29.541	$x_{1,65} = x_{1,81} = 1$
120.000	28.976	$x_{1,81} = x_{2,65} = x_{3,75} = 1$
140.000	28.014	$x_{1,66} = x_{1,81} = x_{2,75} = 1$
160.000	27.372	$x_{1,66} = x_{1,81} = x_{3,75} = x_{3,135} = 1$
180.000	26.680	$x_{1,66} = x_{1,75} = x_{1,81} = x_{3,135} = 1$
200.000	25.980	$x_{1,65} = x_{1,68} = x_{1,83} = x_{1,88} = 1$
220.000	25.396	$x_{1,65} = x_{1,68} = x_{1,83} = x_{2,88} = x_{3,141} = 1$
240.000	24.735	$x_{1,65} = x_{1,68} = x_{1,83} = x_{3,75} = x_{3,135} = x_{3,141} = 1$
260.000	24.093	$x_{1,65} = x_{1,68} = x_{1,75} = x_{1,83} = x_{3,135} = x_{3,141} = 1$
280.000	23.556	$x_{1,65} = x_{1,68} = x_{1,75} = x_{1,83} = x_{1,88} = x_{3,141} = 1$
300.000	23.047	$x_{1,65} = x_{1,68} = x_{1,75} = x_{1,83} = x_{2,141} = x_{3,88} = x_{3,135} = 1$

3.5.2 Computational Results

Here, we continue to consider square threat areas with each grid size $200 \text{ m} \times 200 \text{ m}$. We consider two categories of experiments: small-size test problems and middle-size test problems. For small-size test problems, the dimensions are 12×12 , 13×13 , 14×14 , and 15×15 grids. While for middle-size test problems, the dimensions are 20×20 , 25×25 , 30×30 , and 35×35 grids. For each dimension, the number of entrances may be 3, 6, or 9, the number of targets may be 1, 2, or 3, and the number of blocked grids may be 2, 4, or 6. The corresponding locations for entrances, targets, and blocked grids are randomly generated. In total, we obtain 27 instances for each dimension. Other parameter values are the same as those in the illustrative example.

Computational results for small-size problems are provided in Tables 3.3 and 3.4. In both tables, Dim represents dimension, SBB represents standard branch and bound, LBB represents linearized branch and bound, and GAH represents greedy adding heuristic. GR represents the relative deviation of the greedy heuristic from

an exact algorithm, which is computed by the following formula:

$$\text{GR} = \frac{\text{Heuristic Value} - \text{Exact Value}}{\text{Exact Value}} \times 100\%.$$

From Tables 3.3 and 3.4, we observe that the optimal values of both SBB and LBB match with each other. This is obvious because both are exact algorithms. In most instances, the CPU times of LBB are much shorter than those of SBB. We find two instances for dimension 12×12 , six instances for 13×13 , two instances for 14×14 , and seven instances for 15×15 where LBB CPU times are larger than SBB CPU times. But, for all these instances, both algorithms obtain optimal solutions within six minutes. As for the relative performance of GAH, we observe that in 48 out of 108 instances GAH obtains optimal solutions. The average GR is 1.137% and the maximum GR is 7.765%.

Computational results for middle-size problems are provided in Tables 3.5 and 3.6. Due to the preference of LBB over SBB for small-size problems, only the results for LBB are provided. As for the relative performance of GAH, we observe that GAH obtains optimal solutions in 31 out of 108 instances and the maximum running time among all instances is less than 12 seconds. The average GR is 1.936% and the maximum GR is 8.676%.

TABLE 3.3: Computational results for small-size problems: 12×12 and 13×13

Dim	E	S	D	SBB		LBB		GAH	
				Value	CPU (s)	Value	CPU (s)	Value	GR (%)
12×12	72,132,139	80	59,111	16,023	775.152	16,023	36.417	16,123	0.625
	8,48,135	85	47,102,128,137	16,853	102.035	16,853	16.505	16.853	0.031
	3,108,141	79	26,31,38,99,128,139	18,389	100.322	18,389	9.182	18.525	0.739
	8,84,144	36,86	58,115	15,704	269.532	15,704	100.877	15.836	0.031
	1,60,120	31,123	33,35,57,101	16,251	1520.866	16,251	203.273	16.274	0.140
	2,60,96	57,128	26,29,30,34,74,121	15,957	129.769	15,957	90.306	15.957	0.031
	60,97,132	5,36,139	54,63	15,995	251.096	15,995	170.266	15.995	0.000
	7,48,140	57,94,131	53,61,62,129	19,024	35.870	19,024	20.291	19.024	0.031
	5,120,137	8,63,125	53,66,99,104,108,115	16,868	320.453	16,868	140.680	16.908	0.062
	3,37,96,108,120,139	125	33,85	15,524	66.584	15,524	44.880	15.700	0.031
	24,48,84,133,134,137	101	35,55,96,105	19,388	13.533	19,388	35.359	19.388	0.016
	2,3,11,85,132,134	50	1,27,40,110,121,128	17,078	39.498	17,078	27.133	17.078	0.031
	5,72,84,136,137,141	93,96	132,140	18,208	328.471	18,208	79.712	18.421	1.172
	5,72,134,136,139,140	24,73	16,18,26,124	21,606	476.305	21,606	275.510	21.841	0.125
	3,12,72,133,139,144	45,130	26,80,99,119,141,142	23,985	1616.113	23,985	300.808	24.129	0.604
	8,25,96,132,136,141	59,61,125	15,100	20,331	5580.966	20,331	1947.719	21.497	0.203
	5,11,97,135,137,143	68,91,107	12,42,124,129	23,980	590.020	23,980	587.703	24.481	0.172
	7,72,96,109,135,138	1,31,44	19,22,23,58,68,87	22,980	155.074	22,980	311.824	22.980	0.187
	3,12,13,25,49,61,85,138,143	114	39,100	18,199	1135.613	18,199	69.831	18.808	0.062
	11,12,13,24,36,49,72,137,138	70	18,41,127,143	20,868	33.467	20,868	30.684	20.868	0.047
	8,9,10,13,24,109,134,135,140	52	26,48,50,73,94,102	19,842	1269.117	19,842	110.971	19.876	0.062
	1,4,6,11,37,109,132,133,141	49,81	64,129	24,677	5444.539	24,677	817.478	24.702	0.103
	1,9,10,11,49,109,121,132,142	19,105	86,118,128,137	20,958	2864.878	20,958	510.012	21.466	0.234
	1,5,85,108,120,121,134,135,139	61,69	16,60,86,99,101,114	23,632	4227.774	23,632	820.818	23.753	0.172
	5,6,48,61,72,97,120,138,139	47,50,106	89,104	21,727	16023.899	21,727	2541.104	22.053	0.296
	6,48,60,72,97,133,135,138,140	70,136,141	29,61,96,120	20,022	766.308	20,022	438.222	20.329	0.218
	3,6,11,36,72,132,134,139,144	59,84,137	70,89,109,122,129,131	23,470	1926.104	23,470	1535.137	23.470	0.296
13×13	157,160,163	124	52,87	14,740	13.829	14,740	25.882	14.753	0.091
	11,117,167	67	33,134,144,151	14,683	59.915	14,683	28.139	14.882	0.016
	6,105,118	90	2,19,57,77,84,112	14,335	55.702	14,335	19.505	14.335	0.016
	91,166,168	61,142	72,132	15,858	3.413	15,858	41.398	15.858	0.031
	9,78,161	10,124	31,36,98,128	17,042	446.370	17,042	97.041	17.083	0.031
	12,39,131	62,93	7,54,55,91,98,138	15,864	517.295	15,864	79.706	15.915	0.016
	40,165,168	9,67,79	11,66	15,816	83.411	15,816	93.296	15.816	0.078
	3,92,164	50,77,123	34,95,101,159	16,117	655.498	16,117	193.964	16.272	0.094
	6,9,162	15,30,161	4,26,27,52,114,129	15,893	10.170	15,893	69.074	15.977	0.047
	4,14,27,79,165,169	101	39,121	15,987	134.824	15,987	33.596	15.987	0.047
	39,130,144,158,164,165	36	3,51,123,137	15,907	363.410	15,907	89.948	15.907	0.047
	3,11,66,143,157,169	106	7,57,82,92,117,135	16,936	201.011	16,936	47.852	16.936	0.062
	9,26,92,105,117,162	45,109	7,113	18,871	1518.490	18,871	412.507	19.351	0.094
	2,6,65,161,163,166	7,156	52,54,77,96	18,253	130.943	18,253	76.237	18.664	0.140
	2,14,40,78,117,131	21,43	16,30,41,82,114,164	22,668	19.182	22,668	42.181	22.668	0.109
	7,9,92,117,118,144	32,148,151	11,25	19,594	6673.649	19,594	934.747	19.634	0.203
	4,11,26,53,165,169	28,135,147	77,98,116,157	18,925	2057.123	18,925	1364.440	19.160	0.265
	1,92,118,156,162,164	33,81,148	21,37,69,112,113,124	18,777	1660.410	18,777	325.547	18.777	0.203
	3,4,5,9,10,11,13,65,160	104	20,86	15,595	34.235	15,595	102.719	16.030	0.047
	14,26,27,78,79,118,130,162,165	71	20,47,63,65	19,067	109.976	19,067	39.182	20.548	0.094
	1,6,39,52,53,65,92,144,164	15	32,85,96,122,155,165	17,686	600.288	17,686	33.264	17.686	0.078
	9,13,26,65,91,131,156,161,165	20,61	8,97	23,012	661.465	23,012	441.235	23.298	0.218
	3,8,12,39,130,159,161,164,166	123,144	16,81,136,153	22,293	628.885	22,293	492.069	22.293	0.296
	1,4,7,13,14,53,143,159,163	125,133	74,85,86,93,126,157	20,844	1184.955	20,844	1472.953	21.878	0.250
	3,7,8,9,79,92,156,159,163	33,142,167	73,130	20,628	5336.797	20,628	1861.998	20.628	0.374
	2,8,26,66,91,117,144,159,160	73,93,165	19,87,140,163	26,349	7081.304	26,349	1794.156	26.349	0.499
	5,6,12,78,159,163,166,168,169	40,47,142	31,62,77,101,105,156	23,233	3106.415	23,233	975.715	23.233	0.515

TABLE 3.4: Computational results for small-size problems: 14×14 and 15×15

Dim	E	S	D	SBB		LBB		GAH	
				Value	CPU (s)	Value	CPU (s)	Value	GR (%)
14×14	7,11,99	53	17,37	15,052	108,939	15,052	33,263	15,052	0.000
	4,9,85	147	16,111,130,182	14,720	2,800	14,720	21,841	14,720	0.016
	3,9,188	154	7,21,53,58,114,193	14,697	252,673	14,697	30,195	14,697	0.016
	70,71,183	9,155	36,167	16,306	833,852	16,306	117,436	16,306	0.047
	184,186,195	117,154	36,50,64,106	13,236	35,631	15,236	81,370	15,443	1.361
	6,56,188	20,151	7,22,44,132,169,170	17,171	467,636	17,171	110,161	17,171	0.047
	28,182,190	4,36,169	115,122	15,914	1575,392	15,914	316,186	16,036	0.765
	3,56,113	100,141,147	70,91,105,175	16,205	225,307	16,205	132,365	16,205	0.000
	15,57,169	76,95,184	38,86,151,168,187,192	15,751	277,037	15,751	116,761	16,094	0.062
	6,13,14,85,183,188	70	60,114	16,440	208,823	16,440	88,428	16,735	1.790
	43,56,85,127,188,189	104	13,67,143,165	17,312	73,537	17,312	45,251	17,312	0.031
	1,3,6,112,187,195	162	21,56,70,138,150,151	16,944	561,907	16,944	45,364	16,944	0.047
	3,42,99,108,186,187	11,189	65,66	19,182	4408,313	19,182	483,802	19,932	3.910
	2,8,14,98,126,169	3,171	65,104,136,161	19,305	1023,131	19,305	302,783	19,814	2.640
	9,14,56,57,140,182	105,112	68,102,138,146,165,176	22,971	1804,479	22,971	334,874	22,998	0.125
	5,71,84,112,127,193	62,95,145	108,159	20,341	1119,694	20,341	994,566	21,542	5.906
	57,127,184,185,192,196	2,18,27	23,33,56,65	17,723	51154,304	17,723	411,577	17,816	0.524
	4,15,113,154,169,190	33,48,193	23,38,64,100,104,150	24,068	17720,082	24,068	2361,321	24,206	0.572
	1,4,5,28,70,154,168,185,193	178	69,191	18,870	3800,152	18,870	164,316	19,281	2.175
	6,12,15,71,98,99,185,188,190	48	53,69,90,128	19,554	445,808	19,554	122,154	19,867	1.601
	6,9,28,57,84,154,155,187,192	172	23,24,29,87,93,137	18,138	767,713	18,138	71,157	18,735	0.140
	2,5,14,28,155,182,189,190,193	4,85	16,119	22,864	4884,007	22,864	1367,643	22,919	0.241
	2,11,12,57,113,169,182,189,196	97,179	14,54,109,151	19,863	1827,174	19,863	331,710	19,863	0.421
	5,6,7,84,99,169,191,192,194	114,164	71,74,111,123,176,177	24,154	1547,033	24,154	597,161	24,154	0.000
	11,12,141,182,188,189,190,195,196	62,69,170	45,184	18,354	4894,198	18,354	576,444	18,598	0.452
	11,14,15,56,99,113,127,185,193	31,181,191	43,71,87,154	20,990	42577,062	20,990	367,363	21,805	3.885
	4,5,6,11,28,141,188,194,196	99,173,190	13,104,133,135,153,175	20,577	18695,649	20,577	3254,772	21,074	2.418
15×15	166,210,215	71	218,221	14,894	22,499	14,894	24,824	14,894	0.031
	15,30,195	147	13,79,86,221	14,481	10,932	14,481	13,434	14,481	0.000
	45,75,181	200	11,40,60,70,138,170	14,576	13,040	14,576	22,564	14,576	0.016
	3,150,214	78,135	105,158	17,004	186,147	17,004	66,310	17,004	0.062
	13,16,136	25,207	28,38,110,119	18,131	752,086	18,131	61,959	18,131	0.094
	3,165,219	136,147	10,14,19,67,83,216	19,112	101,608	19,112	30,075	19,112	0.062
	10,165,180	25,132,148	142,162	15,411	12,727	15,411	63,379	15,411	0.016
	5,212,216	4,42,124	70,156,162,172	16,665	1454,017	16,665	233,259	16,829	0.985
	4,12,76	9,92,181	56,58,93,103,106,214	16,002	21,309	16,002	52,756	16,002	0.062
	60,106,165,195,215,221	159	101,169	18,984	313,635	18,984	57,718	20,317	0.047
	6,8,10,60,91,225	18	25,57,67,193	15,665	13,897	15,665	54,234	16,660	6.349
	90,105,136,216,218,225	41	39,54,70,83,176,187	19,302	30,436	19,302	25,407	19,302	0.078
	4,6,75,180,195,215	54,208	16,140	18,135	1083,082	18,135	166,914	18,135	0.000
	3,14,15,30,135,219	29,109	9,43,133,207	23,968	664,753	23,968	166,803	24,681	2.973
	7,210,213,214,215,223	41,183	13,58,93,117,154,186	21,202	130,351	21,202	183,164	21,202	0.172
	11,14,150,195,214,221	56,142,220	48,139	20,629	6968,771	20,629	2226,363	22,197	0.203
	1,3,105,211,216,219	138,149,205	37,124,151,208	19,441	2103,693	19,441	434,739	19,612	0.343
	1,5,106,165,210,221	114,116,148	101,120,156,168,193,214	18,263	800,127	18,263	443,893	18,737	2.596
	6,7,9,75,105,212,214,215,220	3	148,157	15,400	681,917	15,400	158,080	15,400	0.000
	7,12,91,150,151,165,166,218,225	99	26,71,190,204	23,062	248,298	23,062	119,566	23,062	0.109
	3,10,14,45,46,218,220,221,222	208	6,18,41,66,166,187	16,363	223,093	16,363	98,096	16,363	0.000
	1,3,31,45,46,91,150,214,216	48,110	130,225	21,402	8634,507	21,402	472,322	21,647	1.146
	6,150,151,211,213,216,219,222,223	198,225	45,61,71,127	19,292	4860,170	19,292	621,923	19,740	2.323
	1,4,5,8,10,166,216,221,224	45,223	14,31,63,83,87,126	18,849	13323,602	18,849	1313,895	19,563	0.484
	46,60,75,105,180,215,216,221,222	88,110,196	106,178	20,784	51167,478	20,784	8438,560	21,187	1.938
	8,14,45,105,181,213,215,221,222	61,141,220	96,127,214,223	24,981	43098,449	24,981	4019,433	24,981	0.000
	14,16,135,150,213,214,219,222,224	168,176,211	20,27,69,73,83,205	24,168	18191,604	24,168	1752,683	24,168	0.562

TABLE 3.5: Computational results for middle-size problems: 20×20 and 25×25

Dim	E	S	D	LBB		GAH	
				Value	CPU (s)	Value	GR (%)
20×20	81,180,385	328	185,323	16,014	42.158	16,258	0.031
	41,80,391	246	59,240,355,371	16,014	51.781	16,258	1.521
	80,200,382	137	79,120,190,257,349,386	16,014	38.490	16,187	1.081
	360,388,395	350,398	4,46	15,682	51.969	15,752	0.016
	61,260,383	129,198	21,52,60,127	16,030	126.238	16,259	0.078
	12,61,101	67,100	192,208,246,255,294,315	15,257	49.255	15,503	0.047
	12,40,80	17,29,373	158,229	15,323	130.322	15,323	0.000
	120,383,392	110,118,295	11,13,200,242	16,157	133.502	16,310	0.218
	12,15,280	58,192,240	134,142,163,164,203,295	16,020	254.363	16,020	0.078
	2,3,20,381,384,395	255	22,374	17,178	37.538	18,537	0.109
	17,80,81,200,321,381	78	175,190,267,298	18,300	24.066	19,168	0.109
	9,18,81,100,386,392	193	93,147,174,220,298,356	18,090	51.158	18,090	0.000
	18,21,180,201,393,395	15,32	86,189	19,787	657.357	20,670	0.328
	2,14,121,201,320,399	177,190	24,228,291,354	24,184	296.224	24,184	0.000
	16,20,241,320,381,389	106,143	4,27,80,108,319	17,791	385.649	18,000	0.328
	1,3,221,321,386,388	16,60,359	140,152	18,045	1204.516	19,033	1.248
	17,18,81,380,385,388	56,95,238	62,129,327,377	19,290	1273.457	19,739	0.686
	13,221,361,385,389,397	167,172,183	34,111,145,188,255,299	20,540	3657.791	21,016	0.546
	9,120,160,320,321,380,384,388,393	22	122,232	15,643	70.556	15,764	0.468
	5,12,320,341,361,383,385,394,395	227	32,279,286,384	17,152	91.700	17,152	0.172
	140,180,181,321,386,388,389,397,399	354	49,162,209,219,325,336	16,872	57.141	16,872	0.109
25×25	5,20,40,80,201,261,341,381,388	25,365	220,367	24,141	139.347	24,547	0.686
	9,140,180,260,382,385,388,390,393	169,297	29,244,294,376	23,023	393.418	23,360	0.640
	8,15,20,100,121,261,281,381,390	81,119	111,118,120,211,343,384	19,049	203.123	19,049	0.000
	17,80,141,260,341,360,387,393,394	54,178,284	324,340	25,711	2559.565	26,061	1.825
	8,15,17,20,180,240,281,387,397	258,302,325	33,85,158,264	22,011	3670.307	22,590	1.888
	6,7,12,18,20,81,120,390,400	55,182,197	21,38,92,93,152,275	25,901	1881.195	26,057	1.700
	15,201,551	312	79,158	16,015	48.354	16,119	0.031
	151,275,604	190	219,337,347,438	16,015	58.844	16,247	0.031
	21,225,605	265	63,101,305,501,526,620	15,787	22.033	15,823	0.047
	6,151,610	18,389	435,514	15,965	223.191	16,444	0.156
	19,450,605	132,298	210,231,571,578	16,282	197.012	16,282	0.187
	51,615,618	249,254	5,63,70,256,307,452	16,030	198.039	16,261	0.172
	10,375,500	344,355,441	450,469	16,083	308.430	16,263	0.250
	3,550,613	156,241,247	17,341,399,562	19,766	125.172	19,766	0.452
	3,4,300	299,537,553	177,291,436,463,543,589	15,715	114.717	15,715	0.390
	25,151,201,301,602,621	163	320,510,556,601	17,389	109.730	18,063	0.172
	176,225,251,451,606,624	309	57,216,302,429,511,518	17,241	100.931	17,462	0.187
	15,126,300,400,614,622	182,573	160,473	17,414	34.316	18,852	0.172
	4,175,401,476,610,611	13,524	61,128,380,549	18,734	728.586	19,264	0.780
	50,175,376,401,426,613	128,287	14,24,59,313,317,356	17,442	349.204	18,112	0.671
	11,51,280,251,275,601	176,195,439	517,518	23,668	287.136	23,668	0.468
	11,51,76,225,476,525	124,141,402	227,414,515,544	19,165	630.217	20,502	1.529
	2,13,26,300,475,602	10,439,556	21,130,307,371,455,571	19,449	762.117	20,130	1.888
	50,326,351,426,526,625	467	102,403	19,034	1008.547	19,678	1.654
	11,12,23,151,251,275,400,605,614	166	80,227,469,591	18,569	136.671	18,833	0.406
	19,526,551,604,607,608,609,619,621	552	73,76,112,228,297,306	15,582	176.828	15,582	0.328
	25,50,226,475,476,550,605,608,613	122,460	23,26	16,525	83.982	17,939	0.437
	4,7,15,18,19,25,51,450,621	320,440	81,93,169,254	19,829	286.342	20,747	1.732
	9,19,21,226,251,300,450,605,606	36,205	7,62,102,193,475,490	25,599	880.312	25,683	1.092
	5,200,375,501,612,614,621,622,623	153,199,570	198,447	19,041	1258.734	19,400	1.888
	15,18,125,450,525,613,614,621,625	412,587,618	3,100,227,462	22,501	9043.404	22,791	3.276
	16,17,19,51,326,425,450,526,604	121,351,483	70,118,153,168,253,517	22,712	9001.400	22,970	2.792
	1,7,9,10,12,526,551,619,624			20,357	1642.575	20,909	3.307

TABLE 3.6: Computational results for middle-size problems: 30×30 and 35×35

Dim	E	S	D	LBB		GAH	
				Value	CPU (s)	Value	CPU (s)
30×30	60,180,882	495	507,572	14,578	35,281	14,578	0.047
	16,541,871	345	159,209,260,660	14,973	35,014	14,973	0.000
	181,721,880	280	28,97,193,320,836,869	15,226	36,483	15,226	0.000
	23,570,630	357,443	349,893	14,945	83,675	14,945	0.047
	120,890,892	387,689	110,433,674,836	15,455	145,378	15,597	0.094
	17,61,631	175,255	64,228,431,529,727,780	16,037	137,113	16,258	1.376
	30,877,900	121,567,880	231,737	16,360	330,327	16,360	0.983
	4,26,571	12,50,830	300,406,418,727	16,312	297,993	16,312	0.577
	690,691,781	176,438,752	114,192,226,266,571,696	16,429	149,664	16,429	0.406
	1,10,271,690,721,876	50	444,713	16,212	69,919	16,212	0.328
	1,17,30,510,660,895	568	64,338,592,805	18,180	41,652	18,946	0.172
	120,330,511,870,883,895	131	154,259,566,689,705,869	16,151	16,666	16,803	4.213
	24,90,120,690,840,870	144,298	264,445	16,526	214,836	16,526	4.036
	1,24,360,721,840,890	353,433	287,470,658,861	20,616	837,639	21,002	0.156
	2,12,211,301,780,898	694,737	139,461,682,777,819,896	17,626	246,488	18,239	1.170
	8,210,240,480,875,891	56,152,254	24,612	19,540	1574,229	20,876	1.061
	5,8450,872,873,899	154,493,500	375,432,439,677	19,233	898,796	20,782	3.104
	4,17,29,90,721,889	423,434,519	253,272,338,377,514,773	20,364	3027,228	21,317	8.057
	15,28,361,630,780,841,880,886,891	532	511,760	18,867	65,043	19,363	3.658
	18,20,121,361,481,811,874,894,895	160	528,599,810,869	17,780	127,739	17,913	2.629
	4,8,9,24,61,240,331,391,691	284	53,123,482,589,605,708	21,358	76,275	21,358	0.421
	90,451,481,541,571,870,881,890,893	287,679	383,698	20,706	992,294	21,172	0.343
	4,6,8,91,391,451,570,750,881	455,682	387,423,505,661	24,012	968,134	24,161	1.810
	10,61,210,330,420,421,690,840,897	294,879	245,281,289,298,784,874	21,089	475,532	21,089	2.247
	11,19,21,270,271,630,780,889,890	257,276,536	274,404	23,201	7051,625	23,472	0.620
	3,9,19,60,151,541,879,883,896	124,561,643	73,441,475,542	27,085	7733,953	27,652	1.888
	12,27,28,60,841,873,884,887,891	18,501,513	87,141,287,370,531,651	24,910	833,841	24,910	4.165
35×35	13,30,34	293	285,305	14,494	26,263	14,494	2.092
	350,735,805	601	545,599,640,1215	14,098	10,905	14,098	0.000
	4,16,210	1139	117,404,576,606,709,836	14,517	29,953	14,517	0.000
	14,1121,1200	692,826	396,732	15,776	242,465	15,792	0.078
	34,1015,1202	1115,1193	374,408,432,980	16,064	256,835	16,925	0.218
	23,946,1016	104,674	86,239,506,769,984,1010	15,018	242,494	15,018	0.102
	29,315,490	78,450,1157	110,853	16,165	849,906	16,237	5.360
	70,1208,1212	836,988,1029	252,322,702,800	18,069	271,526	18,069	0.000
	2,6,24	639,644,785	401,655,672,842,855,893	15,784	336,767	16,056	0.447
	6,456,666,1050,1217,1218	696	208,279,822,936	14,726	50,290	14,726	0.562
	11,26,30,280,771,1015	1158	97,450,530,626,685,728	16,994	65,721	16,994	1.725
	1,140,210,770,1195,1198	595,803	177,739	16,135	146,320	16,461	0.000
	15,175,315,1120,1212,1221	176,995	246,258,567,1034	18,360	533,294	19,561	0.234
	23,30,386,665,910,1216	588,622	236,345,444,812,921,940	18,966	413,065	20,033	6.539
	32,140,666,735,1210,1224	439,801,812	85,477	18,705	248,169	19,364	5.629
	4,16,31,71,1219,1220	39,162,200	434,550,900,1055	23,958	716,990	24,486	2.387
	7,17,70,631,700,1219	557,558,739	46,53,234,355,437,516	20,516	1127,391	22,018	3.522
	16,20,175,211,490,525,1192,1200,1219	693	775,1209	18,981	223,781	19,032	7.319
	30,32,246,806,1086,1193,1197,1215,1223	221	87,255,810,1156	17,024	150,842	17,181	1.059
	2,3,9,596,665,1015,1085,1198,1210	571	409,420,505,1138,1157,1225	18,190	109,485	18,849	0.269
	105,106,175,210,385,420,911,946,1016	38,1075	727,976	18,141	1421,988	18,517	0.920
	10,16,70,245,246,1201,1204,1211,1212	27,993	140,353,411,443	24,504	1722,755	25,028	3.625
	19,24,29,32,34,141,1086,1212,1223	214,569	172,179,561,951,1140,1161	19,093	673,552	19,728	2.072
	8,13,70,281,456,526,1205,1210,1218	373,881,1107	606,761	23,603	4110,776	24,277	3.139
	6,22,23,36,525,595,806,1194,1222	52,634,897	699,716,830,978	25,616	2901,040	25,731	3.292
	5,14,27,28,30,490,806,1016,1193	252,381,1111	328,411,735,1103,1138,1205	22,647	5592,256	23,579	9.251
							10.421
							11.591
							4.117

Chapter 4

Game-theoretic Detector

Deployment under a Small Vessel

Attack Threat to Port Security

4.1 Introduction

In Chapter 3, we make some contributions to port security by providing a model and algorithms to protect ports from a small boat attack threat. However, it is important to note that there is a basic assumption that the terrorist is blind with respect to the locations of detectors in the model. The assumption makes our mathematical model easier, but it is not realistic because the terrorist is intelligent as exemplified by the tragic events of September 11, 2001. In fact, as

presented in Chapter 1, there are many terrorist organizations that have demonstrated maritime capabilities and far-reaching strategies. Moreover, they could potentially take advantage of the weaknesses of ports and add maritime capabilities to strategically launch attacks on critical infrastructures to cause massive disruptions. Hence, in this chapter, we relax the assumption to address an important challenging problem. The problem is how to locate detectors to minimize the maximum expected damage caused by an intelligent terrorist group who can get full information of the locations of detectors.

As explained in Chapter 2, game theory is naturally used to deal with the strategic behavior of adversary players. Especially, Stackelberg game theory, a branch of game theory, is widely applied to homeland security problems. A Stackelberg game is a sequential game with two players: a leader and a follower. Furthermore, in the first place, the leader makes his decisions. In the second place, the follower makes his optimal responses after observing the leader's decisions. These two decision makers have different or even conflicting objectives. Many homeland security problems have been formulated as Stackelberg games where the leader is a government and the follower is a terrorist group. For example, a Stackelberg game model is developed for scheduling air marshals [75]. Hence, Stackelberg games are appropriate to model our problem.

In this chapter, we continue to focus on small vessel attacks on maritime targets, where small vessels are defined as waterborne improvised explosive delivery devices and maritime targets are high-value assets in ports such as military vessels, cruise

ships, passenger ferries, oil tankers, liquefied natural gas carriers, and other critical infrastructures. We formulate this problem as a defender-attacker Stackelberg game with two players: a leader (a government) and a follower (a terrorist group). The government first decides a detector deployment strategy to minimize the maximum expected damage while keeping the cost of detectors within a budget limit. The terrorist group, after obtaining the full information of the detector locations via intelligence, decides the small boat attacking strategy (that is, determines its attacking routes) with the objective of maximizing the expected damage.

In this work, there are three major contributions. First, we not only consider the evasion probabilities of attackers but also the values of targets. Second, we formulate the problem as a min-max nonlinear integer program which can lead to an optimal solution for the problem. Third, we propose two exact algorithms to get optimal solutions and a heuristic to obtain good solutions in a short period of time.

The chapter is organized as follows. In Section 4.2, we state our model. In Section 4.3, two cutting-plane-based exact algorithms and one greedy heuristic for the defender-attacker model are provided. Finally, in Section 4.4, we give a detailed computational study.

4.2 Problem Modeling

4.2.1 Problem Description

In this research, we study a game-theoretic detector deployment problem to protect a set of costly assets in a port, such as military vessels, cruise ships, oil tankers, liquefied natural gas carriers, and other critical infrastructures, from a small vessel attack threat. We continue to consider a rectangular-shaped threat area in a port which is broken down into $m \times n$ grids. The threat area is characterized by three elements: entrances, targets, and obstructions. We denote by E the set of entrances along the boundary of the area from which the terrorists would enter into the concerned area. Let S be the set of known potential stationary targets that the intelligent terrorist group would prefer to attack. Different targets may have different values. For each target $j \in S$, damage caused by one successful small boat attack is denoted by C_j . In the work, we assume that the damage caused by multiple small vessels is additive. We also assume that all entrances and targets are positioned in the centers of their corresponding grids. We define D as the set of obstruction grids through which a small boat cannot travel. An illustrative example is provided in Figure 4.1 where the shaded grids are blocked grids, the grids with a triangle are entrances, and the grids with a star represent targets. In this example, $D = \{13, 14, 18, 39, 40, 52\}$, $E = \{32, 33, 58, 62\}$, and $S = \{12, 29\}$.

In our game model, there are two players: a government (a leader) and a terrorist group (a follower). The government decides the locations of detectors taking

FIGURE 4.1: An illustrative example for a threat area and the associated network.

into account the terrorist group's strategy. We assume that all grid centers are possible detector locations. Detectors are characterized by an effective detection radius τ and an instantaneous detection rate η . The terrorist group determines how many small vessels are utilized and an optimal trajectory for each small vessel under the assumption that the full information about the locations of detectors is obtained. It is significantly important and challenging to construct the terrorist group's strategy, specifically, the trajectories of small vessels since there are no strict paths in a port area. Inspired by Brown et al. [14], we reduce this problem

by constructing a network on our mesh grids where there is an arc between any horizontal, vertical, or diagonal adjacent grids except block grids. This new network is also shown in Figure 4.1. The path network is denoted by $\mathcal{G}(G, A)$, where $G = m \times n$ represents the grid node set and A represents the arc set. We assume that a small vessel can travel any arc in the network from an entrance toward a target.

To allow the defender to have enough time to interdict the terrorist group after detection, we define timely detection as detection such that there are at least λ seconds remaining before the small vessel reaches the target. We assume that the speed of a small vessel is ψ m/s, and this time period can be converted to a distance of $\lambda\psi$ meters away from the target. Now, we can estimate a timely detection probability for each arc. Let p_{ikj} be the probability that a detector placed on grid $i \in G$ timely detects a small vessel traveling on arc k to target j . To calculate p_{ikj} , we use the formula in [82], that is,

$$p_{ikj} = 1 - e^{-\eta l_{ikj}},$$

where η is a detector's instantaneous detection rate and l_{ikj} is the length of the portion of arc k which is within τ meters from grid i excluding the length of the portion of arc k which is within $\lambda\psi$ meters from target j . We provide an example shown in Figure 4.2, where the grid size is 200×200 , $\lambda = 5$, $\psi = 20$, $\tau = 200$, and arc k is presented by its endpoints. Based on the given data, we get $l_{28,2936,29} = 182.843$. If $\eta = 0.006$, $p_{28,2936,29} = 0.666$.

Define N_{kj} as the set of all grids from which a detector can timely detect a small boat on arc k to target j . We can use a simple geometrical method to get N_{kj} : finding all grids i whose smallest distance to arc k is less than τ and the first detection point on arc k is at least $\lambda\psi$ meters from target j . Continuing the example in Figure 4.2, $N_{2936,29} = \{28, 29, 36, 37\}$.

FIGURE 4.2: An example for N_{kj} and l_{ikj} .

Let $T = \bigcup_{kj} N_{kj}$ be the collection of all potential locations for detectors. We assume that there is at most one detector for each grid and a budget can afford at most M detectors. For each $i \in T$, we define a binary decision variable x_i which

takes a value of 1 if there is a detector placed in grid i and takes a value of 0 otherwise.

In our model, considering maximizing the expected damage, the terrorist group firstly estimates the probability of evasion when traversing a path. We assume that detectors work independently. Let R_j be the set of paths which can lead to target j . For a given path $r \in R_j$, the evasion probability that a single small vessel successfully passes through path r is the product of all evasion probabilities of the arcs along this path, that is,

$$\prod_{k \in r} \prod_{i \in N_{kj}} (1 - p_{ikj})^{x_i}.$$

Then, the terrorist group decides a path for each small boat since there are B small boats. Let $R = \bigcup_j R_j$ be the set of paths which lead to a target. We define another binary decision variable y_r which equals to 1 when path $r \in R$ is chosen by the terrorist group and equals to 0 otherwise. We assume that each arc can be occupied by at most one small boat.

Now, we focus on the objective functions of the two players. We assume that when a detector timely detects a small boat, the defender will take actions to stop the attack with a successful probability θ . Therefore, there are two cases that cause damage using path $r \in R_j$. One is that the small vessel is undetected with damage

$$C_j \prod_{k \in r} \prod_{i \in N_{kj}} (1 - p_{ikj})^{x_i}.$$

The other is that the intervention actions fail to stop the attack, which results in the damage

$$C_j \left(1 - \prod_{k \in r} \prod_{i \in N_{kj}} (1 - p_{ikj})^{x_i} \right) (1 - \theta).$$

Hence, the resulting expected damage is

$$\sum_j C_j \sum_{r \in R_j} y_r \left(\prod_{k \in r} \prod_{i \in N_{kj}} (1 - p_{ikj})^{x_i} + \left(1 - \prod_{k \in r} \prod_{i \in N_{kj}} (1 - p_{ikj})^{x_i} \right) (1 - \theta) \right). \quad (4.1)$$

Rearranging (4.1), we get

$$\sum_j C_j \sum_{r \in R_j} y_r \left(\prod_{k \in r} \prod_{i \in N_{kj}} (1 - p_{ikj})^{x_i} \theta + (1 - \theta) \right). \quad (4.2)$$

In order to ease our notations, each path $r \in R$ is associated with target r and damage value C_r such that (4.2) can be rewritten as follows:

$$\sum_{r \in R} C_r y_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right). \quad (4.3)$$

For greater clarity, we summarize all the parameters in Table 4.1.

TABLE 4.1: Parameters

Parameter	Description
E	Entrance set
S	Target set
D	Blocked grid set
R	Path set
R_j	Set of paths leading to target $j \in S$
C_j	Damage cost at target j per small boat attack
λ	Timely detection time
p_{ikj}	Probability that a detector placed on grid i timely detects a small vessel traveling on arc k to target j
M	Total number of detectors
θ	Probability of successful interdiction if an attack is detected
B	Total number of small vessels
T	Potential detector locations
N_{kj}	Potential detector locations for arc k which leads to target j
ψ	Speed of a small vessel

4.2.2 Problem Formulation

For the ease of exposition, in the rest of this chapter, we use vector $X = (x_1, x_2, \dots, x_{|T|})$ and $Y = (y_1, y_2, \dots, y_{|R|})$ to represent a defense plan and an attacking strategy, respectively.

Firstly, we present the attacker's model for a given defense plan $\bar{X}$ as follows:

$$[\text{AS}] \quad \max_Y \quad \sum_{r \in R} C_r y_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{\bar{x}_i} \theta + (1 - \theta) \right) \quad (4.4)$$

$$\text{s.t.} \quad \sum_{r \in R} y_r \leq B, \quad (4.5)$$

$$\sum_{r \in R} a_{rk} y_r \leq 1, \quad \forall k \in A, \quad (4.6)$$

$$y_r \in \{0, 1\}, \quad \forall r \in R. \quad (4.7)$$

The objective function (4.4) is to maximize the expected damage of all small boat attacks. Constraint (4.5) ensures that the number of chosen paths cannot be larger than that of small boats. Constraints (4.6) mean that for each arc there is at most one small boat passing through it, where a_{rk} is a constant that indicates whether arc k is involved in path r . Constraints (4.7) are binary constraints. In the attacker's model [AS], Y are decision variables and other terms are constants.

Let $L_r = C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{\bar{x}_i} \theta + (1 - \theta) \right)$, we have the following program:

$$[\text{EAS}] \quad \max_Y \quad \sum_{r \in R} L_r y_r \quad (4.8)$$

$$\text{s.t.} \quad \sum_{r \in R} y_r \leq B, \quad (4.9)$$

$$\sum_{r \in R} a_{rk} y_r \leq 1, \quad \forall k \in A, \quad (4.10)$$

$$y_r \in \{0, 1\}, \quad \forall r \in R, \quad (4.11)$$

which is a binary linear program.

Once knowing attacking plan $\bar{Y}$, that is, knowing attacking path set $\bar{R}$, the defender's detector deployment strategy can be expressed by the following binary nonlinear program (the defender's model):

$$[\text{DS}] \quad \min_X \quad \sum_{r \in \bar{R}} C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right) \quad (4.12)$$

$$\text{s.t.} \quad \sum_{i \in T} x_i \leq M, \quad (4.13)$$

$$x_i \in \{0, 1\}, \quad \forall i \in T. \quad (4.14)$$

The objective function (4.12) is to minimize the expected damage. Constraint (4.13) limits the number of detectors. Constraints (4.14) are binary constraints.

The model [DS] is equivalent to the following model:

$$[\text{EDS}] \quad \min_{X, Z} \quad Z$$

$$\text{s.t.} \quad Z \geq \sum_{r \in \bar{R}} C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right), \quad (4.15)$$

$$\sum_{i \in T} x_i \leq M, \quad (4.16)$$

$$x_i \in \{0, 1\}, \quad \forall i \in T. \quad (4.17)$$

We now consider our defender-attacker model. The terrorist group wants to maximize the expected damage by selecting paths for their small boats. The defender wishes to minimize this maximum damage by locating detectors. We formulate

this problem as a path-based min-max nonlinear integer program as follows:

[DAM]

$$\min_X \max_Y \sum_{r \in R} C_r y_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right) \quad (4.18)$$

$$\text{s.t.} \quad \sum_{i \in T} x_i \leq M, \quad (4.19)$$

$$\sum_{r \in R} y_r \leq B, \quad (4.20)$$

$$\sum_{r \in R} a_{rk} y_r \leq 1, \quad \forall k \in A, \quad (4.21)$$

$$x_i \in \{0, 1\}, \quad \forall i \in T, \quad (4.22)$$

$$y_r \in \{0, 1\}, \quad \forall r \in R. \quad (4.23)$$

4.3 Algorithms

A min-max problem is not a standard optimization problem which can be solved by general methods. For our min-max model, we borrow the cutting-plane-based algorithm from [12] where the algorithm is applied to solve the following infinitely constrained optimization problem:

$$\begin{aligned} \min_{x \in \mathbf{X}} \quad & f(x) \\ \text{s.t.} \quad & g(x, y) \leq 0, \quad \forall y \in \mathbf{Y}. \end{aligned}$$

In fact, our defender-attacker model [DAM] is equivalent to the following semi-infinite programming:

$$\begin{aligned} & \min_{(X,Z) \in \mathcal{X}} && Z \\ & \text{s.t.} && \sum_{r \in R} C_r y_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right) - Z \leq 0, \quad \forall Y \in \mathcal{Y}, \end{aligned}$$

where

$$\mathcal{X} = \left\{ (X, Z) = (x_1, x_2, \dots, x_{|T|}, Z) \mid \sum_{i \in T} x_i \leq M, x_i \in \{0, 1\}, 0 \leq Z \leq M_b \right\},$$

in which M_b is a very large number, and

$$\mathcal{Y} = \left\{ Y = (y_1, y_2, \dots, y_{|R|}) \mid \sum_{r \in R} y_r \leq B, \sum_{r \in R} a_{rk} y_r \leq 1, y_r \in \{0, 1\} \right\}.$$

Now, we describe the cutting-plane framework to solve our problem.

Algorithm 4.1. Cutting-Plane Framework

Step 0. Initially choose an attacking plan $Y^1 \in \mathcal{Y}$, that is, initially choose an attacking path set R^1 . Let $K = 1$ and optimality tolerance $\varepsilon \geq 0$.

Step 1. For given $Y^l (R^l)$, $l = 1, \dots, K$, solve the following program (the defender's model):

$$\begin{aligned}
 \text{[EDS-K]} \quad & \min_{X, Z} && Z \\
 \text{s.t.} \quad & Z \geq \sum_{r \in R^l} C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right), \\
 & && l = 1, \dots, K, \\
 & \sum_{i \in T} x_i \leq M, \\
 & x_i \in \{0, 1\}, && \forall i \in T,
 \end{aligned}$$

and get optimal defense plan X^K and the corresponding objective function value Z_{min}^K .

Step 2. For given X^K , solve the attacker's model [EAS] to determine optimal solution Y^{K+1} and its associated objective function value Z_{max}^K .

Step 3. If $Z_{max}^K - Z_{min}^K < \varepsilon$, stop. Otherwise, let $K = K + 1$ and go to Step 1.

This cutting-plane method can be used to solve our min-max problem. However, there are still two subproblems which should be investigated deeply: How to solve program [EDS-K] to get X^K in Step 1 (how to get the optimal detector deployment for fixed attacking plans) and how to solve model [EAS] to obtain Y^{K+1} for given X^K in Step 2 (how to obtain the optimal attacking plan for fixed detector locations) efficiently. We will discuss them in Subsection 4.3.1 and Subsection 4.3.2, respectively.

4.3.1 Optimal Detector Deployment for Fixed Attacking Plans

Now, we investigate how to solve program [EDS-K]. It is a binary nonlinear program whose relaxation problem is as follows:

[EDS-K-R]

$$\min_{X, Z} \quad Z \quad (4.24)$$

$$\text{s.t. } Z \geq \sum_{r \in R^l} C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \theta + (1 - \theta) \right), \quad l = 1, \dots, K, \quad (4.25)$$

$$\sum_{i \in T} x_i \leq M, \quad (4.26)$$

$$0 \leq x_i \leq 1, \quad \forall i \in T. \quad (4.27)$$

Obviously, the relaxation program [EDS-K-R] is a convex program which enables us to solve formulation [EDS-K] by the standard branch and bound algorithm which is presented in Chapter 2. However, during the process of the standard branch and bound algorithm, we need to solve nonlinear relaxation programs repeatedly, which results in expensive computational costs. In view of this, we recommend a linear outer approximation method which is borrowed from [87]. We define two kinds of new decision variables Φ_r and Ψ_r for each $r \in R^l$, $l = 1, \dots, K$, as follows:

$$\Phi_r = \prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{x_i} \text{ and } \Psi_r = \ln(\Phi_r) = \sum_{k \in r} \sum_{i \in N_{kr}} x_i \ln(1 - p_{ikr}).$$

Moreover, in order to apply the linear outer approximation approach, we set lower bound $\mathcal{L}_r$ and upper bound $\mathcal{U}_r$ for each Φ_r . The initial bounds are given by

$$\mathcal{L}_r^0 = \prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr}) \text{ and } \mathcal{U}_r^0 = 1, \forall r \in R^l, l = 1, \dots, K. \quad (4.28)$$

Then, we get an equivalent formulation for [EDS-K] as follows:

$$[\text{EDS-K}(\mathcal{L}, \mathcal{U})]$$

$$\min_{X, Z, \Phi, \Psi} Z \quad (4.29)$$

$$\text{s.t. } Z \geq \sum_{r \in R^l} C_r \left(\Phi_r \theta + (1 - \theta) \right), \quad l = 1, \dots, K, \quad (4.30)$$

$$\sum_{i \in T} x_i \leq M, \quad (4.31)$$

$$\Psi_r = \ln(\Phi_r), \quad \forall r \in R^l, l = 1, \dots, K, \quad (4.32)$$

$$\Psi_r = \sum_{k \in r} \sum_{i \in N_{kr}} x_i \ln(1 - p_{ikr}), \quad \forall r \in R^l, l = 1, \dots, K, \quad (4.33)$$

$$\mathcal{L}_r \leq \Phi_r \leq \mathcal{U}_r, \quad \forall r \in R^l, l = 1, \dots, K, \quad (4.34)$$

$$x_i \in \{0, 1\}, \quad \forall i \in T. \quad (4.35)$$

Program [EDS-K($\mathcal{L}, \mathcal{U}$)] is a binary linear program if nonlinear constraints (4.32) are removed. Here, the nonlinear constraints can be relaxed through their associated polyhedral outer approximations over $[\mathcal{L}_r, \mathcal{U}_r]$. The polyhedral outer approximation consists of a convex envelope and three tangential supports at three points $\mathcal{L}_r$, $\bar{\Phi}_r = \frac{\mathcal{U}_r - \mathcal{L}_r}{\ln(\mathcal{U}_r) - \ln(\mathcal{L}_r)}$, and $\mathcal{U}_r$. The resulting linearized relaxation program is as follows:

[EDS-K($\mathcal{L}, \mathcal{U}$)-R]

$$\begin{aligned}
& \min_{X, Z, \Phi, \Psi} && Z \\
& \text{s.t.} && Z \geq \sum_{r \in R^l} C_r \left(\Phi_r \theta + (1 - \theta) \right), \quad l = 1, \dots, K, \\
& && \sum_{i \in T} x_i \leq M, \\
& && \Psi_r = \sum_{k \in r} \sum_{i \in N_{kr}} x_i \ln(1 - p_{ikr}), \quad \forall r \in R^l, \quad l = 1, \dots, K, \\
& && \Psi_r \geq \ln(\mathcal{L}_r) + \frac{\ln(\mathcal{U}_r) - \ln(\mathcal{L}_r)}{\mathcal{U}_r - \mathcal{L}_r} (\Phi_r - \mathcal{L}_r), \quad \forall r \in R^l, \quad l = 1, \dots, K, \\
& && \Psi_r \leq \ln(\mathcal{L}_r) + \frac{\Phi_r - \mathcal{L}_r}{\mathcal{L}_r}, \quad \forall r \in R^l, \quad l = 1, \dots, K, \\
& && \Psi_r \leq \ln(\bar{\Phi}_r) + \frac{\Phi_r - \bar{\Phi}_r}{\bar{\Phi}_r}, \quad \forall r \in R^l, \quad l = 1, \dots, K, \\
& && \Psi_r \leq \ln(\mathcal{U}_r) + \frac{\Phi_r - \mathcal{U}_r}{\mathcal{U}_r}, \quad \forall r \in R^l, \quad l = 1, \dots, K, \\
& && \mathcal{L}_r \leq \Phi_r \leq \mathcal{U}_r, \quad \forall r \in R^l, \quad l = 1, \dots, K, \\
& && x_i \in \{0, 1\}, \quad \forall i \in T.
\end{aligned}$$

Now, we state the linear outer approximation method which we call the linearized branch and bound algorithm to solve [EDS-K] via [EDS-K($\mathcal{L}, \mathcal{U}$)-R].

Algorithm 4.2. Linearized Branch and Bound Algorithm for [EDS-K]

Step 0. (Initialization) Let stage $s = 0$, active node set $O_s = \{0\}$, current node $n(s) = 0$, $n = 0$, initial bounds $\mathcal{L}^0$ and $\mathcal{U}^0$ be provided by (4.28), and optimality tolerance $\varepsilon > 0$. Obtain an optimal solution $(X^0, Z^0, \Phi^0, \Psi^0)$ and the corresponding objective function value $V[\text{EDS-K}(\mathcal{L}^0, \mathcal{U}^0)\text{-R}]$ by solving [EDS-K($\mathcal{L}^0, \mathcal{U}^0$)-R]. Set upper bound UB be the objective function value of incumbent X^0 calculated through [EDS-K] since X^0 is feasible to [EDS-K]. If $V[\text{EDS-K}(\mathcal{L}^0, \mathcal{U}^0)\text{-R}] \geq$

$UB(1 - \varepsilon)$, stop the procedure. Otherwise, go to the next step.

Step 1. (Division) Based on the optimal solution $(X^{n(s)}, Z^{n(s)}, \Phi^{n(s)}, \Psi^{n(s)})$ of $[EDS-K(\mathcal{L}^{n(s)}, \mathcal{U}^{n(s)})-R]$ which corresponds to node $n(s)$, we find index r' such that

$$|\Psi_{r'}^{n(s)} - \ln(\Phi_{r'}^{n(s)})| = \max_{\substack{r \in R^l \\ l=1, \dots, K}} |\Psi_r^{n(s)} - \ln(\Phi_r^{n(s)})|.$$

Then, we split interval $(\mathcal{L}_{r'}^{n(s)}, \mathcal{U}_{r'}^{n(s)})$ into two subintervals $(\mathcal{L}_{r'}^{n(s)}, \Phi_{r'}^{n(s)})$ and $(\Phi_{r'}^{n(s)}, \mathcal{U}_{r'}^{n(s)})$, which leads to two child nodes $n+1$ and $n+2$. Add the new nodes $n+1$ and $n+2$ to Q_s and denote their bounds as $(\mathcal{L}^{n+1}, \mathcal{U}^{n+1})$ and $(\mathcal{L}^{n+2}, \mathcal{U}^{n+2})$, respectively.

Step 2. (Bound updating and fathoming) Solve the two new programs $[EDS-K(\mathcal{L}^{n+1}, \mathcal{U}^{n+1})-R]$ and $[EDS-K(\mathcal{L}^{n+2}, \mathcal{U}^{n+2})-R]$, respectively. According to the objective function values of $[EDS-K]$ by taking the X parts of their solutions, we improve the incumbent solution and its objective function value UB . Update O_{s+1} by selecting all nodes from O_s whose objective function value is smaller than $UB(1 - \varepsilon)$. Set $n = n + 2$ and $s = s + 1$.

Step 3. (Termination and node selection) Check whether O_s is an empty set. If it is, stop and return the incumbent. Otherwise, select a node $n(s)$ from O_s that yields the smallest objective function value and go to Step 1.

Theorem 4.1. Let $\varepsilon = 0$. The sequence $\{X^{n(s)}\}$ generated by the linearized branch and bound algorithm converges to a global optimal point of $[EDS-K]$.

4.3.2 Optimal Attacking Plan for Fixed Detector Locations

Given the defender's strategy $\bar{X}$, the attacker's problem can be expressed as program [EAS]. It is a binary linear program and its corresponding relaxation problem is as follows:

$$\text{[EAS-R]} \quad \max_Y \quad \sum_{r \in R} L_r y_r \quad (4.36)$$

$$\text{s.t.} \quad \sum_{r \in R} y_r \leq B, \quad (4.37)$$

$$\sum_{r \in R} a_{rk} y_r \leq 1, \quad \forall k \in A, \quad (4.38)$$

$$y_r \geq 0, \quad \forall r \in R, \quad (4.39)$$

when we relax binary constraints (4.11), constraints $y_r \leq 1, \forall r \in R$ are dropped because they are directly implied from constraints (4.10).

Relaxation problem [EAS-R] is a linear program. However, the fact that the number of decision variables grows exponentially with respect to the size of the problem results in a huge number of columns which can not be listed completely. Moreover, obtaining the values of L_r for all $r \in R$ is computationally expensive. Furthermore, even if all decision variables and their corresponding coefficients are given, the program is nontrivial because of its large size. Due to these issues, we apply a column-generation method. There are many works, such as [35, 54, 108], that discuss column generation algorithms which are a pricing scheme for solving large-scale linear programs. For [EAS-R], each iteration generates a new path

whose reduced cost is most positive. The specific process of the column generation approach to solve [EAS-R] is as follows:

Algorithm 4.3. Column Generation Algorithm for [EAS-R]

Step 0. Set iteration $s = 0$ and initial path subset $R^{(0)} \subset R$ which is generated heuristically.

Step 1. Solve model [EAS-R] over subset $R^{(s)} \subset R$, that is, solve the following program:

$$\max_Y \quad \sum_{r \in R^{(s)}} L_r y_r \quad (4.40)$$

$$\text{s.t.} \quad \sum_{r \in R^{(s)}} y_r \leq B, \quad (4.41)$$

$$\sum_{r \in R^{(s)}} a_{rk} y_r \leq 1, \quad \forall k \in A, \quad (4.42)$$

$$y_r \geq 0, \quad \forall r \in R^{(s)}. \quad (4.43)$$

and obtain an optimal primal solution $Y^{(s)}$ and dual solution $\pi_0^{(s)}$ and $\pi_k^{(s)}$ corresponding to constraint (4.41) and constraints (4.42), respectively.

Step 2. Maximize the reduced cost

$$L_r - \pi_0^{(s)} - \sum_{k \in A} a_{kr} \pi_k^{(s)}.$$

Step 3. If the maximum reduced cost is negative, the current solution $Y^{(s)}$ is optimal. Otherwise, we add the path with the maximum reduced cost to subset $R^{(s)}$. Let $s = s + 1$ and go to Step 1.

In the process of Algorithm 4.3, there is a key step, that is, Step 2, which is called a pricing problem for finding a path with the maximum reduced cost. For the convenience of analysis, in the rest of this subsection, we replace $\pi_0^{(s)}$ and $\pi_k^{(s)}$ by π_0 and π_k , respectively. Since $p_{ikr} = 1 - e^{-\eta l_{ikr}}$ and $L_r = C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{\bar{x}_i} \theta + (1 - \theta) \right)$, we get an equivalent reduced cost function as follows:

$$\begin{aligned}
L_r - \pi_0 - \sum_{k \in A} a_{kr} \pi_k &= C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} (1 - p_{ikr})^{\bar{x}_i} \theta + (1 - \theta) \right) - \pi_0 - \sum_{k \in A} a_{kr} \pi_k \\
&= C_r \left(\prod_{k \in r} \prod_{i \in N_{kr}} e^{-\eta l_{ikr} \bar{x}_i} \theta + (1 - \theta) \right) - \pi_0 - \sum_{k \in A} a_{kr} \pi_k \\
&= C_r \left(e^{-\eta \sum_{k \in r} \sum_{i \in N_{kr}} l_{ikr} \bar{x}_i} \theta + (1 - \theta) \right) - \pi_0 - \sum_{k \in A} a_{kr} \pi_k \\
&= C_r \theta \left(e^{-\sum_{k \in r} l'_{kr}} - \sum_{k \in A} \pi'_k a_{kr} \right) + C_r (1 - \theta) - \pi_0, \quad (4.44)
\end{aligned}$$

where, $l'_{kr} = \eta \sum_{i \in N_{kr}} l_{ikr} \bar{x}_i$ and $\pi'_k = \frac{\pi_k}{C_r \theta}$.

Based on (4.44), the essence of our pricing problem is to find a nonlinear objective shortest path on network $\mathcal{G}(G, A)$ from entrances to targets. Hence, we need to find all nonlinear objective shortest paths between entrance and target pairs. For each entrance and target pair (i, j) , we solve the following program to find a nonlinear objective shortest path from entrance i to target j :

$$\begin{aligned}
[\text{SPG-}ij] \quad & \min \quad \sum_{k \in A} \pi'_k a_k - e^{-\sum_{k \in A} l'_{kj} a_k} \\
& \text{s.t.} \quad \sum_{k \in FS(i)} a_k - \sum_{k \in RS(i)} a_k = 1, \\
& \quad \sum_{k \in FS(n)} a_k - \sum_{k \in RS(n)} a_k = 0, \quad \forall n \in G \setminus \{i, j\} \\
& \quad \sum_{k \in FS(j)} a_k - \sum_{k \in RS(j)} a_k = -1, \\
& \quad a_k \in \{0, 1\}, \quad k \in A,
\end{aligned}$$

where $FS(n)$ is the collection of arcs which leave node n and $RS(n)$ is the collection of arcs which enter node n . It is a nonlinear binary program and its relaxation program is non-convex. We also explore the linear outer approximation approach to solve it efficiently. Let us introduce two auxiliary decision variables μ and ν such that

$$\mu = e^{-\sum_{k \in A} l'_{kj} a_k} \text{ and } \nu = \ln(\mu) = -\sum_{k \in A} l'_{kj} a_k.$$

The lower and upper bounds for μ are denoted as L and U and the initial values are given by

$$L^0 = e^{-\sum_{k \in A} l'_{kj}} \text{ and } U^0 = 1. \quad (4.45)$$

According to the description of the two new variables, we obtain the following equivalent program for [SPG- ij]:

$$[\text{ESPG-}ij(\mathbf{L}, \mathbf{U})]$$

$$\min \quad \sum_{k \in A} \pi'_k a_k - \mu \quad (4.46)$$

$$\text{s.t.} \quad \sum_{k \in FS(i)} a_k - \sum_{k \in RS(i)} a_k = 1, \quad (4.47)$$

$$\sum_{k \in FS(n)} a_k - \sum_{k \in RS(n)} a_k = 0, \quad \forall n \in G \setminus \{i, j\}, \quad (4.48)$$

$$\sum_{k \in FS(j)} a_k - \sum_{k \in RS(j)} a_k = -1, \quad (4.49)$$

$$\nu = - \sum_{k \in A} l'_{kj} a_k, \quad (4.50)$$

$$\nu = \ln(\mu), \quad (4.51)$$

$$L \leq \mu \leq U, \quad (4.52)$$

$$a_k \in \{0, 1\}, \quad k \in A. \quad (4.53)$$

We obtain the linearized relaxation program as follows:

[ESPG- $ij(\mathbf{L}, \mathbf{U})$ -R]

$$\begin{aligned}
& \min \quad \sum_{k \in A} \pi'_k a_k - \mu \\
& \text{s.t.} \quad \sum_{k \in FS(i)} a_k - \sum_{k \in RS(i)} a_k = 1, \\
& \quad \sum_{k \in FS(n)} a_k - \sum_{k \in RS(n)} a_k = 0, \quad \forall n \in G \setminus \{i, j\} \\
& \quad \sum_{k \in FS(j)} a_k - \sum_{k \in RS(j)} a_k = -1, \\
& \quad \nu = - \sum_{k \in A} l'_{kj} a_k, \\
& \quad \nu \geq \ln(L) + \frac{\ln(U) - \ln(L)}{U - L} (\mu - L), \\
& \quad \nu \leq \ln(L) + \frac{\mu - L}{L}, \\
& \quad \nu \leq \ln(\bar{\mu}) + \frac{\mu - \bar{\mu}}{\bar{\mu}}, \\
& \quad \nu \leq \ln(U) + \frac{\mu - U}{U}, \\
& \quad L \leq \mu \leq U, \\
& \quad a_k \in \{0, 1\}, \quad k \in A.
\end{aligned}$$

The process of the linearized branch and bound algorithm to solve [SPG- ij] via [ESPG- $ij(\mathbf{L}, \mathbf{U})$ -R] is similar to that of Algorithm 4.2. For the sake of clarity and completeness, we describe the procedure as follows:

Algorithm 4.4. Linearized Branch and Bound Algorithm for [SPG- ij]

Step 0. (Initialization) Let stage $s = 0$, active node set $O_s = \{0\}$, current node $n(s) = 0$, $n = 0$, initial bounds $\mathbf{L}^0$ and $\mathbf{U}^0$ be provided by (4.45), and optimality tolerance $\varepsilon > 0$. Obtain an optimal solution $(\mathbf{a}^0, \mu^0, \nu^0)$ and the corresponding

objective function value $V[\text{ESPG-}ij(\mathbf{L}^0, \mathbf{U}^0)\text{-R}]$ by solving $[\text{ESPG-}ij(\mathbf{L}^0, \mathbf{U}^0)\text{-R}]$. Set upper bound UB be the objective function value of the incumbent $\mathbf{a}^0$ calculated through the objective function of $[\text{SPG-}ij]$ since $\mathbf{a}^0$ is feasible to $[\text{SPG-}ij]$. If $V[\text{ESPG-}ij(\mathbf{L}^0, \mathbf{U}^0)\text{-R}] \geq \text{UB}(1 - \varepsilon)$, stop the procedure. Otherwise, go to the next step.

Step 1. (Division) Based on an optimal solution $(\mathbf{a}^{n(s)}, \mu^{n(s)}, \nu^{n(s)})$ of $[\text{ESPG-}ij(\mathbf{L}^{n(s)}, \mathbf{U}^{n(s)})\text{-R}]$ which corresponds to node $n(s)$, we split interval $(\mathbf{L}^{n(s)}, \mathbf{U}^{n(s)})$ into two subintervals $(\mathbf{L}^{n(s)}, \mu^{n(s)})$ and $(\mu^{n(s)}, \mathbf{U}^{n(s)})$, which leads to two child nodes $n + 1$ and $n + 2$ associated with $[\text{ESPG-}ij(\mathbf{L}^{n(s)}, \mu^{n(s)})\text{-R}]$ and $[\text{ESPG-}ij(\mu^{n(s)}, \mathbf{U}^{n(s)})\text{-R}]$, respectively. Add the new nodes $n + 1$ and $n + 2$ to Q_s and denote their bounds as $(\mathbf{L}^{n+1}, \mathbf{U}^{n+1})$ and $(\mathbf{L}^{n+2}, \mathbf{U}^{n+2})$, respectively.

Step 2. (Bound updating and fathoming) Solve the two new programs $[\text{ESPG-}ij(\mathbf{L}^{n+1}, \mathbf{U}^{n+1})\text{-R}]$ and $[\text{ESPG-}ij(\mathbf{L}^{n+2}, \mathbf{U}^{n+2})\text{-R}]$, respectively. According to the objective function values of $[\text{SPG-}ij]$ by taking the $\mathbf{a}$ parts of their solutions, improve the incumbent solution and its objective function value UB. Update O_{s+1} by selecting all nodes from O_s whose objective function value is smaller than $\text{UB}(1 - \varepsilon)$. Set $n = n + 2$ and $s = s + 1$.

Step 3. (Termination and node selection) Check whether O_s is an empty set. If it is, stop and return the incumbent. Otherwise, select a node $n(s)$ from O_s that yields the smallest objective function value of its program and go to Step 1.

The linearized branch and bound algorithm for $[\text{SPG-}ij]$ also has global convergence property.

Theorem 4.2. Let $\varepsilon = 0$. The sequence $\{\mathbf{a}^{n(s)}\}$ generated by Algorithm 4.4 converges to a global optimal point of [SPG- ij].

We state how to solve the relaxation linear program [EAS-R] by the column-generation algorithm. However, the solution to [EAS-R] is not a solution to [EAS] since it is not always integer. Hence, we apply a branch-and-pricing method, which is similar to the method in [108], to get an integer solution to [EAS] through its relaxation program [EAS-R].

Algorithm 4.5. Branch-and-Pricing Algorithm for [EAS]

Step 0. (Initialization) Let stage $s = 0$, active node set $O_s = \{0\}$, current node $n(s) = 0$, $n = 0$, and optimality tolerance $\varepsilon > 0$. Let lower bound LB be the objective function value of [EAS] at origin. Obtain a solution Y^0 by solving [EAS-R] using column generation method in the network $\mathcal{G}(G, A)$. If all values of Y^0 are integers, the procedure is terminated and Y^0 is an optimal solution. Otherwise, go to Step 1.

Step 1. (Division) According to a solution $Y^{n(s)}$ to the relaxation problem [EAS-R] of node $n(s)$, we seek its first non-integer element index i' . We construct two constraints $Y(i') = 0$ and $Y(i') = 1$ which bring about two new child nodes $n + 1$ and $n + 2$ split from node $n(s)$. Add constraint $Y(i') = 0$ to the problem of node $n + 1$ and $Y(i') = 1$ to the problem of node $n + 2$. That is, for the problem of node $n + 1$, we add a new constraint

$$\sum_{k \in \Omega^{n(s)}} a_k \leq |\Omega^{n(s)}| - 1,$$

to all column generation subproblems [SPG- ij], where $\Omega^{n(s)}$ is the set of arcs which belong to the path corresponding to $Y^{n(s)}(i')$ and its size is denoted by $|\Omega^{n(s)}|$, while for the problem of node $n+2$, we modify its corresponding network by deleting arcs in $\Omega^{n(s)}$ and solve it in the new network. Then, let $O_s = O_s \cup \{n+1, n+2\} \setminus \{n(s)\}$.

Step 2. (Bound updating and fathoming) Update the incumbent solution and its objective function value LB based on the solutions to the relaxation programs of nodes $n+1$ and $n+2$. Update O_{s+1} by selecting all nodes from O_s whose objective function value is larger than $LB(1 + \varepsilon)$. Set $n = n + 2$ and $s = s + 1$.

Step 3. (Termination and node selection) Check whether O_s is an empty set. If it is, stop and return the incumbent. Otherwise, select a node $n(s)$ from O_s that yields the largest objective function value of its program and go to Step 1.

4.3.3 Algorithms for Our Defender-Attacker Model

Based on the two methods to solve [EDS-K], we have provided two exact cutting-plane-based algorithms for solving the defender-attacker model: a nonlinear cutting-plane algorithm where [EDS-K] is solved by the standard branch and bound method, and a linearized cutting-plane algorithm where [EDS-K] is solved by the linearized branch and bound method.

The cutting-plane methods can find an exact solution for our defender-attacker model, but computational costs are very expensive for large-size problems. Hence, we develop a greedy heuristic for large-scale problems to obtain good solutions in a short time period. The detail is as follows:

Algorithm 4.6. Greedy Algorithm for Our Defender-Attacker Model

Step 0. (Initialization) Let $Q = \emptyset$ be the initial placement and T be the initial possible location set.

Step 1. (Termination checking) Check whether $|Q| \geq M$. If yes, stop the procedure. Otherwise, go to Step 2.

Step 2. (Selection) For each location $i \in T$, we calculate the maximum expected damage when the placement set is $Q \cup \{i\}$ by sequentially choosing path r with the largest L_r value in the network where previously chosen paths are deleted from network $\mathcal{G}(G, A)$ until the number of the chosen paths is larger than that of small vessels. In order to simplify this process, we create a dummy node as an artificial entrance, labeled as 0, which just has links with the actual entrances and the weights of these links are zero. Hence, the process is that for each artificial entrance and target pair we find its shortest path based on the evasion probability in the corresponding network. Then, among them we choose one with the largest L_r value. Sum all L_r values of the selected paths to get the maximum expected damage. Then, we choose i' such that

$$i' \in \arg \min_{i \in T} \{\text{the maximum expected damage value if adding } i \text{ to } Q\}.$$

Step 3. (Updating) Let $Q = Q \cup \{i'\}$ and $T = T \setminus \{i'\}$. Then, go to Step 1.

4.4 Computational Study

This section shows computational results relevant to our problem. There are two purposes to be fulfilled. One is to illustrate the application of our model and methodology through describing the surveillance strategy of a real port. The other is to investigate the performance of the proposed algorithms. We conduct our experiments on a PC with Intel Core 2 Duo 3.10GHz CPU and 4GB of RAM. All codes are written in Matlab. In our tests, continuous nonlinear programs are solved by the `fmincon` function in Matlab. Linear programs or mixed integer linear programs are solved by IBM ILOG CPLEX V12.4 in a Matlab environment. The optimality tolerance is set to 10^{-4} .

4.4.1 Illustrative Case

We apply our model and methodology to a bay in the port of New York and New Jersey which is highlighted in Figure 4.3. It is reported that the port of New York and New Jersey is the third largest in the U.S. in terms of tonnage and the busiest on the east coast [30]. In the port, there are two valuable assets: Statue of Liberty National Monument and Red Hook Container Terminal which are also highlighted in Figure 4.3. The area under study is about 10 kilometers by 7 kilometers. We discretize it into 20×14 grids and the width of each grid is 0.5 kilometers, which is shown in Figure 4.4. The entrance set $E = \{1, 2, 3, 6, 7, 8, 9, 10, 11, 12, 260, 272, 273\}$, the target set $S = \{170, 176\}$ with $C_j = 35$ million per small vessel attack for $\forall j \in S$, and the block set D contains all

shaded grids. The attacker owns three small vessels whose speed is 20 m/s. The defender deploys three detectors for this area. The detectors are characterized by a detection radius $\tau = 2$ kilometers and an instantaneous detection rate $\eta = 0.006$. We set a timely detection time $\lambda = 10$ seconds and the probability of successful neutralization $\theta = 0.6$. As shown in Figure 4.4, the optimal detector placement for the defender is $x_{110} = 1, x_{193} = 1$, and $x_{235} = 1$, and the optimal routes for the attackers are $6 \rightarrow 25 \rightarrow 46 \rightarrow 67 \rightarrow 88 \rightarrow 108 \rightarrow 128 \rightarrow 149 \rightarrow 170$, $12 \rightarrow 32 \rightarrow 53 \rightarrow 74 \rightarrow 94 \rightarrow 114 \rightarrow 113 \rightarrow 112 \rightarrow 111 \rightarrow 130 \rightarrow 150 \rightarrow 170$, and $7 \rightarrow 26 \rightarrow 47 \rightarrow 68 \rightarrow 89 \rightarrow 109 \rightarrow 129 \rightarrow 149 \rightarrow 169 \rightarrow 170$. The expected damage is 54.080 million.

FIGURE 4.3: A concerned bay in the Port of New York and New Jersey.

FIGURE 4.4: The optimal detector placement and attacking routes.

Since the solution of our problem is sensitive to the parameters involved in the model, we conduct sensitivity analysis with respect to three most important parameters: the total number of detectors M , the total number of small boats B , and target values C_j . The numerical results are shown in Table 4.2, Table 4.3, and Table 4.4. In these tables, SD represents the optimal solution for the defender, SA represents the optimal solution for the attacker, and OV represents the resulting objective function value. From Table 4.2, we can see that the expected damage will decrease as the number of detectors increases, which is intuitive. When the number of detectors is small, we would like to deploy the detectors in the area around the targets. The monitoring of the regions near entrances is strengthened as the number of detectors increases. The results of Table 4.3 show that the objective value will increase when the terrorist group utilizes more small boats.

Moreover, the larger the number of attacking paths, the closer the locations of detectors to the targets. From the results of Table 4.4, we observe that the attackers prefer to attack the target with a higher value under the same evasion probability. Moreover, the more a target is likely to be attacked, the more protection it gets.

TABLE 4.2: Sensitivity analysis with respect to different detector numbers

M	SD	SA	OV
1	191	6, 26, 47, 68, 88, 108, 129, 150, 170; 260, 259, 258, 237, 216, 195, 176; 11, 31, 51, 71, 90, 109, 128, 149, 170	87.803
2	130,214	10, 31, 52, 73, 94, 114, 133, 132, 151, 170; 11, 32, 53, 74, 93, 112, 131, 150, 170; 260, 259, 258, 237, 216, 196, 176	64.016
3	110,193,235	6, 25, 46, 67, 88, 108, 128, 149, 170; 12, 32, 53, 74, 94, 114, 113, 112, 111, 130, 150, 170; 7, 26, 47, 68, 89, 109, 129, 149, 169, 170	54.080
4	110,151,234,235	272, 252, 231, 210, 190, 170; 11, 32, 53, 74, 94, 113, 132, 151, 170; 260, 259, 258, 237, 216, 196, 176	48.671
5	90,130,193,234,235	6, 25, 46, 67, 88, 108, 128, 149, 170; 272, 252, 231, 210, 190, 170; 260, 259, 258, 237, 216, 196, 176	45.484

TABLE 4.3: Sensitivity analysis with respect to different small boat numbers

B	SD	SA	OV
1	110,173,235	6, 25, 46, 67, 88, 108, 128, 149, 170	18.116
2	109,173,235	272, 252, 231, 210, 190, 170; 12, 32, 53, 73, 93, 113, 133, 153, 172, 171, 170	36.143
3	110,193,235	6, 25, 46, 67, 88, 108, 128, 149, 170; 12, 32, 53, 74, 94, 114, 113, 112, 111, 130, 150, 170; 7, 26, 47, 68, 89, 109, 129, 149, 169, 170	54.080
4	109,173,214	11, 32, 53, 72, 91, 110, 129, 149, 170; 6, 25, 46, 67, 88, 108, 128, 148, 169, 170; 10, 31, 52, 71, 90, 110, 130, 150, 170; 260, 259, 258, 237, 216, 196, 176	71.397
5	109,173,193	11, 32, 53, 72, 91, 110, 129, 149, 170; 6, 25, 46, 67, 88, 108, 128, 148, 169, 170; 10, 31, 52, 71, 90, 109, 129, 150, 170; 260, 259, 258, 237, 216, 196, 176; 12, 32, 52, 73, 93, 113, 133, 154, 175, 176	88.237

TABLE 4.4: Sensitivity analysis with respect to different target values

C	SD	SA	OV
35, 35	110,193,235	6, 25, 46, 67, 88, 108, 128, 149, 170; 12, 32, 53, 74, 94, 114, 113, 112, 111, 130, 150, 170; 7, 26, 47, 68, 89, 109, 129, 149, 169, 170	54.080
35, 30	150,151,235	272, 252, 231, 210, 190, 170; 6, 25, 46, 67, 88, 108, 128, 149, 170; 260, 259, 258, 237, 216, 196, 176	51.214
35, 25	130,151,214	272, 252, 231, 210, 190, 170; 6, 25, 46, 67, 88, 108, 128, 149, 170; 12, 31, 52, 73, 94, 113, 132, 151, 170	49.566
35, 20	109,171,193	272, 252, 231, 210, 190, 170; 6, 25, 46, 67, 88, 108, 128, 149, 170; 273, 252, 232, 211, 191, 170	47.776
35, 15	150,151,191	6, 25, 46, 67, 88, 108, 128, 149, 170; 272, 252, 231, 210, 190, 170; 273, 253, 233, 212, 191, 170	46.790
35, 10	150,151,191	6, 25, 46, 67, 88, 108, 128, 149, 170; 272, 252, 231, 210, 190, 170; 273, 253, 233, 212, 191, 170	46.790

4.4.2 Performance of Algorithms

In this subsection, we investigate the performance of the three algorithms: the nonlinear cutting-plane algorithm (NCA), the linearized cutting-plane algorithm (LCA), and the greedy algorithm (GA) for the defender-attacker model. In this study, we concentrate on square threat areas which are discretized into $n \times n$ grids. The size of a unit grid is 220 m $\times$ 220 m. We conduct experiments for $n = 5, 6, 7, 8$, and 9. For each dimension, the numbers of entrances, targets, and blocked grids are uniformly drawn from sets $\{3, 4, 5, 6\}$, $\{1, 2, 3\}$, and $\{1, 2, \dots, \lfloor \frac{n \times n}{6} \rfloor\}$, respectively. Their corresponding locations are also randomly determined. Moreover, we conduct experiments for two scenarios. One is that all targets have the same value, the other is that different targets have different values. That is, for the former scenario, $C_j = 35, \forall j \in S$, and for the later one, $C_j = \max\{35 - 5(j - 1), 5\}$, $j = 1, 2, \dots, |S|$. We summarize other parameters in Table 4.5. For each dimension of the two scenarios, we randomly run ten instances.

TABLE 4.5: Parameter values

Parameter	Description	Value
θ	Probability of successful intervention	0.6
λ	Timely detection time	10 s
ψ	Speed of a small vessel	20 m/s
M	Total number of detectors	3
B	Total number of small boats	3
τ	Effective detection radius	1000 m
η	Instantaneous detection rate	0.006

Table 4.6 and Table 4.7 show the associated results for the two scenarios, respectively. In both tables, Dim reports dimension and GR represents the relative error of the greedy heuristic which is calculated by the following formula:

$$\text{GR} = \frac{\text{Heuristic Value} - \text{Exact Value}}{\text{Exact Value}} \times 100.$$

From Table 4.6 and Table 4.7, we can see that LCA and NCA have the same optimal values while LCA takes much less time than NCA for most instances. We observe only 9 out of 100 instances where the CPU time of LCA is longer than that of NCA. This implies the efficiency of the linearization technique. As for the relative performance of GA, we see that in 92 out of 100 instances, GA obtains optimal solutions. The average GR is 0.01% and the maximum GR is 3.045%. Moreover, GA significantly reduces CPU time. The maximum CPU time of GA is 25.740 seconds.

TABLE 4.6: Computational results for problems with same target values

Dim	E	S	D	NCA		LCA		GA	
				Value	CPU (s)	Value	CPU (s)	Value	CPU (s)
5×5	2,3,6,15,16,23	22	1,8,10,24	84.419	39.359	84.419	1.451	84.419	0.140
	2,15,16,21,22	1,4	17	76.713	6.817	76.713	2.106	76.713	0.312
	1,4,6,16,21	7	24	84.419	17.145	84.419	5.320	84.419	0.218
	1,2,3,5,20,21	8,12,14	22	89.651	22.901	89.651	4.181	89.651	0.468
	1,15,16,24,25	7	11,20	73.709	9.563	73.709	2.122	73.709	0.172
	11,15,16,20,21,22	3,5,24	12	73.709	33.244	73.709	22.745	73.709	0.468
	1,3,6,10,16,23	15,25	12,19,21,22	76.713	36.551	76.713	8.346	76.713	0.281
	1,5,10	9	17,18,23	84.419	22.698	84.419	3.994	84.419	0.140
	2,15,20,21	1,8	5	84.419	21.419	84.419	8.221	84.419	0.312
	3,5,10	18	6,20	53.425	41.886	53.425	9.001	53.425	0.156
6×6	1,3,4,31	2	8,13,20	87.423	30.077	87.423	28.236	87.423	0.499
	5,6,7,24,31,33	26,27	4,11,19,23,29	89.651	127.094	89.651	8.689	89.651	0.920
	1,3,7,31	12,21,22	13	55.126	139.558	55.126	190.243	55.126	1.498
	2,13,18,19,24,25	16,30,35	3,14,21,23,29,31	74.932	122.461	74.932	46.254	74.932	1.326
	2,6,12,31	1,27,32	36	87.423	144.083	87.423	34.648	87.423	1.498
	1,6,24,30,32	3,5,12	2	89.899	237.075	89.899	102.290	89.899	1.513
	2,25,31	10,29,34	16,21	58.688	156.875	58.688	87.907	58.688	1.451
	3,5,25	32,35	26	66.368	1033.865	66.368	223.721	66.368	0.998
	3,6,32,33	26	10,15	84.419	131.977	84.419	28.923	84.419	0.499
	3,6,12,24,32	4	7	76.713	30.280	76.713	3.120	76.713	0.515
7×7	2,5,8,21,46,47	14,37,41	6,10,11,16,17,30,38	84.419	462.075	84.419	49.250	84.419	3.978
	2,5,8,22,46	7,9,18	16,20,21,26,32,33	89.899	279.803	89.899	280.973	89.899	4.009
	1,7,29,36,43,47	40	10,38,48	74.237	326.073	74.237	66.113	74.237	1.404
	1,4,45,49	10,18,28	3,9,30,33,38,40,43	71.233	71.233	71.233	94.927	71.233	3.931
	2,3,5,15,44,49	16,48	12,21,30,38,42	89.899	89.498	89.899	38.969	89.899	0.683
	1,21,49	29,39,44	5,7,16,42	49.539	250.569	49.539	210.945	50.391	4.103
	5,21,29,35,42,44	30	6,10,22,32,43	76.713	401.500	76.713	210.196	76.713	1.388
	3,21,28,35	29,33	14,15,16,18,41	60.045	2430.215	60.045	252.160	60.045	2.683
	5,6,7,29,44	4,14	11,13,18,22,35,36	95.130	156.204	95.130	58.079	95.130	2.699
	1,3,22,28,44,46	19,24	6,7,12,21,38,40,49	63.526	334.154	63.526	86.690	63.526	2.652
8×8	1,2,24,32,61	36,48	10,19,25,26,29,38,53,56	58.264	2569.617	58.264	547.844	58.264	7.020
	5,8,40,56	43,51	18,32,49,52	43.592	3722.231	43.592	3458.948	43.592	7.238
	24,25,32,63,64	10,14,20	1,4,19,22,23,29,36,38,40,56,60	61.051	109.731	61.051	34.773	61.051	10.405
	7,33,40,48,63	50	15,31,37,47,52,55,57,58	56.765	796.448	56.765	153.536	56.765	3.604
	1,25,58	53,59	18,32	74.237	1474.584	74.237	1668.072	74.237	7.457
	5,17,24,25,57	10,31	3,28	78.940	382.234	78.940	420.001	78.940	7.566
	1,6,58,59,62	30	9	50.115	2794.664	50.115	340.519	50.115	3.666
	17,24,48	32	6,26,47,52,54,56,63	76.713	216.436	76.713	182.100	76.713	3.572
	3,4,7,24,33,59	23,30,45	37,47,50,53	76.713	2309.813	76.713	647.030	76.713	10.826
	7,9,17,59,60	50	6,20,22,29,32,45,48,49,61	73.709	2180.660	73.709	982.245	73.709	3.557
9×9	1,8,37,64,75,76	5,18,41	9,10,12,13,22,27,28,30,32,49,50,60	68.870	7437.020	68.870	772.985	68.870	25.740
	3,6,64,80	68	13,65	48.670	17991.190	48.670	11290.245	48.670	8.206
	2,72,76,78	23,60,74	1,4,5,27,35,45,64	65.455	2006.625	65.455	940.093	65.455	0.553
	2,37,64,76,81	46	36,63,75	76.713	4039.474	76.713	2830.061	76.713	8.362
	46,73,80	15	27,76	42.756	16887.327	42.756	5745.954	42.756	8.299
	6,10,64	39,62,78	1,3,12,14,15,16,40,50,55,59,71,75,76,77	49.867	23196.522	49.867	5793.877	50.370	23.650
	1,2,19,28	21,51,60	4,6,7,12,31,37,38,53,67,69	63.526	560.278	63.526	288.773	63.526	24.009
	7,8,54,72	81	1,38,40,42,45,46,49,50,57,58,59,75	74.237	2369.187	74.237	1883.150	74.237	8.018
	8,10,19,55,72,75	25,77	20,52	71.310	3818.686	71.310	1485.410	71.310	16.833
	2,55,72	42	4,11,16,20,26,28,32,37,47,48,57,63	45.640	2988.324	45.640	734.702	45.640	8.065

TABLE 4.7: Computational results for problems with different target values

Dim	E	S	D	NCA		LCA		GA	
				Value	CPU (s)	Value	CPU (s)	Value	CPU (s)
5×5	3,6,20,24	5,16,18	21	62,431	25,413	62,431	4,462	62,431	0,484
	2,4,16,20,22,24	18,19	5	85,548	30,030	85,548	3,650	85,548	0,328
	2,6,11,16,22,24	19,21	1,18	91,028	24,726	91,028	3,058	91,028	0,296
	2,6,10,11,24	5	7,14,16,22	74,237	17,831	74,237	2,200	74,237	0,140
	2,3,16,15,16,23	22	1,8,10,24	84,419	40,732	84,419	1,919	84,419	0,000
6×6	2,15,16,21,22	1,4	17	74,237	11,029	74,237	4,789	74,237	0,312
	1,4,16,16,21	7	24	84,419	17,878	84,419	9,781	84,419	0,000
	1,2,3,5,20,21	8,12,14	22	84,419	31,637	84,419	12,262	84,419	0,000
	1,15,16,24,25	7	11,20	73,709	17,129	73,709	5,694	73,709	0,156
	1,15,16,20,21,22	3,5,24	12	58,175	40,045	58,175	7,114	59,613	0,499
7×7	7,18,31	26	6,17,27,30	73,709	100,636	73,709	28,283	73,709	0,468
	1,2,18	29	5,15,19,28	56,765	221,334	56,765	97,594	56,765	0,452
	3,5,7,18,32,34	14,15	6,8,30	71,233	134,785	71,233	39,999	71,233	0,983
	4,30,33	19,24,32	1,6,20,35	71,270	196,530	71,270	65,910	71,270	1,420
	5,7,24,32,36	4,18,22	14	82,632	118,280	82,632	32,167	82,632	1,498
8×8	19,35,36	16,23	3,10	53,877	260,022	53,877	10,249	53,877	0,983
	2,3,25,33	8	4,11,13,20,29	84,419	69,077	84,419	7,270	84,419	0,484
	1,32,35	5,25,36	28	67,569	126,626	67,569	33,899	67,909	1,451
	1,4,24,32,34,36	2,3,18	25	86,237	142,304	86,237	213,035	86,577	1,560
	2,4,6,31,35	22,27	5,9,11,19,20,29	55,781	259,866	55,781	11,216	55,781	0,858
9×9	2,5,45	29	22,33,35,37	49,034	280,989	49,034	222,177	49,034	1,342
	3,5,6,21,36,43	9,18,32	11	73,709	84,568	73,709	99,591	73,709	4,243
	3,5,21	16	6,9,10,14,15,29,46	46,799	541,838	46,799	620,978	46,799	1,310
	14,21,28,45,48	17,34,37	32,41	63,655	181,554	63,655	291,129	63,655	4,087
	29,36,48	11,21	1,20,24,30,35,42,43	44,839	1924,787	44,839	322,314	44,839	2,605
10×10	1,14,15,35	30,45,47	5,6,7,13,16,19,33	55,372	1836,974	55,372	1290,768	55,372	3,884
	14,22,43,44	20,45	7,10,19,28,30	79,628	211,288	79,628	53,212	79,628	2,652
	3,5,28,36,49	18,29	20,30,40,43	69,446	176,281	69,446	126,954	69,446	2,683
	3,5,28,29,35,47	42	7,11,37	74,237	257,573	74,237	32,542	74,237	1,373
	2,5,21,35,44,45	13	8,17,25	78,940	139,855	78,940	104,099	78,940	1,404
11×11	5,17,57,59,60,63	43	2,3,9,16,18,29,39	61,745	867,178	61,745	181,772	61,745	3,510
	1,3,6,64	4,32,53	27	76,713	724,656	76,713	1318,973	76,713	10,998
	6,41,60	36,45	20,21,29,46,53,56	51,725	9119,912	51,725	1771,423	51,725	6,911
	2,17,40,57,62	21,22	10,34,35,42,48,51,56,64	48,193	451,124	48,193	188,933	48,193	6,926
	8,41,57	15	4,6,19,26,46,64	73,709	530,419	73,709	238,026	73,709	3,557
12×12	2,33,56	26,53,63	21,24	73,709	5139,422	73,709	591,571	73,709	10,748
	6,25,32,48	55	12,17,36,57	73,709	1298,116	73,709	484,758	73,709	3,526
	1,3,5,48,57,64	38,63	4,11,15,17,29,34,36,42,62	69,446	2648,086	69,446	332,126	69,446	6,848
	1,25,40,41,60,64	58,61	4,5,21,34,35,39,54,56	71,922	624,519	71,922	157,218	71,922	6,880
	1,18,45,77,79,81	4	3,24,34,35,40,45,51,54,60,64	74,237	996,347	74,237	753,360	74,237	3,401
13×13	4,18,46,75	70	3,6,10,13,29,35,43,44,49,55,61,62,75,78	74,237	1524,785	74,237	1131,428	74,237	8,018
	5,7,28	43,80	13,54,66,73	47,136	25163,149	47,136	22919,308	46,656	16,677
	18,27,54,79	36	6,10,14,16,25,27,29,30,45,49,53,57,60,67	44,704	4101,703	44,704	1665,981	44,706	7,956
	4,9,75	53	11,22,59	74,237	8344,197	74,237	3151,298	74,237	8,518
	3,5,7,10,64,75	55,60,80	19,38	52,307	28989,650	52,307	9476,998	52,307	25,210
14×14	9,19,63,77,78,79	44	13,47	45,219	16478,651	45,219	3628,864	45,219	8,533
	10,54,79	32	2,15,21,24,31,34,48,56,57,60,71,75,81	45,106	3362,711	45,106	1208,228	45,106	8,003
	1,37,80	6,16,45	8,13,28,35,47,50,78	54,258	14273,483	54,258	11939,147	54,258	25,210
	9,10,18,27,73,75	63	6,8,9,48,49,70,71,76	49,865	11108,535	49,865	2695,432	49,865	8,096
		6,32,72	35	50,663	21634,109	50,663	10718,470	52,206	25,725

Chapter 5

Summary and Future work

Port and maritime security problems have become more challenging since the number and diversity of maritime threats have increased. Designing efficient plans to protect ports and waterways lies ahead and needs a long way to go. In this thesis, we restrict our attention to developing efficient surveillance systems against a small vessel attack threat. In this chapter, we summarize our findings and explore possible future directions based on our works.

5.1 Summary

Although there have been several studies to design surveillance systems for port security, to our knowledge, few works address surveillance plans against a small vessel attack threat. This thesis focuses on how to deploy detectors to protect maritime targets in ports from terrorist attacks using small vessels as waterborne improvised explosive devices and investigates models, properties, and solution techniques. The

aim of our effort is to minimize the expected damage caused by the threat with limited resources.

We first present the research background and point out the research scope in Chapter 1. Chapter 2 provides an overview of previous relevant works from which we obtain our research topics and the associated methodologies. The literature review is divided into four categories: 1) port security, 2) sensor deployment for surveillance, 3) game theory, and 4) relevant algorithms and techniques. We summarize the research findings for port security and find relatively little research on surveillance plans for a small vessel attack threat in a maritime area. Furthermore, we go over surveillance models in other fields, such as airport security and network security, and discover some research gaps to help us develop novel strategies to fit our problems. For example, most previous surveillance models for sensor deployment problems only consider the evasion probabilities of attackers. We incorporate target values to build novel plans for port security. In a word, the findings of the previous studies provide a platform for our research.

In Chapter 3, we investigate how to effectively deploy different kinds of detectors under a small vessel attack threat to port security. We characterize a concerned area as a rectangle with entrances that are along the boundary, obstructions that a small boat can not travel through, and targets that are attractive to the terrorist. Detectors are not perfectly reliable and the probability of detection depends on both the detector characteristics (detection radius and instantaneous detection rate) and how long the small vessel would stay in the detector's effective detection

area. Each detector separately detects whether a small boat is loaded with explosive materials. The terrorist selects a specific entrance and target combination in a probabilistic way. Under this problem setting, we propose a nonlinear binary integer program to enable us to decide what types of detectors to utilize, how many for each kind to deploy, and where to site the different detectors.

Due to the model complexity, we conduct theoretical analysis on the formulation to establish viable solution techniques. We introduce two dominance definitions (strong dominance and weak dominance) and explore some dominance properties for detector deployment to reduce the size of the program and save computational costs. To get an exact solution, firstly, we use a standard branch and bound algorithm to solve the nonlinear binary integer program. However, this approach has expensive computational costs since it repeatedly solves nonlinear programs. Hence, we use a substitution strategy to get another detector placement problem formulation. We utilize a polyhedral outer approximation technique to develop a linearized branch and bound algorithm to solve the new formulation. Based on a detailed computational study, we observe that the linearized branch and bound algorithm is preferred to the standard one. In order to solve large-size problems efficiently, a greedy adding heuristic is developed and performs well as verified by our numerical study.

There is an implicit assumption in our first study that the terrorist is blind with respect to the locations of detectors. This assumption is not realistic because the adversary will presumably do all he can to determine his attacking scheme. Hence, in our second work, we investigate designing a defensive scheme from a

game-theoretic perspective. We apply Stackelberg game theory to the problem and establish a defender-attacker model between a government who seeks to locate detectors to minimize the expected damage for the worst case and a terrorist group who wants to choose paths to launch attacks under the condition that the locations of detectors are completely known. Since we consider not only the probabilities of evasion but also the values of targets, we formulate this problem as a path-based bi-level nonlinear integer program with a nonlinear binary program in the upper level and a linear binary program in the lower level.

Such a bi-level nonlinear integer program is not a standard optimization problem which can be solved by general methods. In order to solve the bi-level nonlinear integer program exactly and efficiently, we use cutting-plane methods. In the process of the methods, the nonlinear binary program in the upper level and the large-size linear binary program in the lower level are repeatedly solved until a solution within required accuracy is obtained. For the nonlinear binary program, we propose a standard branch and bound algorithm and a linearized branch and bound algorithm to solve it. It is unacceptable to directly solve the large-size linear binary program because the number of decision variables is too large to be listed. We use a branch-and-pricing method to obtain an exact solution fast for the linear binary program. In short, two exact cutting-plane algorithms (a linearized cutting-plane algorithm and a nonlinear cutting-plane algorithm) are developed for our model. Furthermore, we develop a greedy heuristic for large-size problems. Then, we conduct a case study of the port of New York and New Jersey to apply our model and algorithms and get insights on the model. Moreover, we

conduct experiments to test the performance of the proposed algorithms. The numerical results show the preference of the linearized cutting-plane algorithm over the nonlinear cutting-plane algorithm and the efficiency and effectiveness of the greedy algorithm.

In a word, in this thesis, we make some contributions to port security by providing models and the associated solution techniques to protect ports from a small boat attack threat.

5.2 Future Work

It is noted that there are several basic assumptions in our models, for instance, detectors work independently, detectors are stationary, and terrorists get full information of detector placement. These assumptions make our mathematical models easier, but cannot be applicable in some situations. Some important challenging works must be addressed by relaxing these assumptions.

Joint Detection Issue. One assumption in our models is that detectors work independently, that is, each detector separately declares a passing small vessel as either carrying explosive materials or not. This assumption is reasonable if the effective detection area of each placed detector does not substantially intersect. For the case where these areas have significant intersection, joint detection probability should be considered. There are some literature taking care of the joint responses of devices in an airport security checking system [74]. In that framework, the declaration of threat/no-threat of a passenger is based on the passenger scores at

all checking devices he/she goes through. Techniques such as classification theory and data fusion may be useful for this issue.

Mobile Detector Deployment. Another assumption in our works is that detectors are stationary. Due to the vastity of a port area, the cost of deploying a large amount of static detectors is daunting. Recent technological advances have enabled the design and deployment of cost-effective mobile detectors in military and civil applications. For example, sodium iodide detectors have been placed in police cars, fire trucks, buses, and taxi cabs to detect nuclear threats in city areas [40]. Similarly, in a port area, mobile detectors could be attached to patrol boats, leisure boats, or even unmanned aerial vehicles. If mobile detectors are to be deployed, one critical research task involves how to plan patrol routes for them in order to achieve effective detection. There are plenty of research works on deploying mobile sensors, but most of them focus on a coverage perspective (see, for example, [56, 111]). One basic limitation of coverage models is that each location in the area is treated in the same way. This is not the case for a situation where the major focus is on protecting crucial targets from small vessel attacks. How to deploy mobile detectors in such a situation becomes challenging. We expect to develop novel models and algorithms to take care of this. Another interesting research issue is related to how to deploy both mobile and static detectors to improve port security.

Information Asymmetry. In our second work, we assume that terrorists can obtain full information of the locations of detectors. This is not realistic since

information asymmetry always occurs for detector deployment problems. For example, it is very difficult to acquire the information of defensive schemes. In such a situation, attackers will take action with partial or no information. Because terrorists may not know exactly the locations of detectors, we plan to formulate an imperfect information Stackelberg game between a government and a terrorist group. The government first decides a detector deployment strategy to minimize the expected damage. The terrorist group, after obtaining some partial information via intelligence, decides an attacking strategy. Besides maximizing the expected damage, the terrorist group may consider minimizing the probability of being detected as another objective. Furthermore, some specific algorithms will be extensively explored.

Appendix A

Appendix for Chapter 3

Proof of Theorem 3.1.

Before we prove Theorem 3.1, we introduce some notations. For every (k, j) pair, the portion of path $\overline{P}_{kj}$ within the circle centered at grid u with radius τ_{h_1} has at most two line segments. Moreover, for one line segment and two line segments with their intersection point not on the circle boundary cases, there are either two or four intersections with the circle. If the intersection point of two segments is on the circle boundary, there are three intersections with the circle (including the intersection point of the two segments). There are three possible patterns as shown in Figure A.1. Pattern 1 indicates one line segment, Pattern 2 indicates two line segments and their intersection point is either outside the circle or on the circle boundary, and Pattern 3 indicates two line segments and the intersection is within the circle. For Patterns 2 and 3, we use c_{ukj1} and c_{ukj2} to represent the two distances from grid u to the lines passing through the two segments. For

convenience, we label the shorter one with subscript 1, that is, $c_{ukj1} \leq c_{ukj2}$. For Pattern 1, we have that $c_{ukj1} = c_{ukj2}$. Following these notations, the condition in (ii) of Theorem 3.1 becomes $c_{ukj2} \leq c_{vkj1}$.

FIGURE A.1: Possible patterns and the associated distances.

If there are two line segments within the circle, there are two perpendicular lines from the grid center to the lines passing through the two segments. If the vertical point is on the line segment, we claim that the distance from the vertical point to the intersection point is positive. Otherwise, if the vertical point is not on the line segment (on the extension of the segment), we claim that the distance is negative. For grid i , define the algebraic sum of these two distances as d_i . Three possible structures for d_i are shown in Figure A.2. From the figure, $d_u = d_{u_1} + d_{u_2}$ is the sum of two positive distances, $d_v = -(d_{v_1} + d_{v_2})$ is the sum of two negative distances, and $d_w = d_{w_1} - d_{w_2}$ is the sum of a positive and a negative distances.

Proof. From Definition 3.2, we have that $p_{h_1ukj} \geq p_{h_1vkj}$ for all (k, j) pairs. Based on the formula $p_{h_1ikj} = 1 - e^{-\eta_{h_1} l_{h_1ikj}}$, we obtain that $l_{h_1ukj} \geq l_{h_1vkj}$ for all (k, j) pairs. Our target is to show that $l_{hukj} \geq l_{hvkj}$ for each (k, j) pair and each type h whose radius is smaller than τ_{h_1} . We note that if $\tau_h \leq c_{vkj1}$, $l_{hvkj} = 0$ but $l_{hukj} \geq 0$.

FIGURE A.2: Possible structures for d_i .

It is obvious that $l_{hukj} \geq l_{hvkj}$. Therefore, we focus on all types h such that $\tau_h > c_{v kj1}$. Furthermore, if the condition (i) of Theorem 3.1 is true, it is obvious that $l_{hukj} \geq l_{hvkj}$. Hence, we will concentrate on the situations where there are either one or two line segments within the circles centered at grids u and v with radius τ_{h_1} .

From Figure A.1, length $l_{h_1 i k j}$ can be calculated by the following formula:

$$l_{h_1 i k j} = \sqrt{\tau_{h_1}^2 - c_{i k j 1}^2} + \sqrt{\tau_{h_1}^2 - c_{i k j 2}^2} + \varsigma_i, \quad \forall(k, j) \text{ and } i = u, v, \quad (\text{A.1})$$

where $\varsigma_i = 0$ for Pattern 1, $\varsigma_i = \sqrt{\tau_{h_1}^2 - c_{i k j 1}^2} + \sqrt{\tau_{h_1}^2 - c_{i k j 2}^2}$ for Pattern 2, and $\varsigma_i = d_i$ for Pattern 3.

Based on the length of detection radius τ_h , we consider the following two cases:

Case 1. $\tau_h \geq c_{v kj2}$;

Case 2. $c_{v kj1} < c_{v kj2}$ and $c_{v kj1} < \tau_h < c_{v kj2}$.

Case 1. $\tau_h \geq c_{v kj2}$.

For this case, based on different patterns, we have the following four subcases:

Case 1.1. $\varsigma_u = 0$;

Case 1.2. $\varsigma_u = \sqrt{\tau_{h_1}^2 - c_{ukj1}^2} + \sqrt{\tau_{h_1}^2 - c_{ukj2}^2}$ or $\varsigma_u = d_u$ and $d_u \geq \sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2}$;

Case 1.3. $\varsigma_u = d_u$ and $|d_u| < \sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2}$;

Case 1.4. $\varsigma_u = d_u$ and $d_u \leq -(\sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2})$.

Case 1.1. $\varsigma_u = 0$.

For this subcase, as in Pattern 1 in Figure A.1, the portion of $\bar{P}_{kj}$ within the circle centered at grid u with radius τ_h has one line segment. Lengths l_{hukj} and l_{hvkj} satisfy the following two expressions:

$$l_{hukj} = \sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2} \text{ and } l_{hvkj} \leq \sqrt{\tau_h^2 - c_{vkj1}^2} + \sqrt{\tau_h^2 - c_{vkj2}^2} + \varsigma_v.$$

From (A.1) and the fact that $l_{h_1ukj} \geq l_{h_1vkj}$, we obtain that

$$g(\tau_{h_1}) - \varsigma_v \geq 0,$$

where

$$g(\tau) = \frac{c_{vkj1}^2 - c_{ukj1}^2}{\sqrt{\tau^2 - c_{ukj1}^2} + \sqrt{\tau^2 - c_{vkj1}^2}} + \frac{c_{vkj2}^2 - c_{ukj2}^2}{\sqrt{\tau^2 - c_{ukj2}^2} + \sqrt{\tau^2 - c_{vkj2}^2}} \quad (\text{A.2})$$

is a monotonically decreasing function. Hence,

$$l_{hukj} - l_{hvkj} \geq g(\tau_h) - \varsigma_v \geq g(\tau_{h_1}) - \varsigma_v \geq 0.$$

Case 1.2. $\varsigma_u = \sqrt{\tau_{h_1}^2 - c_{ukj1}^2} + \sqrt{\tau_{h_1}^2 - c_{ukj2}^2}$ or $\varsigma_u = d_u$ and $d_u \geq \sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2}$.

For this subcase, as in Pattern 2 in Figure A.1, the portion of $\bar{P}_{kj}$ within the circle centered at grid u with radius τ_h has two line segments and the intersection point is either outside the circle or on the circle boundary. Lengths l_{hukj} and l_{hvkj} satisfy the following two expressions:

$$l_{hukj} = 2\sqrt{\tau_h^2 - c_{ukj1}^2} + 2\sqrt{\tau_h^2 - c_{ukj2}^2} \text{ and } l_{hvkj} \leq 2\sqrt{\tau_h^2 - c_{vkj1}^2} + 2\sqrt{\tau_h^2 - c_{vkj2}^2}.$$

Since $c_{ukj1} \leq c_{vkj1}$ and $c_{ukj2} \leq c_{vkj2}$, we have that $l_{hukj} \geq l_{hvkj}$.

Case 1.3. $\varsigma_u = d_u$ and $|d_u| < \sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2}$.

For this subcase, as in Pattern 3 in Figure A.1, the portion of $\bar{P}_{kj}$ within the circle centered at grid u with radius τ_h has two line segments and the intersection point is within the circle. Lengths l_{hukj} and l_{hvkj} satisfy the following two expressions:

$$l_{hukj} = \sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2} + \varsigma_u \text{ and } l_{hvkj} \leq \sqrt{\tau_h^2 - c_{vkj1}^2} + \sqrt{\tau_h^2 - c_{vkj2}^2} + \varsigma_v.$$

From (A.1) and the fact that $l_{h_1ukj} \geq l_{h_1vkj}$, we have that

$$g(\tau_{h_1}) + \varsigma_u - \varsigma_v \geq 0,$$

where $g(\tau)$ is defined by (A.2). Therefore,

$$l_{hukj} - l_{hvkj} \geq g(\tau_h) + \varsigma_u - \varsigma_v \geq g(\tau_{h_1}) + \varsigma_u - \varsigma_v \geq 0.$$

Case 1.4. $\varsigma_u = d_u$ and $d_u \leq -(\sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2})$.

For this subcase, there is no portion of path $\bar{P}_{kj}$ within the circle centered at grid u with radius τ_h . For this subcase, $l_{hukj} = 0$. At the same time, we must have that $l_{hvkj} = 0$. Suppose that it is not true, that is, $l_{hvkj} > 0$. $l_{hukj} = 0$ means that $\sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2} + \varsigma_u \leq 0$, while $l_{hvkj} > 0$ means that $\sqrt{\tau_h^2 - c_{vkj1}^2} + \sqrt{\tau_h^2 - c_{vkj2}^2} + \varsigma_v > 0$. It follows that

$$\sqrt{\tau_h^2 - c_{ukj1}^2} + \sqrt{\tau_h^2 - c_{ukj2}^2} + \varsigma_u - \sqrt{\tau_h^2 - c_{vkj1}^2} - \sqrt{\tau_h^2 - c_{vkj2}^2} - \varsigma_v < 0,$$

that is,

$$g(\tau_h) + \varsigma_u - \varsigma_v < 0.$$

Furthermore,

$$l_{h_1ukj} - l_{h_1vkj} = g(\tau_{h_1}) + \varsigma_u - \varsigma_v \leq g(\tau_h) + \varsigma_u - \varsigma_v < 0,$$

which contradicts with the fact that $l_{h_1ukj} \geq l_{h_1vkj}$.

Case 2. $c_{vkj1} < c_{vkj2}$ and $c_{vkj1} < \tau_h < c_{vkj2}$.

From the proof process for Case 1, there is a type e with $\tau_e = c_{vkj2}$ and $l_{eukj} \geq l_{evkj}$ for all (k, j) pairs. Hence, in this case, our proof can be converted to show that $l_{hukj} \geq l_{hvkj}$ for all (k, j) pairs and all types h such that $c_{vkj1} < \tau_h < c_{vkj2}$ under the condition that $l_{eukj} \geq l_{evkj}$, which is a special case of Case 1.

Combining Case 1 and Case 2, we obtain that $l_{hukj} \geq l_{hvkj}$, which implies that $p_{hukj} \geq p_{hvkj}$ for all (k, j) pairs. Hence, grid u strongly dominates grid v . $\square$

Proof of Theorem 3.2.

Before we prove Theorem 3.2, we provide the following lemma.

Lemma A.1. Let $N_{kj} = \cup_h N_{kj}^h$ be the potential detector location set for path P_{kj} . If grid u strongly dominates grid v , we have that

- (i) $\{h : u \in T^h\} \supseteq \{h : v \in T^h\}$;
- (ii) $u \notin N_{kj}$ implies that $v \notin N_{kj}$.

Proof. (i) (By contradiction) If $\{h : u \in T^h\} \not\supseteq \{h : v \in T^h\}$, there is a type b with $v \in T^b$ and $u \notin T^b$. From the construction of T^b , there is a path $P_{k'j'}$ such that $p_{bv k'j'} > 0$ and $p_{bu k'j'} = 0$. It contradicts with the fact that grid u strongly dominates grid v .

(ii) (By contradiction) If $v \in N_{kj}$, by the construction of N_{kj} , we have that $p_{bv kj} > 0$ for some type b . On the other hand, $u \notin N_{kj}$ implies that $p_{hu kj} = 0$ for every type h . It contradicts with the fact that grid u strongly dominates grid v . $\square$

The proof of Theorem 3.2 is as follows:

Proof. (By contradiction) Suppose that $\sum_{h:u \in T^h} x_{hu}^* < \sum_{h:v \in T^h} x_{hv}^*$ in an optimal solution X^* . Since both sides of the strict inequality are binary, we have that

$\sum_{h:u \in T^h} x_{hu}^* = 0$ and $\sum_{h:v \in T^h} x_{hv}^* = 1$. Moreover, because all decision variables are binary, we have that $x_{hu}^* = 0$ for $h \in \{h : u \in T^h\}$ and there exists a type $b \in \{h : v \in T^h\}$ such that $x_{bv}^* = 1$ and $x_{hv}^* = 0$ for $h \in \{h : v \in T^h\} \setminus \{b\}$.

From the conclusion (i) of Lemma A.1, we have that $b \in \{h : u \in T^h\}$. Hence, we can find another feasible solution X^{**} with $x_{bu}^{**} = 1$, $x_{bv}^{**} = 0$, and other values

being the same as those of X^* . From the conclusion (ii) of Lemma A.1, we observe that there is no path P_{kj} such that $u \notin N_{kj}$ but $v \in N_{kj}$. Hence, the objective function can be rewritten in the following form:

$$\begin{aligned}
f(X) &= \sum_{\substack{kj: u \notin N_{kj} \\ v \notin N_{kj}}} W_{kj} \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}} \\
&+ \sum_{\substack{kj: u \in N_{kj} \\ v \in N_{kj}}} W_{kj} \prod_{h \in H} \prod_{\substack{i \in N_{kj}^h \\ i \neq u, v}} (1 - p_{hikj})^{x_{hi}} \prod_{h: u \in N_{kj}^h} (1 - p_{hukj})^{x_{hu}} \prod_{h: v \in N_{kj}^h} (1 - p_{hvkj})^{x_{hv}} \\
&+ \sum_{\substack{kj: u \in N_{kj} \\ v \notin N_{kj}}} W_{kj} \prod_{h \in H} \prod_{\substack{i \in N_{kj}^h \\ i \neq u}} (1 - p_{hikj})^{x_{hi}} \prod_{h: u \in N_{kj}^h} (1 - p_{hukj})^{x_{hu}}.
\end{aligned}$$

For convenience, we let

$$I = \sum_{\substack{kj: u \notin N_{kj} \\ v \notin N_{kj}}} W_{kj} \prod_{h \in H} \prod_{i \in N_{kj}^h} (1 - p_{hikj})^{x_{hi}},$$

$$J_{kj} = W_{kj} \prod_{h \in H} \prod_{\substack{i \in N_{kj}^h \\ i \neq u, v}} (1 - p_{hikj})^{x_{hi}},$$

and

$$K_{kj} = W_{kj} \prod_{h \in H} \prod_{\substack{i \in N_{kj}^h \\ i \neq u}} (1 - p_{hikj})^{x_{hi}}.$$

Then, the objective function values of X^* and X^{**} are as follows:

$$f(X^*) = I + \sum_{\substack{kj: u \in N_{kj} \\ v \in N_{kj}}} J_{kj} (1 - p_{bvkj}) + \sum_{\substack{kj: u \in N_{kj} \\ v \notin N_{kj}}} K_{kj} \quad (\text{A.3})$$

and

$$f(X^{**}) = I + \sum_{\substack{kj: u \in N_{kj} \\ v \in N_{kj}}} J_{kj}(1 - p_{bukj}) + \sum_{\substack{kj: u \in N_{kj} \\ v \notin N_{kj}}} K_{kj}(1 - p_{bukj}). \quad (\text{A.4})$$

Based on (A.3) and (A.4), we observe that the objective function value of X^{**} is smaller than that of X^* . It contradicts with the fact that X^* is an optimal solution. $\square$

Proof of Theorem 3.3.

Proof. The proof procedure is similar to that of Theorem 17 in [87] except that $\mathbf{X}$ -part decision variables are integer in our algorithm while they are continuous in [87]. Therefore, we need to prove that there is a convergent subsequence for the solution sequence generated by the linearized branch and bound algorithm in infinite iterations. For each node $n(s)$, let $\mathbf{\Gamma}^{n(s)} = (\mathbf{X}^{n(s)}, \mathbf{Y}^{n(s)}, \mathbf{Z}^{n(s)})$ be an optimal solution of $[\text{DP}(\mathbf{L}^{n(s)}, \mathbf{U}^{n(s)})\text{-R}]$. Because $\mathbf{X}$ -variable parts are finite in any infinite branch, there is a set $N_1 \subseteq N$ such that

$$\lim_{\substack{s \rightarrow \infty \\ s \in N_1}} \mathbf{X}^{n(s)} = \mathbf{X}^*. \quad (\text{A.5})$$

Furthermore, because the feasible regions for $\{\mathbf{Y}^{n(s)}, \mathbf{Z}^{n(s)}, \mathbf{L}^{n(s)}, \mathbf{U}^{n(s)}\}_{s \in N_1}$ are compact sets, there is a set $N_2 \subseteq N_1$ such that

$$\lim_{\substack{s \rightarrow \infty \\ s \in N_2}} (\mathbf{Y}^{n(s)}, \mathbf{Z}^{n(s)}, \mathbf{L}^{n(s)}, \mathbf{U}^{n(s)}) = (\mathbf{Y}^*, \mathbf{Z}^*, \mathbf{L}^*, \mathbf{U}^*). \quad (\text{A.6})$$

Based on (A.5) and (A.6), we obtain a convergent subsequence. The remaining process is identical to that in [87]. $\square$

Bibliography

- [1] Adams, W.P. and H.D. Sherali, A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems, *Management Science*, 32(10): 1274-1290, 1986.
- [2] Agmon, N., S. Kraus, G.A. Kaminka, and V. Sadov, Adversarial Uncertainty in Multi-Robot Patrol, *Proceedings of the Twenty-First International Joint Conferences on Artificial Intelligence*, 1811-1817, 2009.
- [3] Alpcan, T. and T. Basar, A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection, *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2595-2600, 2003.
- [4] Anand, S., D. Madigan, R. Mammone, S. Pathak, and F. Roberts, Experimental Analysis of Sequential Decision Making Algorithms for Port of Entry Inspection Procedures, in: S. Mehrotra, D. Zeng, H. Chen, B. Thuraisingham, and F.X. Wang (Editors), *Intelligence and security informatics, Proceedings of ISI, Lecture Notes in Computer Science 3975*, Springer-Verlag, New York, 2006.

- [5] Atkinson, M.P., Z. Cao, and L.M. Wein, Optimal Stopping Analysis of a Radiation Detection System to Protect Cities from a Nuclear Terrorist Attack, *Risk Analysis*, 28(2): 353-371, 2008.
- [6] Atkinson, M.P. and L.M. Wein, Spatial Queueing Analysis of an Interdiction System to Protect Cities from a Nuclear Terrorist Attack, *Operations Research*, 56(1): 247-254, 2008.
- [7] Bai, X., S. Li, and J. Xu, Mobile Sensor Deployment Optimization for k-coverage in Wireless Sensor Networks with a Limited Mobility Model, *IETE Technical Review*, 27(2): 124-137, 2010.
- [8] Basilico, N., N. Gatti, and F. Amigoni, Leader-Follower Strategies for Robotic Patrolling in Environments with Arbitrary Topologies, *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, 57-64, 2009.
- [9] Ben-Zvi, T. and J. Nickerson, Sensor Placement Learning Automata, *Proceedings of the INFORMS International Conference*, Puerto Rico, July, 2007.
- [10] Berman, O. and A. Gavious, Location of Terror Response Facilities: A Game between State and Terrorist, *European Journal of Operational Research*, 177(2): 1113-1133, 2007.
- [11] Bier, V., S. Oliveros, and L. Samuelson, Choosing What to Protect: Strategic Defensive Allocation against an Unknown Attacker, *Journal of Public Economic Theory*, 9(4): 563-587, 2007.

- [12] Blankenship, J.W. and J.E. Falk, Infinitely constrained optimization problems, *Journal of Optimization Theory and Applications*, 19(2): 261-281, 1976.
- [13] Boros, E., L. Fedzhora, P.B. Kantor, K. Saeger, and P. Stroud, A Large-Scale Linear Programming Model for Finding Optimal Container Inspection Strategies, *Naval Research Logistics*, 56(5): 404-420, 2009.
- [14] Brown, G., M. Carlyle, A. Abdul-Ghaffar, and J. Kline, A Defender-Attacker Optimization of Port Radar Surveillance, *Naval Research Logistics*, 58(3): 223-235, 2011.
- [15] Brown, G., M. Carlyle, J. Salmern, and K. Wood, Defending Critical Infrastructure, *Interfaces*, 36(6): 530-544, 2006.
- [16] Carafano, J.J. and A. Kochems, Making the Sea Safer: A National Agenda for Maritime Security and Counterterrorism, The Heritage Foundation, Washington, D.C., February 17, 2005.
- [17] Cavalier, T.M., W.A. Conner, E. Del Castillo, and S.I. Brown, A Heuristic Algorithm for Minimax Sensor Location in the Plane, *European Journal of Operational Research*, 183(1): 42-55, 2007.
- [18] CBS News, U.S. Wary of Small Boat Terrorism, Washington, D.C., April 28, 2008.
- [19] Chakrabarty, K., S.S. Iyengar, H. Qi, and E. Cho, Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks, *IEEE Transactions on Computers*, 51(12): 1448-1453, 2002.

- [20] Chalk, P., The Maritime Dimension of International Security: Terrorism, Piracy, and Challenges for the United States, Rand Corporation, Available at: http://www.rand.org/pubs/monographs/2008/RAND_MG697.pdf, 2008.
- [21] Chang, S.J., Vessel Identification and Monitoring Systems for Maritime Security, *IEEE the 37th Annual 2003 International Carnahan Conference on Security Technology*, 66-70, 2003.
- [22] Concho, A.L. and J.E. Ramirez-Marquez, An Evolutionary Algorithm for Port-of-Entry Security Optimization Considering Sensor Thresholds, *Reliability Engineering and System Safety*, 95(3): 255-266, 2010.
- [23] Dakin, R.J., A Tree-Search Algorithm for Mixed Integer Programming Problems, *The Computer Journal*, 8(3): 250-255, 1965.
- [24] Department of Homeland Security, Small Vessel Security Strategy, April 2008.
- [25] Dijkstra, E.W., A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, 1(1), 269-271.
- [26] Elsayed, E.A., C.M. Young, M. Xie, H. Zhang, and Y. Zhu, Port-of-Entry Inspection: Sensor Deployment Policy Optimization, *IEEE Transactions on Automation Science and Engineering*, 6(2): 265-276, 2009.
- [27] Falk, J.E. and R.M. Soland, An Algorithm for Separable Nonconvex Programming Problems, *Management Science*, 15(9): 550-569, 1969.
- [28] Fang, F., A.X. Jiang, and M. Tambe, Optimal Patrol Strategy for Protecting Moving Targets with Multiple Mobile Resources, *Proceedings of the 2013*

- International Conference on Autonomous Agents and Multi-Agent Systems*, 957-964, 2013.
- [29] Ferrier, D., K. Pysareva, and A. Rucinski, Using Technology to Bridge Maritime Security Gaps, *Sea Technology*, 46(8): 53-58, 2005.
- [30] Finnegan, W., A Reporter at Large: Watching the Waterfront, *The New Yorker*, August 3, 2012.
- [31] Frank, T., Small Boats Seen as a Terror Threat, *USA Today*, October 31, 2007.
- [32] Gatti, N., Game Theoretical Insights in Strategic Patrolling: Model and Algorithm in Normal-Form, *Proceedings of the 18th European Conference on Artificial Intelligence*, 403-407, 2008.
- [33] Gaudette, K., Court Approves Bush Request to Open West Coast Ports, *The Seattle Times*, October 8, 2002.
- [34] Ghare, P.M., D.C. Montgomery, and W.C. Turner, Optimal Interdiction Policy for a Flow Network, *Naval Research Logistics Quarterly*, 18(1): 37-45, 1971.
- [35] Gilmore, P.C. and R.E. Gomory, A Linear Programming Approach to the Cutting-stock Problem, *Operations Research*, 9(6): 849-859, 1961.
- [36] Gozani, T., The Role of Neutron Based Inspection Techniques in the Post 9/11/01 Era, *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 213: 460-463, 2004.

- [37] Hall, J.M., S. Asztalos, P. Bilotto, J. Church, M.A. Descalle, T. Luu, D. Mantt, G. Mauger, E. Norman, and D. Petersen, The Nuclear Car Wash: Neutron Interrogation of Cargo Containers to Detect Hidden Snm, *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 261(1-2): 337-340, 2007.
- [38] Helmick, J.S., Port and Maritime Security: A Research Perspective, *Journal of Transportation Security*, 1(1): 15-28, 2008.
- [39] Hill, B.P., Maritime Terrorism and the Small Boat Attack Threat to the United States: A Proposed Response, Master Thesis, Naval Postgraduate School, 2009.
- [40] Hochbaum, D.S. and B. Fishbain, Nuclear Threat Detection with Mobile Distributed Sensor Networks, *Annals of Operations Research*, 187(1): 45-63, 2011.
- [41] Hong, N. and A.K. Ng, The International Legal Instruments in Addressing Piracy and Maritime Terrorism: A Critical Review, *Research in Transportation Economics*, 27(1): 51-60, 2010.
- [42] Horst, R. and H. Tuy, Global Optimization: Deterministic Approaches, 3rd Edition, Springer, Berlin, 1996.
- [43] Howard, A., M.J. Mataric, and G.S. Sukhatme, An Incremental Self-Deployment Algorithm for Mobile Sensor Networks, *Autonomous Robots*, 13(2): 113-126, 2002.

- [44] Howard, A., M.J. Mataric, and G.S. Sukhatme, Mobile Sensor Network Deployment Using Potential Fields: A Distributed, Scalable Solution to The Area Coverage Problem, in: H. Asama, T. Arai, T. Fukuda, and T. Hasegawa (Editors), *Distributed Autonomous Robotic Systems 5*, 299-308, Springer, Japan, 2002.
- [45] International Maritime Organization, International Ship & Port Facility Security (ISPS) Code, IMO-I116E, London, 2003.
- [46] Israeli, E. and R.K. Wood, Shortest-Path Network Interdiction, *Networks*, 40(2): 97-111, 2002.
- [47] Jourdan, D.B. and O.L. De Weck, Layout Optimization for a Wireless Sensor Network Using a Multi-Objective Genetic Algorithm, *IEEE 59th Vehicular Technology Conference*, 2466-2470, 2004.
- [48] Kaplan, E.H. and M. Kress, Operational Effectiveness of Suicide-Bomber-Detector Schemes: A Best-Case Analysis, *Proceedings of the National Academy of Sciences of the United States of America*, 102(29): 10399-10404, 2005.
- [49] Kelley, J.E., The Cutting-Plane Method for Solving Convex Programs, *Journal of the Society for Industrial & Applied Mathematics*, 8(4): 703-712, 1960.
- [50] Kharchenko, V. and V. Vasylyev, Application of the Intellectual Decision Making System for Vessel Traffic Control, *14th International Conference on Microwaves, Radar and Wireless Communications*, 639-642, 2002.

- [51] Kompella, S., S. Mao, Y.T. Hou, and H.D. Sherali, On Path Selection and Rate Allocation for Video in Wireless Mesh Networks, *IEEE/ACM Transactions on Networking*, 17(1): 212-224, 2009.
- [52] Kopacz, Z., W. Morgas, and J. Urbanski, The Maritime Safety System, Its Main Components and Elements, *The Journal of Navigation*, 54(2): 199-211, 2001.
- [53] Land, A.H. and A.G. Doig, An Automatic Method for Solving Discrete Programming Problems, *Econometrica*, 28(3): 497-520, 1960.
- [54] Leventhal, T., G. Nemhauser, and L. Trotter, A Column Generation Algorithm for Optimal Traffic Assignment, *Transportation Science*, 7(2): 168-176, 1973.
- [55] Lin, F.Y.S. and P.L. Chiu, A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage-discrimination in Sensor Networks, *IEEE Communications Letters*, 9(1): 43-45, 2005.
- [56] Liu, B., P. Brass, O. Dousse, P. Nain, and D. Towsley, Mobility Improves Coverage of Sensor Networks, *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 300-308, 2005.
- [57] Liu, Y., B.D. Sowerby, and J.R. Tickner, Comparison of Neutron and High-Energy X-ray Dual-Beam Radiography for Air Cargo Inspection, *Applied Radiation and Isotopes*, 66(4): 463-473, 2008.
- [58] Luft, G. and A. Korin, Terrorism Goes to Sea. *Foreign Affairs*, 83: 61-71, 2004.

- [59] Luo, R.C. and O. Chen, Mobile Sensor Node Deployment and Asynchronous Power Management for Wireless Sensor Networks, *IEEE Transactions on Industrial Electronics*, 59(5): 2377-2385, 2012.
- [60] Lye, K.W. and J.M. Wing, Game Strategies in Network Security, *International Journal of Information Security*, 4(1-2): 71-86, 2005.
- [61] Madigan, D., S. Mittal, and F. Roberts, Sequential Decision Making Algorithms for Port of Entry Inspection: Overcoming Computational Challenges, *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, 1-7, 2007.
- [62] Maritime and Port Authority of Singapore, Available at: http://www.mpa.gov.sg/sites/global_navigation/news_center/mpa_news/mpa_news_detail.page?filename=nr140107a.xml, January 7, 2014.
- [63] McLay, L.A., J.D. Lloyd, and E. Niman, Interdicting Nuclear Material on Cargo Containers Using Knapsack Problem Models, *Annals of Operations Research*, 187(1): 185-205, 2011.
- [64] McMasters, A.W. and T.M. Mustin, Optimal Interdiction of a Supply Network, *Naval Research Logistics Quarterly*, 17(3): 261-268, 1970.
- [65] McNeill, J.B., Getting the Small-boat Threat Right, The Heritage Foundation, August 23, 2010.
- [66] Merrick, J.R.W. and L.A. McLay, Is Screening Cargo Containers for Smuggled Nuclear Threats Worthwhile?, *Decision Analysis*, 7(2): 155-171, 2010.

- [67] Moller, B., Piracy, Maritime Terrorism and Naval Strategy, Danish Institute for International Studies, Available at: http://www.diis.dk/graphics/Publications/Reports2009/DIIS_Report_2009-02_%20Piracy_maritime_terrorism_and_naval_strategy.pdf, 2009.
- [68] Molyboha, A. and M. Zabarankin, Stochastic Optimization of Sensor Placement for Diver Detection, *Operations Research*, 60(2): 292-312, 2012.
- [69] Morton, D.P., F. Pan, and K.J. Saeger, Models for Nuclear Smuggling Interdiction, *IIE Transactions*, 39(1): 3-14, 2007.
- [70] Neild, B., The Small Vessel Threat: Nearly a Decade after 9/11, US Ports Are Still Vulnerable, *GlobalPost*, February 9, 2011.
- [71] Nelson, E.S., Maritime Terrorism and Piracy: Existing and Potential Threats, *Global Security Studies*, 3(1): 15-28, 2012.
- [72] Nguyen, K.C., T. Alpcan, and T. Basar, Security Games with Incomplete Information, *Proceedings of the IEEE International Conference on Communications*, 1-6, 2009.
- [73] Nie, X., R. Batta, C.G. Drury, and L. Lin, Optimal Placement of Suicide Bomber Detectors, *Military Operations Research*, 12(2): 65-78, 2007.
- [74] Nie, X., R. Batta, C.G. Drury, and L. Lin, The Impact of Joint Responses of Devices in an Airport Security System, *Risk Analysis*, 29(2): 298-311, 2009.

- [75] Nie, X., R. Batta, and L. Lin, Conceptualization of a Game Theoretic Approach to Air Marshal Scheduling, in: M. Bell, S. H. Hosseinloo, and U. Kanturska (Editors), *Security and Environmental Sustainability of Multimodal Transport*, 213-224, Springer, Netherlands, 2010.
- [76] Orphan, V.J., E. Muenchau, J. Gormley, and R. Richardson, Advanced Gamma Ray Technology for Scanning Cargo Containers, *Applied Radiation and Isotopes*, 63(5): 723-732, 2005.
- [77] Parfomak, P.W. and J. Frittelli, Maritime Security: Potential Terrorist Attacks and Protection Priorities, CRS Report for Congress, January 9, 2007.
- [78] Park, J.H., G. Friedman, and M. Jones, Geographical Feature Sensitive Sensor Placement, *Journal of Parallel and Distributed Computing*, 64(7): 815-825, 2004.
- [79] Pashko, S., A. Molyboha, M. Zabaranin, and S. Gorovyy, Optimal Sensor Placement for Underwater Threat Detection, *Naval Research Logistics*, 55(7): 684-699, 2008.
- [80] Pita, J., M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald, GUARDS: Game Theoretic Security Allocation on A National Scale, *Proceeding of the 10th International Conference on Autonomous Agents and Multiagent Systems*, 37-44, 2011.
- [81] Powell, R., Defending against Terrorist Attacks with Limited Resources, *American Political Science Review*, 101(3): 527-541, 2007.

- [82] Przemieniecki, J.S., Mathematical Methods in Defense Analyses, American Institute of Aeronautics and Astronautics, Virginia, 2000.
- [83] Ramirez-Marquez, J.E., Port-of-Entry Safety via the Reliability Optimization of Container Inspection Strategy through an Evolutionary Approach, *Reliability Engineering and System Safety*, 93(11): 1698-1709, 2008.
- [84] Roell, P., Maritime Terrorism – A Threat to World Trade?, Institut für Strategie-Politik-Sicherheits-und Wirtschaftsberatung, Berlin, 2009.
- [85] Sandler, T. and D.G. Arce, Terrorism and Game Theory, *Simulation & Gaming*, 34(3): 319-337, 2003.
- [86] Schmidt, S., T. Alpcan, S. Albayrak, T. Basar, and A. Mueller, A Malware Detector Placement Game for Intrusion Detection, in: J. Lopez and B. Himmerli (Editors), Critical Information Infrastructures Security, Lecture Notes in Computer Science 5141: 311-326, Springer, Heidelberg, 2008.
- [87] Sherali, H.D., J. Desai, and T.S. Glickman, Allocating Emergency Response Resources to Minimize Risk with Equity Considerations, *American Journal of Mathematical and Management Sciences*, 24(3-4): 367-410, 2004.
- [88] Sherali, H.D., J. Desai, and T.S. Glickman, Optimal Allocation of Risk-Reduction Resources in Event Trees, *Management Science*, 54(7): 1313-1321, 2008.
- [89] Sherali, H. and C. Tuncbilek, A Global Optimization Algorithm for Polynomial Programming Problems Using a Reformulation-Linearization Technique, *Journal of Global Optimization*, 2(1): 101-112, 1992.

- [90] Sherali, H. and C. Tuncbilek, A Reformulation-Convexification Approach for Solving Nonconvex Quadratic Programming Problems, *Journal of Global Optimization*, 7(1): 1-31, 1995.
- [91] Sherali, H. and C. Tuncbilek, New Reformulation Linearization/Convexification Relaxations for Univariate and Multivariate Polynomial Programming Problems, *Operations Research Letters*, 21(1): 1-9, 1997.
- [92] Singapore Customs, Secure Trade Partnership, Available at: <http://www.customs.gov.sg/leftNav/trad/Supply+Chain+Security.htm>, June 21, 2010.
- [93] Singh, S. and M. Singh, Explosives Detection Systems for Aviation Security, *Signal Processing*, 83(1): 31-55, 2003.
- [94] Statista, Number of Pirate Attacks against Ships Worldwide from 2009 to 2014, Available at: <http://www.statista.com/statistics/266292/number-of-pirate-attacks-worldwide-since-2006/>, 2014.
- [95] Stroud, P.D. and K.J. Saeger, Enumeration of Increasing Boolean Expressions and Alternative Digraph Implementations for Diagnostic Applications, *Proceedings of Computer, Communication and Control Technologies*, 328-333, 2003.
- [96] Szechtman, R., M. Kress, K. Lin, and D. Cfir, Models of Sensor Operations for Border Surveillance, *Naval Research Logistics*, 55(1): 27-41, 2008.
- [97] Tsai, J., C. Kiekintveld, F. Ordonez, M. Tambe, and S. Rathi, IRIS – A Tool for Strategic Security Allocation in Transportation Networks, *The Eighth*

- International Conference on Autonomous Agents and Multiagent Systems - Industry Track*, 37-44, 2009.
- [98] United Nations Conference on Trade and Development, Review of Maritime Transport 2013, New York and Geneva, 2013.
- [99] United States Customs and Border Protection, Container Security Initiative, Available at: http://www.cbp.gov/xp/cgov/newsroom/fact_sheets/trade_security/csi.xml, March 23, 2012.
- [100] United States Department of Homeland Security, DHS Announces West Coast Maritime Radiation Detection Project, September 5, 2007.
- [101] United States Department of Homeland Security, Small Vessel Security Strategy, Washington, D.C., April, 2008.
- [102] United States Department of State Publication Office of the Coordinator for Counterterrorism, Country Report on Terrorism 2009, August 2010.
- [103] United States Government Accounting Office, Nation Faces Formidable Challenges in Making New Initiatives Successful, GAO-02-993T, Washington, D.C., August, 2002.
- [104] United States Government Accounting Office, Combating Terrorism, Actions Needed to Improve Force Protection for DOD Deployments through Domestic Seaports, GAO-03-15, Washington, D.C., October, 2002.

- [105] United States Government Accounting Office, Maritime Security: The Safe Port Act: Status and Implementation One Year Later, GAO-08-126T, Washington, D.C., October 30, 2007.
- [106] United States White House Office, The National Strategy for Maritime Security, Available at: <http://georgewbush-whitehouse.archives.gov/homeland/maritime-security.html>, September 20, 2005.
- [107] Van Weele, S.F. and J.E. Ramirez-Marquez, Optimization of Container Inspection Strategy via a Genetic Algorithm, *Annals of Operations Research*, 187(1): 229-247, 2011.
- [108] Vance, P.H., C. Barnhart, E.L. Johnson, and G.L. Nemhauser, Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound, *Computational Optimization and Applications*, 3(2): 111-130, 1994.
- [109] Vaněk, O., Z. Yin, M. Jain, B. Bošanský, M. Tambe, and M. Pěchouček, Game-Theoretic Resource Allocation for Malicious Packet Detection in Computer Networks, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, 2: 905-912, 2012.
- [110] Veinott, A.F., The Supporting Hyperplane Method for Unimodal Programming, *Operations Research*, 15(1): 147-152, 1967.
- [111] Wang, G., G. Cao, P. Berman, and T.F. La Porta, Bidding Protocols for Deploying Mobile Sensors, *IEEE Transactions on Mobile Computing*, 6(5): 563-576, 2007.

- [112] Wang, G., G. Cao, T. La Porta, and W. Zhang, Sensor Relocation in Mobile Sensor Networks, *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 4: 2302-2312, 2005.
- [113] Wang, J. and N. Zhong, Efficient Point Coverage in Wireless Sensor Networks, *Journal of Combinatorial Optimization*, 11(3): 291-304, 2006.
- [114] Washburn, A.R., Two-Person Zero-Sum Games, 4th Edition, Springer, New York, 2003.
- [115] Washburn, A. and K. Wood, Two-person Zero-sum Games for Network Interdiction, *Operations Research*, 43(2): 243-251, 1995.
- [116] Wein, L.M. and M.P. Atkinson, The Last Line of Defense: Designing Radiation Detection-Interdiction Systems to Protect Cities from a Nuclear Terrorist Attack, *IEEE Transactions on Nuclear Science*, 54(3): 654-669, 2007.
- [117] Wein, L.M., A.H. Wilkins, M. Baveja, and S.E. Flynn, Preventing the Importation of Illicit Nuclear Materials in Shipping Containers, *Risk Analysis*, 26(5): 1377-1393, 2006.
- [118] Wilhelm, W.E. and E.I. Gokce, Branch-and-Price Decomposition to Design a Surveillance System for Port and Waterway Security, *IEEE Transactions on Automation Science and Engineering*, 7(2): 316-325, 2010.
- [119] Wollmer, R., Removing Arcs from a Network, *Operations Research*, 12(6): 934-940, 1964.

- [120] Wood, R.K., Deterministic Network Interdiction, *Mathematical and Computer Modelling*, 17(2): 1-18, 1993.
- [121] Wu, X., S. Lei, W. Jin, J. Cho, and S. Lee, Energy-Efficient Deployment of Mobile Sensor Networks by PSO, in: H. Shen, J. Li, M. Li, J. Ni, and W. Wang (Editors), *Advanced Web and Network Technologies, and Applications*, Lecture Notes in Computer Science 3842, 373-382, Springer, Berlin, 2006.
- [122] Yates, J.T., R. Batta, and M. Karwan, An Integrated Vulnerability-Based Detection/Interception Model for the Protection of Regional Infrastructure from Covert Attack, Available at: <http://www.acsu.buffalo.edu/~batta/critical.pdf>, 2010.
- [123] Young, C.M., M. Li, Y. Zhu, M. Xie, E.A. Elsayed, and T. Asamov, Multiobjective Optimization of a Port-of-Entry Inspection Policy, *IEEE Transactions on Automation Science and Engineering*, 7(2): 392-400, 2010.
- [124] Zou, Y. and K. Chakrabarty, Sensor Deployment and Target Localization Based on Virtual Forces, *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, 2: 1293-1303, 2003.
- [125] Zou, Y. and K. Chakrabarty, Sensor Deployment and Target Localization in Distributed Sensor Networks, *ACM Transactions on Embedded Computing Systems*, 3(1): 61-91, 2004.