DECISION ANALYTICS FOR SONAR PLACEMENT TO MITIGATE MARITIME SECURITY RISK

By

AMIR GHAFOORI

A Dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Industrial and Systems Engineering

Written under direction of

Professor Tayfur Altiok (posthumous)

Professor Susan L. Albin

Professor James T. Luxhoj

and approved by

_____

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2013

ABSTRACT OF THE DISSERTATION

Decision Analytics for Sonar Placement to Mitigate Maritime Security Risk

By AMIR GHAFOORI

Dissertation Directors:

Dr. Tayfur Altiok (posthumous)

Dr. Susan Albin

Dr. James Luxhoj

Two optimization methodologies are proposed for sonar placement in ports and waterways to keep the environment under surveillance against security threats. The optimization models are named Probabilistic Risk Model (PRM) and Strategic Risk Model (SRM). The PRM resembles a typical sensor placement problem in the sense that they share some constraints and a similar objective function, yet the PRM integrates a number of features that are specific to our problem. The SRM is a game theoretic model that takes the intelligent actions of the attackers into account.

This study focuses on the attacks that are initiated through the water and are targeting the infrastructures at a port or a waterway. The sonars are placed under the water. They are utilized to detect anomalies such as divers, torpedoes or explosives mounted on the hull of vessels that can be potential sources of a terrorist attack. The proposed models are grid based meaning that a hypothetical two dimensional grid of cells is placed on the environment to discretize it. This process allows us to measure various specifics of different sections of the environment via cells.

The contribution of this study to the literature of maritime security risk is that it is the first study that models the sonar placement problem via game theory in ports and waterways. Moreover, both models address a number of key concepts of sensor placement which are mostly ignored in the literature.

The SRM's advantage over PRM is the integration of the attacker's intelligent factor into the modeling effort. SRM allows the attacker to be intelligent. This approach is translated to a two player game where the opponents seek to maximize their own payoffs. Various models of this two player game are discussed and modeled. The last model which is a general-sum two player game is the most general model and is capable of integrating real world assumptions.

ACKNOWLEDGEMENTS

First and foremost, I offer my sincerest gratitude to my deceased advisor, Dr Tayfur Altiok, may his gentle soul rest in peace. He supported me throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my degree to his encouragement and effort. One simply could not wish for a better or friendlier supervisor.

I would like to extend my gratitude to my thesis advisors, Dr. Susan Albin and Dr. James Luxhoj, for the guidance, encouragement and support they provided to me since they honored me to supervise my thesis. I am also thankful for their support and guidance throughout this thesis research. I also take this opportunity to express my thankfulness to my thesis committee members Dr. Endre Boros, Dr. Myong-K Jeong and Dr. Wang Honggang.

I was really lucky to have the support of synergetic colleagues and friends namely Dr. Alper Ozhan Almaz, Dr. Birnur Ozbas and Ayberk Onur Gokseven at the Laboratory for Port Security (LPS). Their kind help was really significant throughout my research.

My family was a real motivation for me and made me persistent in achieving my goals. I would like to send special thanks to my parents and parents-in-law, brother, sisters-in-law, brother-in-law and my cute niece who supported me with a plethora of energy and good wishes.

My deepest appreciation is reserved for my lovely wife, Niloufar, to whom I owe this thesis. Words are simply not enough to express my thankfulness for her.

This research is supported partially by New Jersey Department of Transportation and South Jersey Port Corporation. However, the contents of this thesis reflect the views of the author who is solely responsible for the accuracy of the facts, analyses, conclusions, and recommendations represented herein.

DEDICATION

This dissertation is dedicated to:

The memory of my adviser, Professor Tayfur Altiok, for the great inspiration he left for me;

My dear parents for their endless love and support;

And my lovely wife, Niloufar, for all her support and accompaniment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

1.1. Introduction

The increasing international trade in containerized cargo, oil, minerals and ores in the last couple of decades has placed significant emphasis on maritime transportation. Annual sea-born trade more than tripled since the 1970s resulting in over 90% of international cargo moving through marine terminals (U.N. 2011). Maritime transportation is economical and in many situations the only means of transportation. Clearly, this trend significantly increased maritime traffic involving varying sizes of vessels and tankers carrying all types of cargo from containers to bulk, liquid and gas of which a significant portion is dangerous cargo. This in turn applies significant pressure on ports to improve their logistics operations for acceptable vessel waiting times as well as the safety and security of cargo, vessels, port infrastructure and the people living in the port communities.

Furthermore, the global economy has become vulnerable to shutdowns of the transportation system. Any stoppage or appreciable slow-down in port operations impact numerous supply chains, with extended stoppages resulting in some vessels being diverted to other ports, as was the case in the 2003 strike of longshoremen in the port complex of Los Angeles/Long Beach. As a response to this growth in the maritime sector, the US Department of Homeland Security developed the National Strategy for Maritime Security of 2004 that has the objectives to enhance international cooperation to prevent terrorist or criminal attacks in the maritime domain, to protect maritime-related population centers and critical infrastructure, to minimize damage and expedite recovery following an incident in the maritime domain and to safeguard the ocean and marine resources. Maritime Domain Awareness is an important component of the National Strategy for Maritime Security that is built on efforts currently carried out by the US Coast Guard and Customs and Border Protection, National Oceanographic and Atmospheric Administration and other Federal agencies to gather information on vessel operation and ownership, the maritime cargo supply chain and the marine environment.

Unfortunately, the more critical an infrastructure is for a nation, the more susceptible it is to terrorism efforts. The geography of a port makes it vulnerable to attacks that may be initiated from air, ground and water. In the literature of critical infrastructure protection against intentional attacks, numerous articles have studied and modeled situations where the attack is launched and performed on the ground (Yates et al. 2011; Dimitrov et al. 2009; Lee and Kulesz 2008). However the literature of seaborne terrorist attacks is not large. An Autonomous Underwater Vehicle (AUV), a diver or a peculiar object mounted at the bottom of a vessel can be a potential source of a terrorist attack to the infrastructures located around a port.

In general, two approaches dominate the literature of resource allocation problems against terrorist attacks; namely Probabilistic Risk Model (PRM) and Strategic Risk Model (SRM). Although it may not seem logical in the beginning to use the PRM for security risk analysis, it performs reasonably well in allocating resources. The idea of the PRM originates from the safety risk problems where the goal is to mitigate the consequences of unintentional events such as natural disasters, machinery failures, accidents and etc. Based on the literature, it is suitable to assume that these events occur by chance and there are factors that increase the chance of their occurrence. That is, evaluating the probability of occurrence of such events as a function of factors causing them; e.g. poor visibility due to bad weather conditions can lead to higher accident probabilities. The same approach is used in many literature articles to model the security risk analysis. It means to assume that the terrorist attacks happen in the same manner as unintentional events. This approach assumes that any factor that attracts the terrorists' attention increases the probability of a potential attack. Then it attempts to allocate resources in the sections of the field of interest where the potential attack probabilities are high in order to prevent them. The shortcoming of this methodology is the fact that it ignores any type of response that the intelligent attacker may do to neutralize the defender's actions. In other words PRM misses the interaction between the attacker and defender based on their decisions.

On the other hand, the SRM considers both parties as intelligent players of a game, where each player wants to maximize his gain from the situation based on his conception about the other player's decisions. The SRM seeks to find an equilibrium in which neither of the players can do better by just changing his decision. The structure and assumptions of the SRM fit better to the resource allocation problem against security risks. As a result, most of recent studies utilize the SRM.

1.2. Contribution

This study contributes to the literature of maritime security risk management via proposing methodologies for allocating sensors under the water in a port to detect underwater anomalies. The major contribution is in the game theoretic (strategic) model. Numerous articles in the literature have modeled the security risk problem by game theoretic methodologies (Zhuang and Bier 2007; Bier, Samuelson, and Oliveros 2007; Powell 2007; Paruchuri et al. 2008; Kunreuther and Heal 2002). However, the problem of interest for most of these studies is to allocate a limited resource among candidate sites located far apart. This implicitly suggests that assignment of resources to one site does not improve (or deteriorate) the coverage on any other site due to the distance between sites. This approach is not capable of capturing the essence of sensor placement in ports.

In order to apply a sensor allocation methodology to a body of water (a port), it needs to be discretized into sites. A practical discretization technique is to put a grid of square cells on the environment (Dhillon and Chakrabarty 2002; Kim and Park 2006; Yates et al. 2011). In such a case typically the range of detection for sensors is larger than cell size. As a result, allocating resources to cells (placing a sensor in a cell) certainly affects the coverage of adjacent cells. To the best of my knowledge, this effect (called "interdependent allocation" in this manuscript) is not

considered in any of the game theoretic models for security risk analysis. The game theoretic model proposed in this study features interdependent sensor allocation.

Two approaches are used to model the sonar allocation problem in this study. The first one is called the probabilistic modeling approach. It is a mixed integer linear optimization program and is inherently probabilistic due to the uncertainty associated with sonar coverage (a sonar covers objects that fall in its coverage range with a probability which is called detection probability). The second one is the strategic modeling approach. It is a game theoretic model that converts the sonar allocation problem to a game between two players (the defender and the attacker).

In the probabilistic approach, the proposed model is quite complex. It is not feasible to solve this model for optimality using available Solver Packages. Another contribution in this study is devising a heuristic solution methodology for the probabilistic model. It can be used to obtain close-to-optimal solutions in a reasonable amount of time for large scale problems.

As mentioned earlier, similar approaches have been developed for security risk analysis problem in the literature. Yet this study focuses on some significant aspects of this problem which are often ignored or left off through simplifying assumptions. Instances of these contributions are:

- Details of sensor specifications and various coverage types (not necessarily 360˚ coverage) are considered. Not all sensors have 360˚ coverage. To reduce costs, it is possible to use sensors with lower degree coverage for special purposes such as acute angle coverage. Various sensors with different angular coverage can be used in the proposed models. Another sensor attribute that affects the modeling effort is the sensor detection range. It helps to specify the cell size as mentioned earlier.

- Geographical features of the environment such as curvatures and barriers are taken into account. If a barrier blocks the line of sight for a sensor, then the cells behind the barrier are not

going to be covered by the sensor. Moreover, the more curvatures and details exist in the environment, the smaller the cell size is needed to be to capture the details.

- Higher detection probabilities are assumed for the cases where multiple sensors cover a cell. This effect is referred to as “multiple detection” in this manuscript. According to interdependent allocation concept described before, a sensor allocated in a cell can cover an adjacent cell too. There might be circumstances where multiple sensors are placed in the vicinity of a cell. If this cell falls within the detection range of more than one of those sensors, the cell will be under multiple detection. In this case, the cell’s detection probability is higher than the detection probability it receives from any single one of those sensors.

- Detection probabilities of sensors decrease as a function of the distance from the sensor. This fact is called “range-dependent detection probability” and is regarded in this study. As sound waves travel in a medium, they lose their energy. This effect ultimately leads to the reduction of detection probability as a function of distance.

- Characteristic values of cells play a significant role in both modeling approaches. This parameter shows how significant a cell is and how critical it is to keep the cell under surveillance (surveillance and coverage are used in this manuscript interchangeably). The characteristic values affect the sonar allocation scheme directly. That is, both models decide the location of sonars based on this measure. The process of calculating cell characteristic values is also described in this study.

The objective of the study is to develop an allocation scheme for sensors, such that the probability of detecting potential attacks is maximized. A budget constraint limits the number of sensors to be allocated. Hence it becomes critical where to place these resources to achieve the maximum surveillance. In order to quantify the characteristics of the field of interest, a discretization process is required.

- The discretization process and its effect on model's parameters and its performance are discussed. Putting a grid of cells on a field of interest may seem an easy preprocessing task. However, the size of cells depends on the geographical features of the field and the desired level of accuracy in the model. On the other hand, decreasing the cell size leads to greater number of cells. As the number of cells increase, the size of the optimization model expands and reaching its optimal solution becomes more and more difficult. Hence a tradeoff between the level of details in the model and the effort to reach optimality can specify a feasible range for cell size. It is possible to tighten the feasible range for the cell size further by considering the sensor specifications. This process is explained in more details in Chapter 2.

1.3. Structure of Chapters

The remainder of this study is structured in the following manner.

In Chapter 2, the literature of sensor allocation problem for security is discussed first. Then the proposed PRM is presented in two steps. First a basic model is presented to illustrate the idea behind the model. This model is followed by an example. Then the full version of the model is built on top of the basic model. It is a mixed integer linear programming model that seeks to minimize the overall risk of the port by placing a limited number of sonars under the water in a subset of candidate points set. The full model features range-dependent detection probability and multiple detection of cells. It also allows multiple sonar types to be used. A few preprocessing steps are required to prepare the parameters for the sensor allocation model. These steps are discussed in this Chapter as well.

The proposed model is an NP hard problem and it cannot be solved for optimality efficiently in case of large-scale problems. In Chapter 3, an example is provided to show how long it takes to solve even a small case of the full model. Then a greedy heuristic solution technique that utilizes the greedy solution for knapsack problem is provided to solve the model for large-scale instances

in a reasonable amount of time. The computational complexity of this methodology is then discussed and a number of test cases are followed to exercise the performance of the heuristic method. The heuristic solution technique performs very well when the density of sonars is low in the field of interest. This fact allows the solution method to be used to in real world scenarios. Due to budget limitations in such scenarios, the sonars are quite scarce in the field and the solutions from the heuristic technique are close to optimal.

Next, the SRM is proposed in Chapter 4. A number of definitions in the area of game theory are presented at the beginning of this Chapter. Then the literature of application of game theory against terrorist attacks is surveyed. A few concepts about game theory that are going to be used in the model are introduced before presenting the modeling methodology. A basic though powerful concept is the duality theory that relates a zero-sum game (where the payoff of two players add up to zero for all possible situations) to a duplet of dual linear program models. It helps to find the mixed strategy equilibrium of the game by solving the equivalent linear programs. The solution for the primal linear model provides the attacker's optimal mixed strategy and the solution to the dual model is the defender's optimal mixed strategy. Our interest in SRM is to obtain the defender's optimum strategy that leads to sonar allocation. The attacker's problem solution is not of much interest here. However, it can be used later to generate attack scenarios to compare the performance of various models.

The payoff matrix of this particular SRM (which is the input for the dual linear programs) is then exploited. This matrix is populated based on various combinations of attacker and defender strategies. Then a few simplifying assumptions are made. These assumptions let us propose the game theoretic model in a more straightforward way. The final step in this Chapter is to provide the linear program of the defender which can be a guideline for sonar allocation.

Chapter 5 is dedicated to the removal of simplifying assumptions that are made through proposing the game theoretic model. These assumptions are removed step by step and each makes

the mathematical linear program of the defender or the payoff matrix more complex. Once these assumptions are removed, the model becomes a binary linear program and the size of payoff matrix becomes dependent on the number of sonar types that are allowed in the model. This model is comparable with probabilistic model of Chapter 2 in the sense that they incorporate the same level of features and details. These two models will be compared by defining three performance measures at the end of this chapter.

The final step in extending the SRM of Chapters 4 and 5 is to allow the payoff matrices to be general-sum (rather than zero-sum). This leads to flexibility of SRM in accepting any payoff matrices for players. A mixed integer linear program approach is selected from the literature to build the general-sum game. This model is able to integrate the features of the sonar placement problem into it. We extend this model step by step to integrate these features (as we did in Chapter5). Finally, the performance of the final general-sum SRM is tested via a test case. This test case illustrates the significance of attacker's intelligence and interest in the final solution of the game.

1.4. Notes

While reading the rest of this study, some questions may rise in the mind of reader; such as the reason for using sonars or the rationale behind assuming a two dimensional environment while it is three dimensional. Here, a few discussion points are presented that help to resolve these questions to a great extent.

- Electromagnetic sensors that are widely used for detection in many applications are not feasible to be used under the water in ports and waterways. Due to chemical properties of the field of interest (salty water) in our study, electromagnetic waves lose their energy as travelling through salty water so fast that their detection range does not exceed 50 ft. Instead of electromagnetic sensors, a different type of sensor called sonar (sound navigation and ranging) is

assumed to be deployed for detection purposes. As suggested by its name, sonar operates based on sound waves as opposed to electromagnetic waves. Sound waves can propagate in salty water without losing much energy. Sonar detection range under the water may even exceed 10000 ft.

- The region of interest in this study is a body of water which is clearly a three dimensional (3D) space and the objects can move in any direction under the water. Yet the grid to put on the field of interest is assumed to be two dimensional (2D) in our modeling effort. In other words, the candidate points to place sonars lie on a plane and the coverage region of sonars is an area (as opposed to a volume). Comparison of the depth of water in the field of interest in this study and the range of sonar coverage provides the rationale behind this assumption. Water depth (d) at ports and waterways fall below 100 ft. (National Dredging Needs Study of U.S. Ports and Harbors 2002), while the range of coverage (r) is at least 2000 ft. for a typical sonar. The sonars are to be deployed at the seabed on a foundation and they (a sonar with 360 ˚ coverage) can cover the half-sphere with radius r above them. Since the water is shallow, the coverage volume of the sonar is cut by the water surface at the distance r-d above the location of sonar. The volume under the coverage of a sonar is approximately a disc (cylinder) with radius r and height d. Instead of dealing with this volume, it is possible to map it to a circle with radius r in a 2D space and assume that the sonar covers the circle. Clearly everything falling in the volume of the disc will also appear in the projected circle. Moreover the seabed is usually a flat surface in ports and waterways and any steep surface that may obstruct the line of sight for sonars can be treated as a barrier.

The above explanation is just used as a justification for mapping a 3D field to a 2D equivalent of it for the modeling effort. However, in reality the sonar can recognize in what 3D location the sonar is and also in what direction it is moving. The seabed is usually a flat surface in ports and waterways and any steep surface that may obstruct the line of sight for sonars can be treated as a barrier and can be modeled with both modeling methodologies.

- The network of sonars to be allocated are assumed to be linked to each other and to the hub in land with wireless connections. The network topology of these nodes and their connection issues and limitations are not the subject of focus in our study. Due to the budget limitations and relative area of a region of interest and the sonar area coverage, the number of sonars to be placed in a port or a waterway will not exceed the order of ten. Hence, we assume that the connectivity of sonar nodes is not a major issue practically.

- In our study no effort is designated to distinguish the type of anomalies a sonar has detected. Recognition of anomalies from the signals that sonars receive is a hot topic itself in signal processing. We assume that if a sonar detects an object, it sends the corresponding signals to the hub. When the signals reach the hub, they are processed properly to extract the information about what object has fall within the coverage, where it is located and in what speed and direction it is moving.

- The problem of interest in the PRM may resemble a number of other optimization problems in the literature; e.g. facility location problem (FLP), wireless sensor networks optimization (WSNO) problem. Although these problems may look the same in general, in details they differ to a great extent.

Based on its nature, each of these problems imposes a number of limitations to the optimization model that makes the problem unique; e.g. in WSNO the nodes are required to be connected to each other. Also due to large number of sensors to be placed, data transfer capacity issues may arise in this type of problem. These lead to a different set of constraints compared the ones in our problem. Also the physics of our problem makes it different from an FLP. That is, the coverage of sonars is over an area of interest, while in the FLP, the resources are required to service points of interest.

According to the application the model is developed for, every model may seeks different goal; e.g. in facility location problems the objective is to minimize the cost of material transportation,

while in our problem this cost does not apply and the goal is to maximize the coverage of sonars. In WSNO the objective may be to maximize the coverage or connectivity, to minimize the cost or bandwidth problems or any combination of them.

Moreover, the structure of problem is fixed in all of the problems mentioned above; e.g. the nodes to be served and their requirements are known in an FLP, or the number of users in a WSNO problem is at least probabilistically known. However, in an SRM the attacker will choose the node to attack based on his intelligence. As a result the defender cannot act according to a pre-specified structure. Instead, he needs to play a game with the attacker. This idea makes the SRM approach totally different from the aforementioned models.

2. Probabilistic Risk Model

2.1. Introduction

Critical Infrastructure Protection has become one of the highest priorities of department of Homeland Security as a result of the terrorist attacks of September 11th, 2001. These attacks showed how dramatic the aftermath of such an attack can be. According to Bram, Orr, and Rapaport (2002), it led to a loss of almost 3000 lives. It also affected the US and world economy tremendously based on Makinen (2002), caused health issues for several months in one of the most highly populated cities of the world as suggested by Barry (2006) and many more consequences that no one thought of before. All nations learned important lessons from these incidents and have planned to become more aware and ready against similar efforts.

According to the Department of Homeland Security (2011), there are 18 categories of critical infrastructures in the US and at least hundreds in each category. Clearly, it is not feasible to protect all of them against undesirable attacks with a limited budget. Among all, there are instances that a number of infrastructures are located closely and operating interdependently such as ports. Ports typically consist of cargo vessel terminals delivering the nation's essentials, oil terminals, refineries, chemical plants and transportation systems. Hence, a typical port includes tens of infrastructures operating in the same region and in case one of these infrastructures is compromised it will affect the others; e.g. the whole port may shut down due to security issues for a period or the efficiency of the port may decrease dramatically for a much longer time.

This Chapter contributes to the literature of critical infrastructure protection and maritime security risk management via proposing a sonar allocating methodology under the water in ports and waterways. The objective is to keep the underwater environment under surveillance of sonars to minimize the risk of potential attacks by underwater anomalies such as divers, torpedoes or etc.

Our approach integrates technical features of sonars into the mathematical model. Multiple coverage of sonars (meaning that the detection probability in a cell increases as the number of

sensors covering that section increases) and range-dependant detection probability (implying that the detection probability decreases as the distance from a sensor decrease) are two major specifics of sonar that are incorporated in this study. The preprocessing steps that affect the model parameters are also discussed here. These steps which are mainly referred to as the discretization process are usually missing in the literature articles.

The next section provides a background on security risk analysis and sensor placement and discusses the deficiencies in the literature where the content of this work can help. Then, the mathematical model is presented in two steps. First a small model is provided to illustrate the concept of the optimization model. An example shows how this model works. Then the full-scale model is introduced and modified through a few steps. The preprocessing steps to prepare model parameters are then discussed and followed by the conclusion of this chapter.

2.2. Literature Review

The number of articles studying security risk analysis of logistics operations and transportation systems is ample. Although the modeling approach may differ in various articles, they all seek the same objective, controlling and minimizing the probability and/or the impact of undesirable events. According to Kaplan and Garrick (1981), in order to address security risk analysis in an environment, one need to answer these three questions; what may happen, how likely is that, and what are the consequences. However, many of the citations in the literature use a more quantitative and formal approach to evaluate and mitigate risk. This approach takes advantage of the concept of risk as the expected consequence of incidents. In other words, the expected value of the consequences of incidents which are likely to happen in a field can be thought of as the risk of that field. Hence to calculate the expected consequences, it can be conditioned on the types of possible incidents. That is

$$E(C) = \sum_j E(C|I_j)p(I_j) \tag{2.1}$$

where $E(C|I_m)$ is the expected consequence given the incident type m and $p(I_m)$ is the probability that incident type m happens. Since any attack does not necessarily lead to an incident, we need to further condition the incidents on the attack types in order to quantify the probability of incidents. Note that the consequences are not directly dependent on the attack types, but the dependence is through the incident. That makes the following equation (Willis 2007) legitimate.

$$E(C) = \sum_j E(C|I_j)p(I_j) = \sum_j \left\{ E(C|I_j) \left[\sum_k p(I_j|E_k)p(E_k) \right] \right\} \tag{2.2}$$

$P(I_j|E_k)$ is the conditional probability of incident occurrence (successful attack) given that an attack has happened and $p(E_k)$ is the probability of attack occurrence. In security risk terminology, $p(E_k)$ is called the threat (T) probability, $P(I_j|E_k)$ is the vulnerability (V) of the asset against the attack, and $E(C|I_j)$ is the expected consequence given the attack. Thus, this analogy leads to the well-known risk formula proposed by multiple studies in the literature (Willis 2007; McGill, Ayyub, and Kaminskiy 2007; Ezell et al. 2010):

$$R = C \cdot V \cdot T \tag{2.3}$$

That is the basis for a number of articles in risk analysis. Unlike Golany et al. (2009) and Garrick et al. (2004) whom incorporate all three factors in their models, most literature articles focus on one or two of the factors and attempt to mitigate the risk by alleviating those factors and provide a decision support system to combat terrorism; e.g. Brown et al. (2005) and Brown et al. (2006) attempt to minimize the consequence cost of invading an infrastructure by letting the attacker and defender play a game, where their interaction defines the threat probabilities. Yates et al. (2011)

try to minimize the vulnerability of an infrastructure by decreasing the attack probability via supervising the possible assault routes.

As a result of the US Department of Homeland Security's concentration on defending critical infrastructures against terrorist attacks after the 9/11 incident, numerous post 2001 articles study different aspects of security risks of infrastructures (Simonoff, Restrepo, and Zimmerman 2007; Leung, Lambert, and Mosenthal 2004; Berry et al. 2005; Golen, Mishra, and Shenoy 2010; Yates et al. 2011). Despite studying the same subject, these articles use different approaches to model risk and suggest a wide range of solutions to mitigate it. This variety may be due to two factors. First, there exists a wide range of infrastructures each having different specifics; e.g. Simonoff, Restrepo, and Zimmerman (2007) basically analyze the historical data of disruptions such as terrorist attacks to electric power grids in the US and Canada over time and their consequences. Leung, Lambert, and Mosenthal (2004) prioritize bridges for protecting them against terrorist attacks and then perform a threat and vulnerability analysis for each specific case. Berry et al. (2005) provide a resource allocation scheme for placing sensors in municipal water networks against maliciously injected contaminants such that the expected fraction of the population at risk is minimized. Besides, the variety of attack modes i.e. air, ground and water adds another dimension to the problem. Golen, Mishra, and Shenoy (2010) provide an underwater sensor allocation scheme for an area clearance scenario, and Yates et al. (2011) apply a similar idea for protecting critical infrastructures on the ground by protecting the routes leading to them.

A generic approach in protecting critical infrastructures is to increase their surveillance by installing detective instruments in their proximity such that the probability of undetected attack and its success decreases. Sensors are the most widely known detective objects used in this field. This risk mitigation approach leads to a well-known problem in optimization theory which is the Sensor Placement Problem (SPP). Due to the diversity of sensors, they can be used in many applications, e.g. Dimitrov et al. (2009) propose a mixed integer linear optimization method to

deploy radiation detectors against nuclear material smugglers in a transportation network, Lee and Kulesz (2008) also propose a sensor placement methodology to protect the population against exposure to chemical, biological and radiological threats. Chakrabarty et al. (2002) and Dhillon and Chakrabarty (2002) provide models for grid coverage sensor placement to keep a region under surveillance and use heuristic techniques to solve their models. Clouqueur et al. (2002) deploy sensors on the ground using a sequential heuristic technique such that the exposure of the paths (leading to the infrastructures) to sensors is maximized and hence the surveillance is maximized. Wilhelm and Gokce (2010) provide a model to design a surveillance system for ports and waterways by utilizing various types of sensors to detect anomalies above water. They propose a mixed integer linear model and use the branch and price decomposition technique to solve it. Golen, Mishra, and Shenoy (2010) provide an underwater sensor allocation scheme (based on game theory) to protect a specific part of the ocean against submarine threats.

In the literature on sensor placement problem, the models are divided into three categories based on what they are seeking to cover. Most are focused on covering a set of points of interest (infrastructures) and are called point coverage problems such as Chakrabarty et al. (2002) and Dhillon and Chakrabarty (2002). This category is quite analogous to the Facility Location Problem (FLP) and the Art Gallery Problem (AGP). The approaches and algorithms developed for FLP and AGP can be used to solve the point coverage problem with minimal modifications. Some such as Yates et al. (2011) are interested in covering the routes and paths that lead to infrastructures. This group is called barrier or path coverage problems. The last group seeks to cover an area of interest and is known as area coverage problems. Since the physics of the area coverage problem is different from the two other classes (characteristics of cells are distinguishable from a point or line assumption), it is not usually possible to use solution techniques for other categories to solve the area coverage problem.

Golen, Mishra, and Shenoy (2010) study the area coverage problem. This article is the most relevant research to the problem of our study. Although Pompili, Melodia, and Akyildiz (2006) and Aitsaadi et al. (2007) investigate underwater sensor networks allocation, they seek dissimilar objectives and have different constraints. They study the full area coverage problem in which there are ample sensors for deployment. In this type of problem the cost is considered as an objective function sought to be minimized (this is not a realistic approach in real case scenarios, where budget is a pre-determined value). Moreover, the model proposed in Pompili, Melodia, and Akyildiz (2006) is suitable for the uniform sensing problem which is not consistent with sensor placement against terrorist attacks. Golen, Mishra, and Shenoy (2010) look for the optimal placement of sonars under water such that maximum surveillance is achieved with a limited budget. However, due to the vastness of the environment under study (sections of ocean or open sea), the authors assert some assumptions that make the model less deficient for ports and waterway surveillance. The authors discretize the environment into cells within which the acoustic characteristics are quite homogeneous. Due to the slow rate of change in water characteristics in the oceans, the cell sizes are relatively large such that multiple sonars are allowed to be placed in a single cell. As another result of considering large cells, the authors assume that placing a sonar in one cell may have no effect on the adjacent cells. Since the transition in environmental characteristics that affect the sonar coverage is much faster in coastal waters comparing to oceans (Knauss 2005), the cell size needs to be smaller to keep the homogeneity in a cell. As cells become smaller it is not safe anymore to assume no relation between a sonar in one cell and the coverage in a nearby cell.

As an area coverage problem, our study assumes that a sonar that is placed in one cell can cover (the area of) as many cells as its range of detection allows. The model allows a cell to be covered by multiple sonars, in which case the detection probability increases. Additionally the change (decrease) in detection probability of a sonar by moving farther from the sonar is considered in

our model. This relationship is not linear and in most cases there is no closed-form expression to explain it. It is usually measured through experimentations in the water body of interest and analyzed using graphs as given in Urick (1983). This is one of the significant benefits of our modeling effort in this study as will be explained in the following sections.

2.3. A Grid View (discretization)

In order to effectively allocate sonars in a given body of water, we propose to identify the area with a grid consisting of a number of cells with the same size each having a set of key attributes. That is we discretize the area into a number of units of smaller areas potentially to be covered by a sonar in the vicinity. This approach is a typical way to study sensor placement problems in large scale regions of interest and numerous literature articles have utilized this procedure (Dhillon and Chakrabarty 2002; Golen, Mishra, and Shenoy 2010; Lin and Chiu 2005). The cell size of the grid depends on the desired level of accuracy, geographical shape of the body of water and the level of activity in the port. Generally, the more irregular the boundary lines of the waterway are and the more congested it is, the smaller the cell size should be to make the placement process more accurate. Resolution of sonars and complexity of the mathematical model are also important factors to keep in mind while deciding on the cell size. A more detailed discussion about the cell size and how to discretize the detection probability is presented later in this Chapter.

2.4. The Mathematical Model

In this section we present two models that seek optimal placement of sonars such that the port and its infrastructures are under surveillance of sonar coverage to provide maximum risk reduction against any undesirable intrusion or attack under water. A budget constraint will be the limiting factor.

2.4.1. The Basic Model

A basic mathematical model that can be used in sonar allocation involves a linear risk minimization problem with binary decision variables. The model parameters are:

a_{ij}: characteristic value of cell (i,j). This parameter shows how significant a cell is and how critical it is to keep the cell under surveillance. In fact the value of this parameter governs the problem and specifies a priority for cells to be covered by sonars.

dp: sonar detection probability.

s: price of a sonar.

b: budget to place and maintain sonars.

N_{ij}: set of neighboring cells of cell (i,j) including itself that a sonar can cover.

Note that in this study, we have chosen the lower left corner of each cell to represent the cell and be the candidate point for sonar placement. Figure 2.1 shows a sonar being placed in cell (i,j) (the hatched one) that covers the 4 cells including cell (i,j) itself. In this figure $|N_{ij}|$=4, where | | refers to the size of the set.

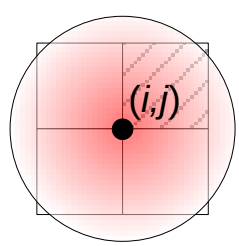


Figure 2.1. A sonar being placed in cell (i,j)

The decision variables in the basic model are:

$$x_{ij} = \begin{cases} 1 & if\ a\ sonar\ is\ placed\ at\ cell\ (i,j) \\ 0 & otherwise \end{cases}$$

and

$$y_{ij} = \begin{cases} 1 & if\ cell\ (i,j)\ is\ covered\ by\ a\ sonar \\ 0 & otherwise \end{cases}$$

Note that all the sonars to be placed in this model are of the same type. Using this notation, the model can be written as follows:

$$\min \sum_i \sum_j a_{ij}(1 - dp.y_{ij}) \tag{2.4}$$

s.t.:

$$y_{ij} \leq \sum_{(g,h)\in N_{ij}} x_{gh} \qquad \forall(i,j) \tag{2.5}$$

$$\sum_i \sum_j s.x_{ij} \leq b \tag{2.6}$$

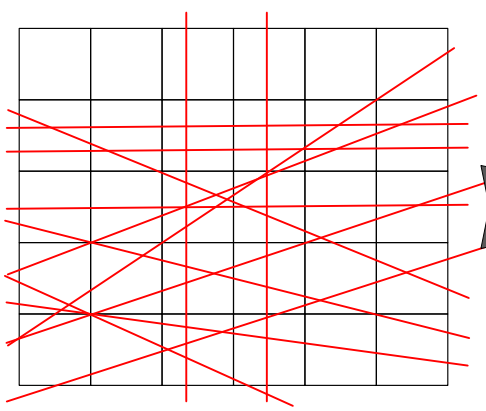
$$x_{ij}, y_{ij} \in \{0,1\} \tag{2.7}$$

The objective function (equation (2.4)) is a risk measure that the model is minimizing. The a_{ij} value shows the criticality of the cell (i,j) and the bigger it is, the higher the significance of the cell is. If a cell is covered by a sonar, then y_{ij} for that cell is one and therefore (for dp values close to one such as 0.95) the coefficient of a_{ij} is small which can be interpreted as if a cell is covered by a sonar then the cell is better prepared against undesirable intrusions and hence exposed to less risk. Conversely, if the cell is not covered, y_{ij} is zero and the a_{ij} value is multiplied by one and will remain the same which means the cell is not under surveillance and remains vulnerable to attacks. Inequality (2.5) assures that for a cell to be covered there must be a sonar in the vicinity which is covering the cell. Since the objective function seeks to minimize the weighted summation of $-y_{ij}$'s (maximize the weighted summation of y_{ij}'s) it prefers to enforce the maximum number of y_{ij}'s to be equal to one. Consequently inequality (2.5) works as equality and

whenever there is a sonar in the vicinity of a cell that can cover the cell, the cell's corresponding y_{ij} becomes one. This intuition also helps us to see that the binary constraint on the variable y_{ij} is not necessary. In other words, equations (2.4) and (2.5) enforce the y_{ij}'s to be binary. Inequality (2.6) is just the budget limitation to purchase, deploy and maintain the sonars and the final statement (equation (2.7)) forces the decision variables to be binary.

2.4.2. An Example

A simple example is provided here to better understand the workings of the model. The problem includes a field of 30 cells with some hypothetical ship lanes passing through them as shown in Figure 2.2. For simplicity, the a_{ij}'s are obtained merely based on the number of paths passing through each cell; e.g. the bottom right cell of the grid is crossed by two lines (vessel paths) and therefore the characteristic value for this cell is two. Sonars are assumed to have 360˚ coverage and each covers 4 cells (refer to Figure 2.1).



0	0	1	1	0	1
3	3	3	5	4	3
2	2	5	6	3	2
4	3	3	4	3	2
3	3	4	4	1	2

Figure 2.2. Characteristic values of each cell in the grid

The proposed model is a binary integer programming model and since it involves a small grid, the number of variables and constraints is small enough to use an optimization solver to obtain the optimal solution. As illustrated in Figure 2.3 the optimal sonar placement tends to cover the cells with the highest characteristic values. Each sonar is assumed to cost four unit of price and the

total budget is 13 units. Thus due to budget limitations, just two sonars are used to cover eight cells with highest characteristic values in the optimal allocation.

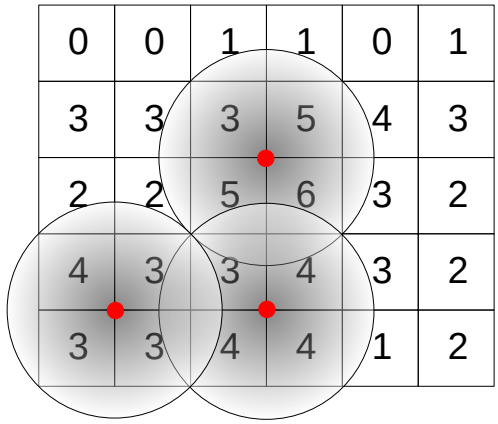


Figure 2.3. Optimal sensor placement for a simple example

Since one sonar type is allowed in this basic model, the number of cells (c) and budget (b) are the only variables that affect the complexity of the model. For fixed c and b values the number of possible ways to allocate sonars is easy to calculate. The number of sonars is $n = \left[\frac{b}{s}\right]$ (the integer part of $\frac{b}{s}$). Then the number of possible sonar allocations is the number of possible combinations of n out of c (c_n^c). Hence when the budget constraint is low (the case in real case problems) even if the number of cells is large (less than millions) the number of possible allocations is still possible to check in a reasonable amount of time. Although this model does not cover most of the features of the main model presented in next section, it still can be used for large-scale problems (if the assumptions do not spoil the quality of the problem) to provide optimal solutions, since solving the main model for optimality might be infeasible in such cases.

2.4.3. The Comprehensive Model

The optimization model of the last section is not detailed and general enough to be used in real world scenarios. It needs to be modified such that it can integrate more realistic assumptions.

Here, a more advanced model will be presented that covers the deficiencies of the basic model to a great extent. It allows multiple sonar types to be used in the model. It considers that the detection probability of a sonar decreases as the distance from the sonar increases. It also takes the extra coverage (when multiple sonars cover a cell) into account. From another point of view, this model works the same as the basic model fundamentally. They share a similar objective function and all the constraints of the basic model exist in the advanced model. Many of the parameters and decision variables are the same as well. Next, the comprehensive model is presented.

The model parameters are:

a_{ij}: characteristic value of cell (i,j).

n: index for type of coverage, i.e. n=1 indicates the highest detection probability and n=2 indicates the second highest detection probability, and so on.

dp_n: type of detection probability due to proximity to the sonar. This parameter is needed to represent the change in detection probability as a function of distance from the sonar. For example, n equals one and the detection probability is close to one for those cells within the close proximity of the sonar, and as the distance between the sonar and the cells increase, n increases and the detection probability decreases, as shown in Figure 2.4.

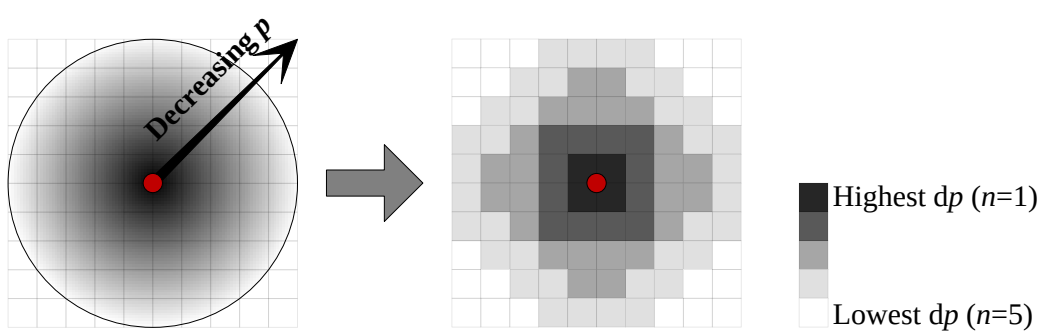


Figure 2.4. Types of detection probability

dp_{max}: the maximum detection probability considered. ($dp_{max}>dp_n$, $\forall n$)

m: index for sonar type (sonars differ in range, detection probability, angular coverage or other specifications).

s_m: price of a sonar type m.

b: available budget to purchase sonars.

N_{ijmn}: set of neighboring cells of (i,j) including cell (i,j) itself, that a sonar type of type m positioned in cell (i,j) can cover by detection probability of type n. For example in Figure 2.4 if the place that the sonar (the red circle) is placed at is cell (i,j), for the shown sonar type, if n=1 then the 4 darkest cells in the center of the discretized shape are the only elements of N_{ijm1} and for n=2 the 12 cells that are surrounding the cells at the center are the elements of N_{ijm2}.

u: maximum number of the same type multiple coverage to be allowed.

The decision variables are

$$x_{ijm} = \begin{cases} 1 & if\ a\ sonar\ of\ type\ m\ \ is\ placed\ in\ cell\ (i,j) \\ 0 & otherwise \end{cases}$$

$$y_{ijn} = \begin{cases} s & if\ cell\ (i,j)\ is\ covered\ by\ coverage\ type\ n\ of\ s\ sonars \\ 0 & otherwise \end{cases}$$

and

$$t_{ij} = \begin{cases} 1 & if\ cell\ (i,j)\ is\ covered\ by\ more\ than\ 1\ sonar \\ 0 & otherwise \end{cases}$$

The model can be constructed as:

$$Min \sum_i \sum_j a_{ij} \cdot \left\{ 1 - \left[\left(\left(1 - t_{ij}\right) \sum_n dp_n . y_{ijn} \right) + t_{ij} . dp_{max} \right] \right\} \tag{2.8}$$

s.t.:

$$y_{ijn} \leq \sum_{m} \sum_{(g,h) \in N_{ijmn}} x_{ghm} \quad \forall i, j, n \tag{2.9}$$

$$\sum_{n} y_{ijn} - 1 \leq M \cdot t_{ij} \quad \forall i, j \tag{2.10}$$

$$M(1 - t_{ij}) + \sum_{n} y_{ijn} \geq 2 \quad \forall i, j \tag{2.11}$$

$$\sum_{i} \sum_{j} \sum_{m} s_m \cdot x_{ijm} \leq b \tag{2.12}$$

$$x_{ijm}, t_{ij} \in \{0,1\} \tag{2.13}$$

$$y_{ijn} \in \{0,1,2, \dots, u\} \tag{2.14}$$

The objective function (equation (2.8)) is a risk measure to be minimized. In fact the rationale behind this objective function comes from equation (2.1). The a_{ij} value, as described earlier, shows the criticality of the cell (i,j) and it can be thought of as the consequence level in case that cell is attacked successfully. The term multiplied by a_{ij} in equation (2.8), as will be described below captures the vulnerability of the cell and is called the vulnerability function. Threat probability is assumed to be taken into account with the definition of a_{ij}. That is, the higher the value of a_{ij} is, the more prone it is to an attack by the adversaries. Since the attacker is likely to prioritize cells for attack based on the significance values of the cells, the threat probabilities will be rational to the significance values of the cells. Accordingly, the significance values of cells appear twice in the equation implicitly. Instead it is reasonable to assume that the significance level of the cells embodies the threat probabilities in it and remove the threat probability from the equation.

Now let us explain equation (2.8). If a cell is covered by at most one sonar, then the t_{ij} value for that cell is zero and $1\text{-}t_{ij}$ is one. If $t_{ij}=0$ and the cell is not covered, the preparedness is zero for that cell and the coefficient of a_{ij} is one. If $t_{ij}=0$ and the cell is covered, depending on the type of detection probability, the preparedness of that cell is calculated (using dp_i for type i) and *1-preparedness* results in unpreparedness, which is then multiplied by a_{ij}. If the cell is covered by more than one sonar, t_{ij} is one and hence $1\text{-}dp_{max}$ is multiplied by a_{ij}. Therefore the more a cell is covered by sonars the less it is vulnerable against attacks.

In order to explain the way multiple detection of sonars over one cell is modeled, consider the vulnerability function of a cell as

$$1 - \left[(1 - dp_i) \cdot \left(1 - dp_j\right) \ldots (1 - dp_s)\right] \tag{2.15}$$

where $i,j,\ldots,s$ refer to the type of coverage the cell receives from all the covering sonars according to its distance with sonars. When a cell is not covered by any sonar its unpreparedness (vulnerability) is one. If it is covered by one sonar the vulnerability will be $1\text{-}dp_i$, where i corresponds to the type of coverage the cell receives. In case the cell is covered by more than one sonar, the vulnerability comes out to be as given in equation (2.15). Since the values for detection probability are quite large (values close to one such as 0.95), the term $[(1\text{-}dp_i)\cdot(1\text{-}dp_j)\ldots(1\text{-}dp_s)]$ tends to get smaller and smaller as the number of covering sonars increase; e.g. if the smallest detection probability of sonars is 0.9, the vulnerability of a cell with two covering sonars is 0.01, and by increasing the number of sonars this value merges to zero. Hence, beyond double coverage the gain in vulnerability by increasing the sonars is minimal. Therefore, to make the model less complicated a maximum value of detection probability (dp_{max}) is introduced and any cell with multiple coverage is assumed to be covered by dp_{max} and equation (2.15) is replaced by the vulnerability function in equation (2.8).

Inequality (2.9) implies that in order for a cell (i,j) to be covered by detection type n ($y_{ijn}\geq 0$), there should be a sonar in the vicinity of that cell covering the cell with detection probability of type n. Since the objective function seeks to maximize the value of y_{ijn} implicitly, this inequality is always active (binding) at the optimal solution meaning that both sides of the inequality become equal. Equations (2.10) and (2.11) are meaningful when they work together. In fact, they make t_{ij} to be either zero or one. When the number of sonars that cover cell (i,j) is more than one, inequality (2.10) becomes tight and forces t_{ij} to be one, while constraint (2.11) is redundant. If at most one sonar covers the cell, inequality (2.11) becomes tight and forces t_{ij} to be zero and expression (2.10) becomes redundant. That is these two constraints are disjunctive. Inequality (2.12) is the budget constraint. Finally based on expressions (2.13) and (2.14) all the decision variables are binary except y_{ijn} which can be any positive integer.

In reality, with a limited budget, it is preferable to cover as many cells as possible rather than covering a number of cells with multiple sonars and keeping the rest uncovered. Furthermore, multiple coverage of the same type is even more unlikely to happen. As a result, it is reasonable to limit y_{ijn} to a small integer such as two or three. Besides, due to the functionality of equations (2.10) and (2.11), y_{ijn} is enforced to be integer. Consequently it is possible to remove the integrality constraint from y_{ijn} and just put an upper bound such as 2 or 3 on it (u=2 or 3).

2.4.4. Linearization of the Objective Function

Typically solving an optimization model with a linear objective function is easier than the ones with polynomial objective functions. That is why it is preferred to build a model with the least possible degree in the objective function. Luckily by introducing a set of auxiliary variables and constraints, this model can be converted to a new identical model with a linear objective function. The new binary variable w_{ijn} takes care of the multiplication of $(1-t_{ij})$ and y_{ijn} such that the conditions in Table 2.1 are satisfied.

Table 2.1. Required behavior of the new variable w_{ijn}

$1\text{-}t_{ij}$	y_{ijn}	Desired value of w_{ijn} (with respect to t_{ij} and y_{ijn} values)
1	1	1
1	0	0
0	1	0
0	2 or more	0

Due to equations (2.10) and (2.11), the other combinations of $(1\text{-}t_{ij})$ and y_{ijn} do not occur. In order to enforce the conditions in Table 2.1 a pair of new constraints is added to the model as shown below.

$$\frac{y_{ijn}}{u} + \left(1 - t_{ij}\right) - 1 \leq 2w_{ijn} \qquad \forall i, j, n \tag{2.16}$$

and

$$\frac{y_{ijn}}{u+1} + \left(1 - t_{ij}\right) + \frac{u}{u+1} \geq 2w_{ijn} \quad \forall i, j, n \tag{2.17}$$

Note that u in equations (2.16) and (2.17) is an input parameter in the model, as described earlier. After integrating W_{ijn} into the model, the multiplication of $(1\text{-}t_{ij})$ and y_{ijn} in the objective function is replaced by w_{ijn}. Hence, the objective function can be rewritten as

$$Min \sum_{i} \sum_{j} a_{ij} \left\{ 1 - \left(\sum_{n} \left(dp_n . w_{ijn} \right) + dp_{max} . t_{ij} \right) \right\} \tag{2.18}$$

with these modifications the objective function is linearized and the problem can be solved using a mixed integer linear programming model.

2.5. Preparing Model Parameters

Most of model parameters and variables depend directly on the cell size and number of cells to place on the field. A brief discussion on the discretization process follows.

2.5.1. Cell Size Calculation

The initial step of solving the sensor placement problem is to calculate the appropriate cell size of the grid that is going to be placed on the field, as it determines the number of decision variables to be used in the model. In most of the earlier work the cell size is assumed known, which is not the case most of the time. It is important to have an effective cell size that is not too small or not too large. The process of cell size determination is discussed briefly here.

As mentioned in the discretization process, several variables affect the cell size, most of them favoring smaller cell size. In order to obtain a reasonable cell size, it is necessary to study the field and test various cell sizes to find an appropriate one which includes the desired details of the field, while keeping in mind that too small a cell size will complicate the model and may not help catching increased details due to sonar resolution accuracy. The process can start with an approximate cell size and additional steps are taken to arrive at the optimal size considering different types of sonars. The basic idea is to maximize the overlap between the area covered by the sonar and the cells of the grid within the sonar's coverage, such that less sonar coverage is wasted and less accuracy is sacrificed. In Figure 2.5, the hatched area shows the wasted coverage and the dotted area illustrates the points that the accuracy is sacrificed. Although the dotted area falls out of coverage (with less detection probability), it is assumed that the whole cell including the dotted area is covered by the sonar with the same detection probability.

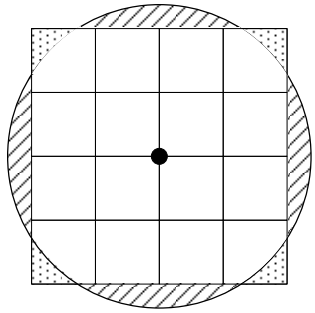


Figure 2.5. Idea of cell size calculation (in this figure all 16 cells are assumed to be covered)

Having an interval for cell size in mind and knowing the range of sonars considered, one can iterate the following search algorithm to find an optimal value for cell size where coverage wastes and loss in accuracy are minimized.

The algorithm to obtain the optimal cell size is given below:

L: lower bound for the cell size

U: upper bound for the cell size

q: sonar types

w_{ki}: wasted coverage for sonar type k when the cell size is i

l_{ki}: lost coverage for sonar type k when the cell size is i

f_i: total in efficiency for cell size i

<u>Algorithm</u>:

<u>Step 1</u>:

$\forall i \in \mathbb{N}, L \leq i \leq U$ //For any integer in the range of cell size

$\forall k \in \{1,2,\dots,q\}$ //For all sonar types

Calculate w_{ki} and l_{ki}

Calculate $f_i = \sum_{k=1}^{q}(w_{ki} + l_{ki})$

Step2:

Find $i^* = argmax_i\{f_i, L \leq i \leq U\}$

Use i^* as the cell size

Since the number of each type of sonar in the final solution is not known, this technique may not lead to the optimal cell size for the final solution of the problem, but at least it assures that an appropriate cell size is fed to the main model.

2.5.2. Discretizing Detection Probabilities

Fixing the cell size makes it possible to discretize the sonar coverage accordingly. Assuming binary detection probability for sensors (sonars) is a very rough approximation. Instead, it is more realistic to presume that detection probability reduces as the distance from the sonar increases. Exponential decay is the most common model for the change in detection probability as a function of distance. However, in the literature of underwater acoustics this assumption is not well-justified. Alternatively, it is suggested to find the corresponding relation through experimental measurements for the body of water under study as proposed by Urick (1983). Aitsaadi et al. (2007) suggest using multi-level detection probability to capture more details in the model. Our study considers the experimental relation between detection probability and distance from the sonar, and discretizes the detection probability as follows.

To discretize the detection probability, suppose that a sonar is placed at a candidate cell as shown in Figure 2.6 and we are interested to find the approximate detection probability for the grayed cells. Note that the detection probability is assumed to be the same for the grayed cells. The closest distance from the grayed cells and the sonar location is d_1 and the farthest distance is d_2. A straightforward way to approximate the detection probability for these cells is to average the corresponding detection probabilities for d_1 and d_2. It is also possible to use more complicated

and accurate ways to approximate the average detection probability, but the change in the average is really minimal for different calculation methods.

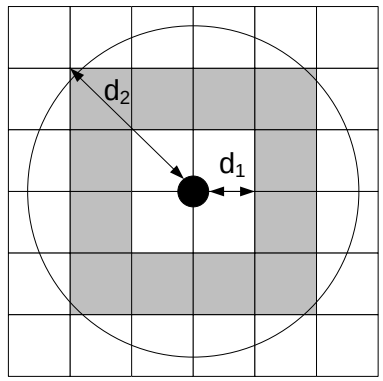


Figure 2.6. Process of discretizing the detection probability

Moreover, assuming non-binary detection probability leads to integration of uncertainty into the modeling approach. If we define an indicator function for preparedness of each cell (i,j) as *I{cell (i,j) is prepared against attacks (or 100% under detection)}*, then the expected preparedness for that cell under coverage of sonars is the term $[(1\text{-}dp_i)\cdot(1\text{-}dp_j)\ldots(1\text{-}dp_s)]$ in equation (2.15). Hence, the expected unpreparedness or vulnerability is $1\text{-}[(1\text{-}dp_i)\cdot(1\text{-}dp_j)\ldots(1\text{-}dp_s)]$. In other words the objective function (equation (2.8)) in the mathematical model includes the expected vulnerability in it. This methodology is used widely in the literature to embed the uncertainty in detection probability into the sensor placement model (Dhillon and Chakrabarty 2002; Bar-Noy, Brown, and Shamoun 2010; Cavalier et al. 2007; Golany et al. 2009).

2.5.3. Significance Values Calculation

As discussed earlier and observed in the model the cell significance values (a_{ij}'s) play a very important role in the allocation scheme that the model proposes. That is, the higher the significance values in a section (consisting a group of cells) of the environment are (relative to the other sections), the higher the probability of sonar allocation to that section is. Hence, while

defining these values all the factors that may increase the chances of a terrorist attack to a specific section of the port must be considered; e.g. the relative significance of infrastructures in the port and their proximity to water, the maritime transportation routes and their relative importance regarding types of cargo being carried, the frequency of ferries passing through and the passenger capacities of ferries, among others.

Moreover, some of the information cannot be gathered through data, such as the criticality of each infrastructure for the nation's well-being. These need to come from a detailed study of the infrastructures and their effect on the nation's survival. Another approach for estimating such information is to elicit from expert opinions. This is a complicated task by itself. Each specific port needs an elicitation process with customized questions and experts who know the port and its significance to gather the required information.

2.6. Conclusion

A mixed integer linear programming model is proposed to place sonars underwater to mitigate the risk of terrorism that might be instigated via the water side in ports and waterways. The proposed model contains a risk minimization objective function along with constraints to ensure plausible placements under a budget limitation. The approach requires discretizing the environment by putting a grid of cells on the field of water under study. The size of the grid cells depends on the geography, desired level of accuracy and sonar specifications. A simple and straightforward heuristic algorithm is presented to calculate the cell size. Finding the cell size helps us to specify the model parameters that depend on the cell size, such as the number of cells, the types of detection probabilities to allow in the model and significance values of the cells.

The model features two key issues for the sensor placement problem, which are usually missing in the literature. In case a cell is covered by multiple sensors, their detection probabilities are aggregated for that cell. The range dependency of the detection probability is also considered.

The model is presented in two stages. The first model is quite simple and it provides an understanding of the elements of the objective function and shows how the constraints guarantee the coverage. The second model basically builds on the basic model, but it involves more realistic scenarios. Since the latter is a quadratic mixed integer programming problem, a linearization approach is presented.

The computational complexity of the proposed model is discussed in Chapter 3 and a fast heuristic technique is presented as a solution methodology for the model. Then a number of test cases are created and used to show the flexibility of the model and performance of the heuristic technique.

3. The Heuristic Algorithm and Test Examples

3.1. Introduction

The probabilistic risk model of Chapter 2 features a number of significant details of the sensor placement problem such as multiple detection and range-dependent detection probability. It also provides the flexibility of using various types of sonars with different ranges and angular coverage (such as 90˚, 180˚ and 360˚). In order to use the model for a body of water a few preprocessing steps are required to be done. These steps include calculation of cell size, discretization of detection probability and calculation of significance values. The latter ones are directly affected by the cell size calculation. Once the cell size is calculated, the total number of cells required to cover the environment can be calculated as well. The model has the flexibility to use different number of sonars for surveillance according to the budget constraint. The higher the budget is, the greater the number of sonars becomes.

These features make the model so general that it becomes appropriate to be used in real world problems without undergoing many simplifications. However, computational complexity is the drawback for such a model. In other words, after modeling a real world scenario with the methodology of Chapter 2, one needs to worry about the way to reach the optimal placement of sonars. Number of cells, number of sonar types, budget constraint, multiple coverage of sonar and range-dependent detection probability are factors that increase the complexity of the model.

In this Chapter, first a test example featuring multiple sonar types, multiple coverage of sonar and range-dependent detection probability, but with low budget and low number of cells is presented to show how big the problem becomes even in case of small number of cells and sonars. Then the effect of these parameters on the complexity of the problem and the feasible set is discussed. Previous articles have also shown that this problem is quite complex and solving it for optimality for large-scale problems may not be feasible. As a result a heuristic solution technique is devised

to provide reasonably good solutions for large-scale problems. A number of test instances are run to contrast the performance of the heuristic and a solver package.

3.2. A Test Case

In order to provide a realistic scenario for the comprehensive mathematical model of Chapter 2 (equations ((2.8) through ((2.14)), the New York Harbor is considered to be the test case. There are a plenty of reasons to select such a strategic location to study. A diverse set of infrastructures is located all around the harbor; vessels carry different types of cargo to terminals in the port, a considerable number of ferries transport people between New York and New Jersey, among others. Figure 3.1 provides a geographical overview of the region.

Figure 3.1. An aerial map of New York Harbor (the map is extracted from Google Earth)

Initially the model parameters need to be determined, such as the cell size, a_{ij}*'s*, detection probabilities and sonar types. Since a full-scale elicitation process is not undertaken, the a_{ij}'s are calculated based on the number of routes crossing each cell, the frequency of vessels using the

paths and the type of cargo each vessel is carrying. Using these factors, the grid for the harbor and the color coded a_{ij}'s will possibly look like the one in Figure 3.2.

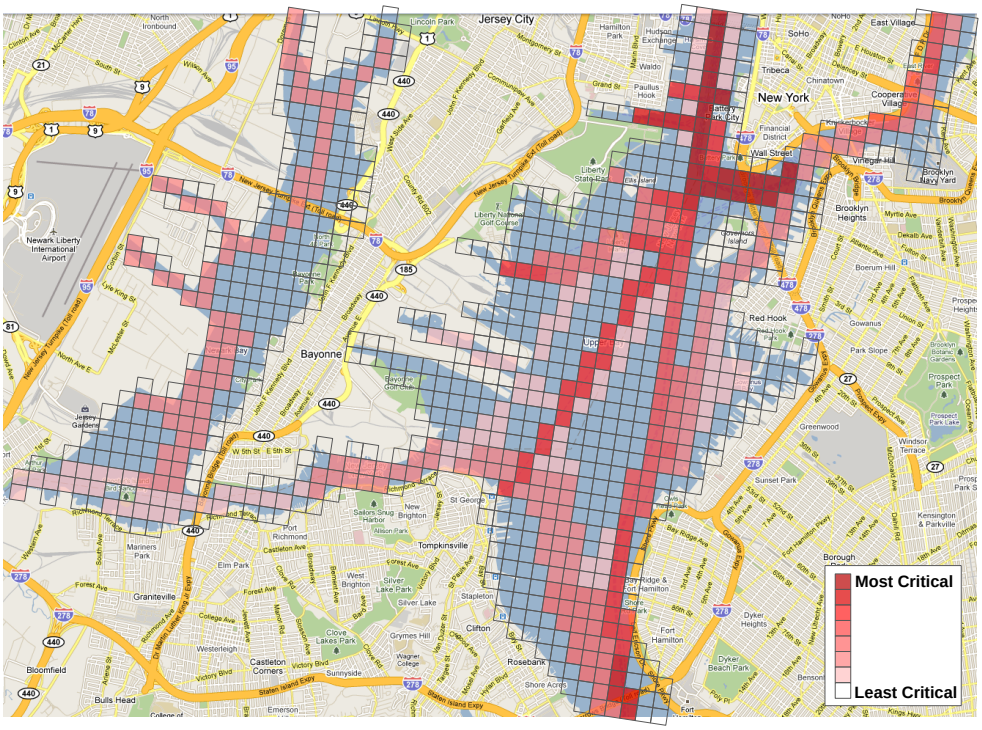


Figure 3.2. Criticality of cells calculated based on vessel routes (the background map is extracted from Google Maps)

Three types of sonars with 360°, 180° and 90° angular coverage are used in the model with ranges of 750, 900 and 1000 meters and prices of \$48, \$37 and \$26 respectively[1]. The cell size is chosen to be 300 meters. The square shown in Figure 3.2 is covered by approximately 65·55=3575 cells. Since more than half of the cells are not located on the water, they are removed from the picture (and also from the analysis). Also, three types of detection probabilities are assumed to exist in this model. Accordingly as shown in Figure 3.3, the closest cells to the sonar (the darker ones) having probability of detection (*dp*) equal to 0.99, the mid class cells have a *dp* of 0.95 and the lightest cells a have *dp*=0.9.

[1] The real cost of sonars is in the scale of thousands of dollars.

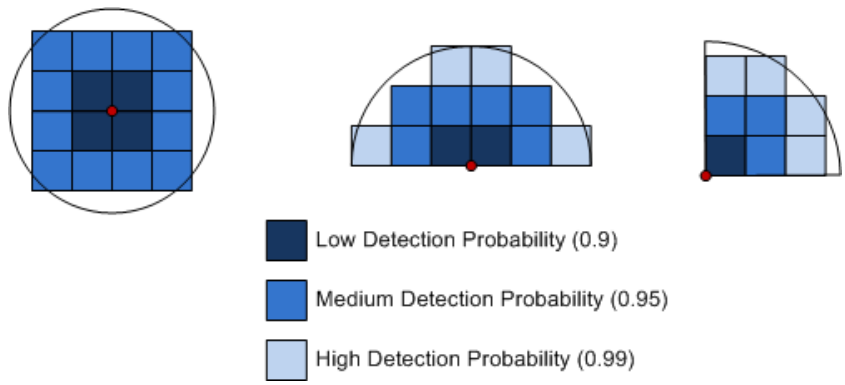


Figure 3.3. Types of detection probabilities (coverages) to be used for the 3 sonar types in the example

After defining all the parameters, it is possible to find the optimal sonar allocation for the test case problem. Since this case does not have a large number of variables (due to relatively large cell size) it is still possible to solve the problem with an objective value close to the true optimal. In order to reach a competitive solution in a reasonable amount of time, an α percent relative optimality tolerance is enforced, such that the solution procedure is stopped as soon as the current solution reaches the $1\text{-}\alpha$ percent of the optimal solution (solver first relaxes some of the constraints to reach a lower bound to the optimal value and then according to this lower bound, the solver can assure that each feasible solution is in what gap of the optimal solution). The CPLEX 12 (IBM 2011) solver is used for this purpose and it takes almost 50 hours to reach the solution. The solution for a specific budget constraint (b=\$400) is presented in Figure 3.4. As Figure 3.2 and Figure 3.4 suggest, the model is sensitive to characteristic values and places sonars accordingly.

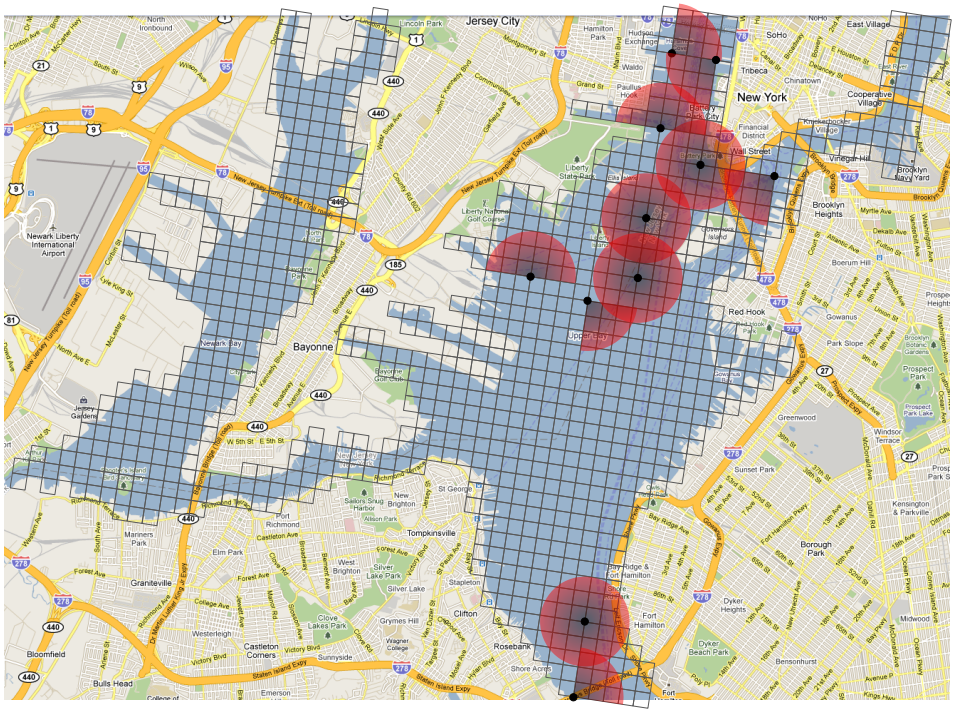


Figure 3.4. A sonar placement scheme for the New York Harbor (based on the characteristic values obtained from vessel routes)

3.3. The Heuristic Solution Technique

As seen earlier, solving the full model explained previously for large-scale fields of water with details is quite challenging (the problem is NP hard; see Aspnes, Goldenberg, and Yang (2004)). There are parameters that increase the complexity of the model rather exponentially and make it difficult to solve for the optimal solution; namely the number of sonar types (q), budget (b) and number of cells (c). The number of sonar types is typically few (due to compatibility with the environment). It is reasonable to ignore this parameter in dealing with the complexity of the model. Unfortunately this is not the case for the budget and the number of cells. Budget is the variable that the decision makers impose on the model and it can have any value (although often limited), and the number of cells is the parameter that defines the accuracy of the output and is expected to be large. In order to still benefit from the model for such scenarios with acceptable

accuracy, a heuristic solution technique that is fast (does not grow exponentially in budget and number of cells) and accurate is needed. In this section such an algorithm is proposed and discussed for its accuracy and runtime.

The proposed solution method resembles the so called greedy algorithm, where a locally optimum solution is chosen at each iteration of the solution process. Suppose that a grid of cells (Λ) with known characteristic values and q types of sonar with corresponding detection probability p_k, range r_k and price s_k ($k=1,\ldots, q$) are available. The set of candidate points for sonar placement (Δ) is also known. Before getting into the iterative placement algorithm, in step 1 the method looks at the total budget for sonar placement, and then searches among all possible combinations of sonars that exhaust the budget and chooses the combinations that leave the least unused budget and puts them in a set called the dominant set D. For example, suppose in a simple problem with two sonar types (first type costing \$3 and the second type costing \$5), the budget is \$14. Among all possible combinations of these two types, the one with three sonars of the first type and one sonar of the second type consumes the budget the most (and probably will generate more risk reduction compared to other combinations). The dominant set may include one or more elements denoted by X_w, where $w=1,2,\ldots,|D|$. If the cardinality of the dominant set is high, a number of combinations are selected at random to be considered in the iterative algorithm, since going through several combinations of sonars helps the model search in more sections of the feasible region to find the optimal solution.

Next, the main part of the algorithm (the iterative placement process) is executed for each element of the dominant set. For each element X_w, the iterative algorithm is run until all sonars of the element are placed. In every iteration, the algorithm goes through all candidate locations in Δ. For each point, the algorithm calculates the amount of risk (characteristic value) that each type of sonar will cause in case it is placed in that location. Considering all candidate points provides the risk reduction (v_{ijk}) for each cell (i,j) by sonar type k. Then, for each sonar type, the maximum risk

reduction ($v_{max,k}$) and its location $(i,j)_k^*$ is obtained. Finally the division of the maximum risk reduction by the price of the corresponding sonar type ($v_{max,k}/s_k$) is the criterion for sonar selection in every iteration. The higher this value is, the more that sonar reduces risk per dollar of budget. In fact this ratio comes from the greedy algorithm (Dantzig 1957) which is typically used for the knapsack problem. The only difference between the knapsack implementation and this technique is the need to update the criterion ratio for each sonar type in the greedy technique at the end of each iteration (after placing a sonar). In fact when a sonar is placed, the corresponding a_{ij} values (characteristic value of the cells that the placed sonar is covering) need to be updated as shown in

Figure 3.5. Then that sonar is removed from the current element of the dominant set. Since some of the characteristic values are updated, the maximum risk reduction for each sonar type needs to be calculated again for the next iteration.

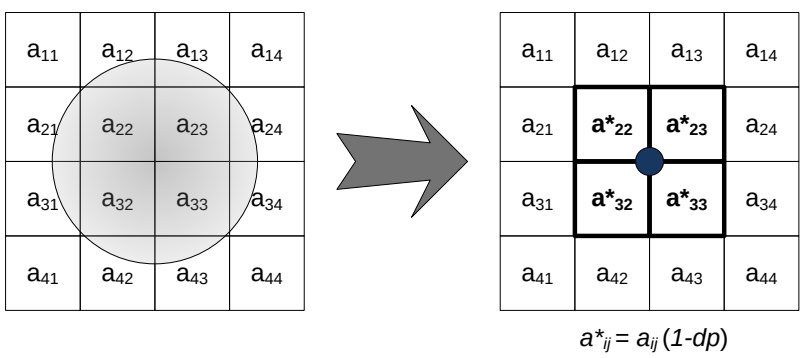


Figure 3.5. Updating a_{ij}'s

The algorithm loops until all of the sonars in that element of the dominant set are placed. The summation of the final characteristic values is the objective value O_w for this placement scheme. Then, the same process is repeated for all other elements of the dominant set. The final result of each element is called a placement scheme. Consequently a placement scheme for each of the elements of the dominant set is achieved. To further improve the solutions, for each of these solutions a complimentary process is performed, such that all combinations of one, two, three and

four sonars are chosen to be removed temporarily from the placement scheme and replaced by new sonars that the greedy chooses (as much as the budget constraint permits to do so). This process helps to visit various sections of the feasible region and avoids the model becoming stuck at a local optimum. Throughout this complimentary process the objective value is observed and whenever it falls below the current value, the replacement becomes permanent and the process continues until all combinations are evaluated. When the complementary process is executed for all the placement schemes, the one with the lowest objective function is returned as the final solution. The algorithm representation of the greedy approach is presented below.

Heuristic Greedy Algorithm:

Step 1:

Find all $X_w = \{x_{w1}, x_{w2}, \dots, x_{wq}\}$ such that $s_1 x_{w1} + s_2 x_{w2} + \cdots + s_k x_{wq} \leq b$ (where x_{wk} is the number of sonars of type k in the wth combination and $w=1,2,\dots,R$ where R is the number of possible combinations of sonars that exhausts the budget)

Step 2:

Construct the dominant set D, a subset of X_w's with the minimum remaining budget of $b - (s_1 x_{w1} + s_2 x_{w2} + \cdots + s_k x_{wq})$

Step 3:

$\forall\, X_w \in D\,, w = 1,2, \dots, |D|$ //For any element of the dominant set

While $|X_w| > 0$ //While the element is not empty

$\forall\, x_{wk} > 0, k = 1,2, \dots, q$ //For sonar types that still exist in X_w

$\forall\, (i,j) \in \Delta$ //For any candidate point to place a sonar

Calculate the risk reduction for sonar type k, if it is placed at (i,j)

Find the maximum risk reduction $v_{max,k}$ at location $(i,j)_k^*$ for sonar type k

Obtain $f = argmax_k\{{}^{v_{max,k}}/_{s_k}$, k = 1,2,..,q} and place a sonar of type f at $(i,j)_f^*$

$x_{wf} = x_{wf} - 1$ //Remove the sonar from the combination set

If $x_{wf} = 0$, then $|X_w| = |X_w| - 1$

Evaluate the objective function O_w for this combination set (X_w)

<u>Step 4</u>: //Search of an improved solution

$\forall\, X_w \in D\,, w = 1,2,\ldots,|D|$

$\forall\, i \in \{1,2,3,4\}$

Remove all combinations of i sonar(s) at a time, ant try to replace them (it) with sonars of highest ratio of $v_{max,k}/s_k$ (max risk reduction/sonar price) and recalculate the objective function $O_w^{'}$

If $O_w^{'} < O_w$, then confirm the replacement

Else undo the replacement

Find the set with lowest O_w and choose it as the best solution

3.4. A Discussion on the Heuristic Algorithm

To check the efficiency of the proposed algorithm, first its computational complexity is compared with the size of the feasible set, and then its results and runtime are compared to the solutions obtained from the CPLEX 12 solver called from the GAMS (GAMS 2012) software.

Suppose that the budget is b and there exist q sonar types to choose among and c cells in the field. If the average sonar price is p, then the number of possible selections to exhaust the budget is equal to the number of integer solutions to the following equation,

$$s \cdot x_1 + s \cdot x_2 + \cdots + s \cdot x_q = b \tag{3.1}$$

or approximately equal to the number of integer solutions to

$$x_1 + x_2 + \cdots + x_q = \left\lfloor \frac{b}{s} \right\rfloor = n \tag{3.2}$$

where x_i represents the number selected from sonar type i. The number of possible solutions to this problem is C_{q-1}^{q+n-1}based on (Murty 1981).

Moreover, for each of the possible selection above, there exists approximately $\frac{P_n^c}{\prod_{i=1}^{q} x_i!}$ different placement schemes. Putting these together, results in $C_{q-1}^{q+n-1} \cdot \frac{P_n^c}{\prod_{i=1}^{q} x_i!}$ many different possibilities in the feasible set. Basically the time to find the solution is of order $O(c^n)$. For the heuristic algorithm after finding the best combinations of sonars which also requires C_{q-1}^{q+n-1} possibilities to be evaluated, the sonar allocation takes place by checking all cells for each type of sonar for each iteration of placement. Therefore in each iteration $c \cdot q$ possibilities are checked, and there are n iterations (sonars) in total, resulting in $c \cdot q \cdot n$ possibilities. Finally in the neighborhood search process, various combinations of two, three and four placed sonars are temporarily removed from the model. Then the main part of the algorithm is called again to place new sonars in the model. If these new sonars result in higher risk reduction than the removed sonars, the removal and placement become permanent. Since at most four sonars are removed at the same time there will be around $C_4^n \cdot c \cdot q$ possibilities to check in this process which is a polynomial of degree four for the number of cells and linear for the budget and the number of sonar types. Altogether the heuristic algorithm is of order $O(n^l)$ where l is $max\{4,q\}$ (as mentioned before in real case problems the number of sonar types rarely exceeds two or three).

Moreover it is possible to modify the algorithm for large scale problems such that it runs much faster; e.g. one can skip finding best combinations set and just begin with placing sonar until the budget is consumed (thus skipping C_{q-1}^{q+n-1} possibilities in step 1) or one can just look at combination of a subset (e.g. 2) sonars instead of going from one to five sonars in the final step

(removal and replacement process). As a result, the computational complexity of the algorithm reduces to *O(n)*, while the final solution is still close to optimal (see that the modified heuristic results in Table 3.2 to compare the speed and accuracy of this approach with the heuristic itself). Clearly the number of possibilities to check in this heuristic is much less than the ones in the feasible region, which makes the algorithm applicable for large scale problems with a reasonable level of accuracy. In Table 3.1 each of the three main parameters are doubled to show how the number of possibilities change in the heuristic algorithm and the feasible set. One can see how dramatically the number of possible sonar placements increases in the feasible set, while for the heuristic algorithm the increments are minor.

Table 3.1. Effect of model parameters on the heuristic and possibilities space

Parameter change			**Impact on the heuristic model**	**Impact on the number of possible cases**
parameter	**From**	**to**		
number of cells	c	$2c$	twice more calculation	2^{2n} times more possibilities
Budget	n	$2n$	Does not require much more than twice calculations for small values of q (q less than 10)	At least c^n times more possibilities
Sonar types	(q<5)	$2q$	twice more calculation	Around n^2 times more possibilities

It is also important to mention that when the density of sonars is not high (sonars are sparsely distributed in the environment and their coverage overlap is minimal), the results of the heuristic algorithm are close to the results from solver package (see the upper rows of Table 3.2 and lower rows of Table 3.3). Fortunately this is the case in most of the real scenarios, where due to the budget limit the firm desires to reach the highest protection although the environment is not under coverage of sonars thoroughly.

3.5. Test Instances

In order to determine how close the results of the algorithm are to the optimal solution and also compare its runtime with the CPLEX solver, the results of a number of small test cases are generated using the heuristic algorithm and the branch-and-cut technique used in the solver (note that the solver package does not go over all possible situations in the state space of the problem, and hence the runtimes are not comparable to the results in Table 3.1). The same computer was used for all the runs with an Intel Xeon 3 GHz CPU. To reach a comparable solution with the solver in a reasonable amount of time, a three percent relative optimality tolerance is enforced, such that all solutions from the solver package are guaranteed to be within a three percent interval of the optimal solution (In order to reach solutions in a reasonable amount of time, we had to follow this process). The results for the solver in Table 3.2 and Table 3.3 are generated accordingly. As the budget constraint is more relaxed (budget increases), the execution time of the solver increases exponentially as shown in Table 3.2, such that it becomes impractical to obtain a solution in a reasonable amount of time. That is why some cells in the "Time" column of Table 3.2 have the value of ">100000" in the solver solution section. It means that in 100000 seconds the solver could not guarantee that the current solution is in the 3 percent gap of the optimal solution and the current solution is returned.

Table 3.2 provides a comparison of the results of the solver, heuristic and modified heuristic for a number of synthetic cases with different budget levels. For each case five similar scenarios are generated and the results are averaged to provide more robust results. A sample scenario case with 300 cells is illustrated in Figure 3.6. The other four scenarios have the same configuration (a grid of 10×30), but with slightly different characteristic values.

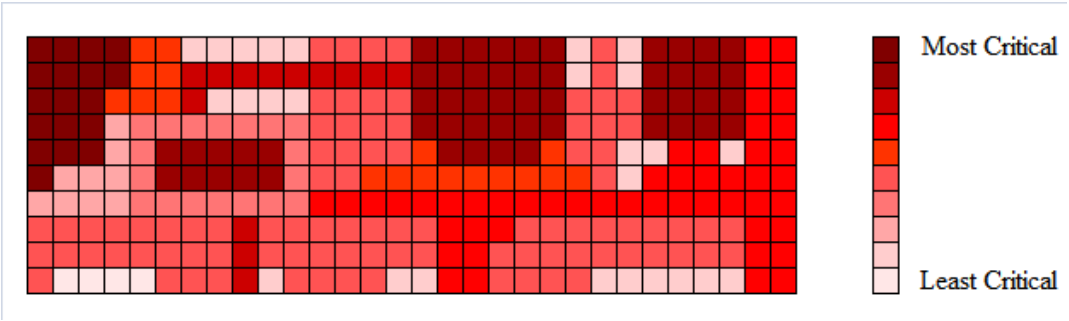


Figure 3.6. A typical test case scenario with 300 cells

The lowest value of the budget (100) corresponds to the case with at most three sonars (according to sonar prices). Note that the values of the budget are monetary, but the dollar sign is dropped for simplicity. In this case the sonars are scarce in the environment. The highest budget value translates to sonar abundance, such that almost all cells (with positive a_{ij}) are covered by sonars. A number of intermediate cases in increments of 100 are also provided to illustrate the effect of sonar density on the accuracy of the heuristic and also the runtime to reach the solution. The number of cells is 300 (c=300) for all the cases in Table 3.2 as shown in Figure 3.6.Three types of sonars (q=3) with 360˚, 180˚ and 90˚ coverages were considered. The sonars cost \$48, \$35 and \$26, respectively.[2] The average value of the objective function (summation over all a_{ij}'s) over all five scenarios is 21861 before placing sonars. Table 3.2 shows the dramatic increase in the runtime of the solver as the number of sonars increases. The heuristic algorithm solves each case in a reasonable amount of time, while the modified heuristic solves all the cases in less than a second. On the other hand, the value of the objective function is quite close in all three methods, especially when the budget is low. Figure 3.7 depicts the results of objective values for the solver and heuristic technique from Table 3.2 graphically. Clearly, the heuristic method provides better solutions in low budget scenarios as the graphs in Figure 3.7 suggests. Due to the greedy nature of the heuristic method, it performs better in scenarios with smaller feasible sets. However, as the

[2] The real cost of sonars is in the scale of thousands of dollars.

number of sonars increases and the multiple coverage issue appears, the solver outperforms the heuristic method.

Table 3.2. Comparison of solver and heuristic as budget increases

Budget Constraint (*b*)	Solver		Heuristic		Modified Heuristic	
	Objective Value	Time (sec)	Objective Value	Time (sec)	Objective Value	Time (sec)
100	18557	19	19497	<1	19497	<1
200	15464	235	16654	<1	16654	<1
300	12701	962.6	14059	2	14059	<1
400	10204	19359.6	11779	5	12340	<1
500	7915	>100000	9652	10	11829	<1
600	5845	>100000	7693	21	9879	<1
700	3918	>100000	5800	37	7876	<1
800	2776	>100000	4157	60	6278	<1
900	1306	>100000	2921	98	5047	<1
1000	471	>100000	2121	147	3978	<1

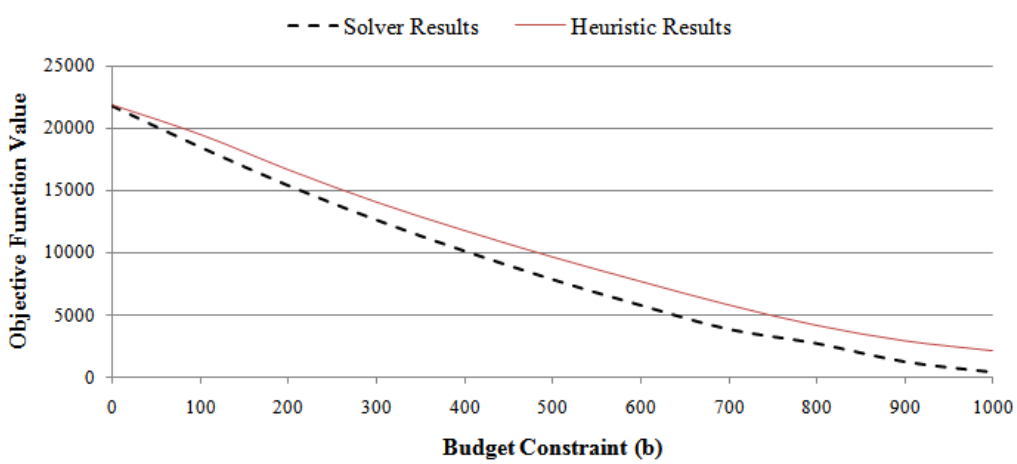


Figure 3.7. Performance of the heuristic algorithm as budget increases

In order to observe the effect of the number of cells on the runtime of the solver and the heuristic, the results of 8 cases with different number of cells are provided in Table 3.3. In this setting, a

grid with 300 cells is used as the base case. For the sake of easier comparison for the various number of cells, the base grid is duplicated to build fields with larger number of cells; e.g. for the second row of Table 3.3, two base case grids are combined to obtain the grid for this instance (the objective before placement for the second row is slightly lower than twice the first row due to changes in a_{ij} of boundary cells when two grids are put together to build a bigger one). The same logic is used to build the rest of the cases in this table. That is why the total risk increases proportional to the number of cells. For this set of instances the budget is set to be 300 and again three types of sonars are used. Again, for the sake of robustness of results, 5 similar scenarios are generated for each case and the results are averaged and given. The results of the heuristic and the solver are quite close, while the runtime of the solver is much more than the heuristic. The values of the objective function for the heuristic technique are even smaller than the ones for the solver for the lower rows of Table 3.3. This is again due to scarcity of sonars in such cases. Since the budget is kept fixed, as the number of cells increase the density of sonars in the field decreases and the greedy heuristic results in closer-to-optimal solutions.

Table 3.3. Comparison of solver and heuristic as number of cells increase

No. of Cells (*c*)	**Objective before placement**	**Solver**		**Heuristic**	
		Objective Value	**Time (sec)**	**Objective Value**	**Time (sec)**
300	21861	12701	962	14059	2
600	41846	32253	17207	32527[3]	2
900	61157	51294	12265	51389	3
1200	83450	73580	490	73566	3
1500	107838	97871	792	97561	4
1800	127184	117189	824	117014	5
2100	145824	135996	1160	135496	5
2400	166914	157177	1296	156541	12

The results of Table 3.3 are used in Figure 3.8 to illustrate the performance of the heuristic. The results from the solver and the heuristic should be projected on the left vertical axis and the objective difference needs to be read from the right vertical axis. The point in which the objective difference crosses zero and goes below that is where the heuristic starts to perform better than the solver. Notice that the three percent optimality gap is also enforced here to obtain results from the solver.

[3] The odd behavior of the runtime for the solver in Table 3.3 is due to the branching rules that the solver uses.

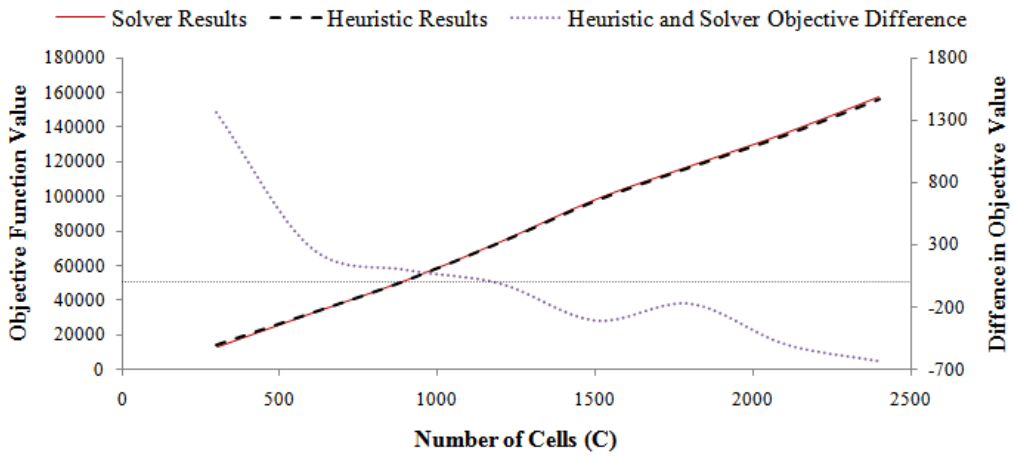


Figure 3.8. Performance of the heuristic algorithm as the number of cells increases

3.6. Conclusion

The optimization model that presented in Chapter 2 is NP hard and it is difficult to solve it for optimality for large-scale problems. A small example in the beginning of this Chapter shows how long it takes to solve this problem to reach the optimal solution. Then the complexity of the sonar allocation problem and the effect model parameters on the complexity is discussed.

A heuristic algorithm is developed to achieve close to optimal solutions in a much shorter time. The heuristic works like the one used to solve the knapsack problem developed by (Dantzig 1957). The difference is that in our approach the utility of each object type changes after assignment of a single object to the knapsack. This utility needs to be recalculated for each object type after each assignment. Hence, the method is iterative and it continues until the budget for sonar placement is exhausted. In order to show the performance of the algorithm, a number of test cases are generated and run with both the heuristic algorithm and a commercial Solver (CPLEX). The problem parameters such as number of cells and the budget constraint (number of sonars) are also varying in these test examples to show their effect on the runtime of the heuristic and the

Solver package. It is shown that the heuristic complexity is of order $o(n^l)$, where n is the number of sonars to place and l is dependent on the sonar types. Since the number of sonar types to be allowed in the model is not large in real world problems, the heuristic can be applied to solve large-scale problems as well. A modified version of the heuristic is also presented that does not go over the last neighborhood search and it stops as the budget is depleted. The modified heuristic is of order $O(n)$.

The test cases show that the heuristic algorithms perform very well when the budget is limited and the density of sonars in the environment is low. As the number of sonars in the field increases the optimality gap for the heuristic algorithm increases as well. This observation makes the heuristic useful for real world large-scale problems. In such cases the budget is low and the operator seeks to reach the highest risk reduction with a limited budget in hand.

4. Background on Game Theory

4.1. Introduction

The probabilistic modeling approach of Chapter 2 is a powerful resource allocation tool to protect a field of interest against potential undesirable events. As mentioned earlier, variations of this model have been widely used in the literature for resource allocation in different applications. However, the use of this approach in security risk analysis problems has come under scrutiny. Studies such as Golany et al. (2009), Bier, Cox, and Azaiez (2009) and Cox (2008) believe that the probabilistic risk approach does not model the intelligent behavior of the opponent (in case of a security threat). In other words, in most applications where the probabilistic resource allocation is used, the undesirable event is known a priori or at least it is predictable, while in security risk analysis this is not the case.

In the context of security risk, the opponents are a group of people who decide where, when and how to attack to maximize the loss they are interested in. Clearly, it is not smart to treat such a plan as an event happening by chance (probabilistic risk models). A modeling approach that is able to consider the attackers interests can perform much better. Strategic modeling techniques are instances of such techniques. They are capable of modeling the behavior of the adversary based on his interests and propose defensive and preventive actions accordingly.

A suitable strategic methodology is the game theoretic modeling approach. Due to its nature, a game is played by players who seek to win the game (obtain their interests) based on their wisdom and notion of other players' actions. The security risk analysis problem has a similar structure. Consider a terrorist (attacker) as a player who tries to maximize his damage to an infrastructure and the defender as a player who seeks to minimize the chances and consequences of a terrorist attack. These two players have conflicting interests and they both seek to maximize their achievements based on their conception of each other. This setting matches the idea of a

game perfectly. Based on this rationale, the game theoretic modeling approach has been frequently used in the literature of security risk analysis.

The rich literature of game theory empowers the application of this general modeling approach in security risk related problems. Recently, a great number of security risk analysis literature articles have utilized game theory in their studies in various settings and with different assumptions.

Before getting into the literature review, a few game theoretic definitions are introduced in the next section. Then the literature review section discusses the selected articles of the literature which are more related to the subject of this study. Two basic though powerful concepts that are helpful in the modeling and solution process are then explained. The modeling effort which seeks to fit the problem of interest in this study to a well-known type of game is then followed by its solution methodology. A test case is presented at the end of this Chapter to show the operation of the presented model. This approach is extended later in Chapter 5 to reach the strategic model that is comparable with PRM.

4.2. Basics of the Game

In general, a game consists of two (or more) players who compete with each other and try to maximize their benefits (minimize their losses) by taking actions at each stage of the game, based on the actions of the other players (or a perception of it). Each game is defined if the following four questions can be answered; who the players are, what are the possible actions, when each player gets to play, and how much is the gain. These factors are characterized by the rules of the game. In order to propose a strategy of how to play a game these factors need to be known.

For a game, after specifying the players i=1,2,…,p the set of possible actions A_i for each player i (i=1,2,…,p) needs to be defined. The action set of a player at each stage of the game is the set of possible strategies (movements) that the player is allowed to play. Each action (move) of a player is associated with a payoff that may be negative or positive. This payoff is usually dependent on

the actions of the other players, too. Hence the payoff for each player is calculated after all the players have played; e.g. in Figure 4.1 if player 1 decides to take action 1, his payoff will not be completely specified until player 2 chooses his action. In this case if player 2 chooses to play action *A*, the payoff for player 1 is 30. Note that the first number in the duplet in each cell of the game matrix is the payoff for the first player (row player) and the second number is the payoff for the second player (column player) when the players take the corresponding row and column actions.

The games can be classified based on various properties. From one perspective, if the players of the game are committed to a unique goal, the game is called a cooperative game, otherwise it is non-cooperative. Usually it is assumed that information transmission is allowed in cooperative games such that each player knows the actions of the other. From another point of view a game can be simultaneous or sequential. In a simultaneous game, the players perform their action at the same time or if they do not play simultaneously the later players are not aware of the actions of the earlier players. If the later player has some sort of information about the action of earlier players the game is called sequential.

A game is called zero-sum if the sum of the payoffs for all players for any combination of actions is zero while in a general-sum game it can be any number. This categorization is important due to some characteristics of zero-sum games that make them easier to solve. Figure 4.1 illustrates the normal form of a two player game. Player 1 has two possible actions and player 2 can choose among three possible actions. Since the sum of payoffs for each cell of the table is zero, the game is called a zero-sum game.

While playing a game the players have the choice to either always perform the same action or choose among a set of actions according to a pre-specified probability distribution. The former strategy is called the pure strategy while the latter is the mixed strategy; e.g. in Figure 4.1 if

player 1 decides to play action 1 all the time he is following a pure strategy and if he chooses to play action 1 70% and action 2 30% of the time he is playing according to a mixed strategy.

		Player 2		
		A	**B**	**C**
Player 1	1	30,**-30**	-10,**10**	20,**-20**
	2	-10,**10**	20,**-20**	-35,**35**

Figure 4.1. Normal form of a zero-sum two player game

These definitions and classifications are useful in understanding the literature of game theory application in security risk and also the methodology that has been developed in this study.

4.3. Literature Review

In the security risk analysis literature most of the game theoretic models fit to the following settings:

The game consists of a defender (or a set of defenders) that has some resource (budget) on hand and he is seeking the optimal way to allocate the resource among his assets to protect the assets against the attacks by an external intruder. The resources are expended on preparing the assets against potential attacks. Generally the more resources the defender assigns for one set, the less that asset is vulnerable to attacks. On the other hand, the attacker's goal is to select an asset (a set of assets) among all and invade it (them). Hence the defender is trying to minimize his loss in case of a terrorist attack and the attacker wants to maximize his damage to the asset he chooses (Sandler and Lapan 1988; Golen, Mishra, and Shenoy 2010; Powell 2007; Golany et al. 2009; Zhuang and Bier 2007). Sandler and Lapan (1988) are among the first articles that model the security risk from a game point of view. They propose a simple solution technique for the case

with two targets to be attacked. They then discuss national and international terrorism and describes the role of information sharing between countries on the performance of each country against terrorist attacks.

Powell (2007) proposes a framework for allocating scarce resources against intelligent terrorist attacks in four settings. First a baseline case which is a classical resource allocation against terrorist attacks (as discussed in the introduction) is presented. Then it is assumed that the defender can also allocate resources for border protection. In this case the solution will lead to the distribution of resources among the sites and the borders. The third scenario is the case where the threat has strategic and non-strategic components, the same as what is studied by Zhuang and Bier (2007). The last scenario discusses the case where the defender is not confident about the attacker's preferences.

Golany et al. (2009) present the generic forms of probabilistic risk models and strategic risk models and discuss the differences of these approaches using a real world example. Zhuang and Bier (2007) propose a game theoretic optimization model to decide how to allocate the resources to protect against both natural disasters and terrorist attacks. Based on their model, they claim that it is preferable for a defender to play a sequential game rather than a simultaneous one against an intelligent attacker. That is, the defender prefers to play first and advertise his preventive action instead of keeping it secret. The intuition is that in case the attacker knows the preventive actions of the defender, the optimal solution he reaches is worse than the case where he knows nothing about the defender actions. It is true because the defender's actions work like constraints for the attacker's problem and hence make the attacker's optimization problem more constrained. On the other hand, if the attacker plans to attack without any prior information about the defender's actions he will choose a site which is already protected by the defender with higher probability. In this case the attacker's harm to the site will be less than he expected and it can be a benefit to the defender. Zhuang and Bier (2010) discuss these counter arguments.

Unlike the article by Zhuang and Bier (2007), most of the game theoretic models in security risk analysis are considered to be non-cooperative (Kardes 2005; Paruchuri, Pearce, and Kraus 2008). In such models, both the defender and attacker may benefit from the reliable information they might gather from the other party's actions. Kardes (2005) introduces robust stochastic games and present robust optimization techniques for optimal strategies under uncertainty about the opponent. Paruchuri, Pearce, and Kraus (2008) provide a model where a defender plays the game against multiple attackers and optimizes resource allocation to protect against all attackers. In this article the authors use the MinMax approach and integrate the primal and dual problem into one optimization model. Kunreuther and Heal (2002) propose a game model with multiple defenders playing against an attacker. They also apply their model into airline security in a later work by Heal and Kunreuther (2005).

Although the settings defined in these articles are similar, they all make different assumptions and use various modeling and solution techniques. Golany et al. (2009) and Golen, Mishra, and Shenoy (2010) use the linear programming approach for solving the proposed optimization model. Sandler and Lapan (1988) utilize the concept of derivatives to solve their problem and Zhuang and Bier (2007) use the NE to obtain the optimal solution. Powell (2007) also uses the concept of NE to come up with solution ideas.

Most of the aforementioned articles examine the numerous infrastructures of a nation as far apart nodes that do not have any effect on each other; e.g. they assume that allocating resources to one site has no effect on any other site. However in this study the body of water is the environment under study and it is discretized into relatively small cells. Clearly allocating resources (sonars) into one cell has a significant impact on the surveillance in an adjacent cell. To elaborate more on this issue let us discuss it considering the most relevant article of the literature to the work presented in this study. Golen, Mishra, and Shenoy (2010) propose a game theoretic solution technique to allocate sensors (sonars) in an underwater environment to protect a part of ocean or

open sea against terrorist attacks. In this article the environment is divided into sectors (cells) such that the acoustic characteristics of the water are homogeneous in each sector. The authors assume that the cells are large enough that allocating sonars in one cell has no effect on the protection of other cells. They also allow multiple sonars to be assigned to one cell. The usage of this model for protecting vast sections of oceans is reasonable, however it does not fit quite well in the context of protecting ports and waterways (due to their smaller size compared to large sections of the ocean).

In order to obtain homogeneous acoustic characteristics within cells in coastal waters (ports and waterways) the cell size needs to be smaller, due to a faster transition in environmental characteristics that affect the sonar coverage in coastal waters compared to ocean water according to Knauss (2005). Smaller cell size conflicts with the assumption that a cell placed at one cell has no effect on the adjacent cells. Moreover, it is not reasonable anymore to allow multiple sonars per cell. Hence the model presented by Golen, Mishra, and Shenoy (2010) becomes deficient in this context. Instead, in our proposed model it is possible to decrease the cell size arbitrarily and capture water characteristic in great detail. Since the effect of placing a sonar in one cell in the coverage of adjacent cells is taken into account, the small cell size will not lead to any modeling deficiencies.

Except for multiple sonar detection, all details that are included in the probabilistic risk model of Chapter 2 can be incorporated in the game theoretic model as well. However, to avoid ambiguities in the full-scale model, the modeling process is presented in several steps. First, a model with a few simplifying assumptions is proposed in this Chapter. These assumptions are relaxed in Chapter 5 to obtain the full-scale model that can be used for real world scenarios.

4.4. Preliminaries

4.4.1. Nash Equilibrium

Formulating a real strategic problem as a game with the characteristics specified in the previous sections can be quite involved. However the more challenging and interesting part of the effort is to find a way to solve the game and reach a reasonable solution. This procedure may be hard to achieve in the case of some complex games, but in many game settings there exist straightforward ways to solve the game.

The Nash Equilibrium (NE) is a famous solution concept proposed by Nash (1951) for the games for which the players are assumed to have a reliable perception of other players' equilibrium strategies. NE is the state of the game where no individual player can increase his benefits by changing his own strategies given that the strategy of the other players remains the same; i.e. in a 2 player game, if player 1 does his best action considering the decision of player 2 and player 2 does his best action considering the decision of player 1, these players are playing the game in an NE. As a result NE is quite interesting and widely used in the context of non-cooperative games, where the players (may) know about the payoff of others, but they are not aware of the strategies of other players (and the players cannot agree on performing specific strategies). According to Nash (1951) any non-cooperative game with a finite set of actions has at least one mixed strategy NE.

In order to better understand the concept of NE, consider the normal form of the game shown in Figure 4.2. Clearly this game is not zero-sum. Let us ignore the possibility of mixed strategy NEs and just focus on the pure strategy NEs. Finding pure NE strategies in a payoff matrix of a game is straightforward. Based on the definition of NE, player 1 needs to consider what player 2 may do. It is not reasonable for player 2 to play action *Z*, because by selecting a mixture of *X* and *Y*, he can reach higher payoffs. In the notations of game theory, it is said that strategy *Z* is a dominated

strategy for player 2. So player 1 should focus on maximizing his benefits in case player 2 plays either *X* or *Y*. If player 2 plays *X*, then the best choice for player 1 is to play *B* and if player 2 chooses *Y*, then player 1 prefers to select *A*. Hence (*A*,*Y*) and (*B*,*X*) are the 2 (pure strategy) NEs for this game, because neither of the players can get higher payoffs if they just change their own decision and the decision for the other player remains the same.

It is possible to provide a more direct way of finding (pure strategy) NEs based on the previous discussion. Consider all duplets of payoff values in the game matrix, if for one duplet the first element is the maximum value of that column and the second element is the maximum of that row then the corresponding strategy is the NE for that game. The first element should be the maximum of the column because player 1 wants to achieve the highest payoff if the action for player 2 is fixed. With the same logic, second element is needed to be the highest value of the row. It is easy to check that this condition holds for (*A*,*Y*) and (*B*,*X*) strategies in Figure 4.2.

		Player 2		
		X	Y	Z
	A	0,0	**20,40**	5,5
Player 1	B	**25,30**	5,5	35,20
	C	10,25	15,15	15,20

Figure 4.2. Nash Equilibriums (NE's) of a game

It is important to note that the NE does not always guarantee the maximum payoff for all players. In some games there are strategies for which if the players agree on playing them beforehand they will reach higher payoffs comparing to NE. To show this argument more clearly, let us change the payoff values when player 1 plays *C* and player 2 plays *Z* from (15,20) to (30,45) as shown in Figure 4.3. The NEs of the game do not change. However if player 1 and 2 can both agree beforehand to play (*C*,*Z*) then their payoffs will be more than what they get in NE. This is an

example of the case where players can earn more than NE if they share the information beforehand. Note that if either of players decide to play his last action (*C* for player 1 and *Z* for player 2) on his own and does not let the other player know about it, and if the other player plays reasonably, the average payoff for both of the players will be less than what they get out of NE. This is the reason why NE is so important in non-cooperative games, where both players want to maximize their payoff while trying to keep their information secret.

		Player 2		
		X	Y	Z
	A	0,0	**20,40**	5,5
Player 1	B	**25,30**	5,5	35,20
	C	10,25	15,15	30,45

Figure 4.3. Nash Equilibrium versus best action

As mentioned before, most of the game theoretic models in the field of security risk are non-cooperative and the usage of NE is quite common in the literature. As the game of interest in our study is non-cooperative, NE is utilized here as well. However, it is used together with the duality theory for zero-sum games that is presented next.

4.4.2. Duality Theory for a Zero-sum Game

Although it is straightforward to find the pure strategy NEs in a game, but in most cases a pure NE does not exist and one needs to find the mixed strategy NEs for such cases. On the other hand, finding a mixed strategy NE of a game is more involved than the pure strategy NE and may require extensive effort and time. However there exists a relationship between game theory and duality theory in linear programming that helps to find the mixed NE for zero-sum games. The application of linear programming theory in finding the solution for the zero-sum game is

discussed in this section. The reason for modeling the problem of interest with a zero-sum game is presented in the next section.

Consider a two person zero-sum game; players are called player 1 and 2 and the payoff matrix for player 1 is $A_{m\times n}$ (The payoff matrix for player 2 is clearly $-A$). Suppose we are after the mixed strategy NE for the players. Let us represent the mixed strategy for player 1 by vector $X = (x_1, x_2, \cdots, x_m)$, where $x_i \geq 0$ and $\sum_{i=1}^{m} x_i = 1$ and the mixed strategy for player 2 by vector $Y = (y_1, y_2, \cdots, y_n)$, such that $y_j \geq 0$ and $\sum_{j=1}^{n} y_j = 1$. Note that x_i's and y_j's are the probabilities that player 1 and 2 will select actions i and j based on. Player 1 needs to care about the strategy of player 2. Player 2 is trying to maximize his own expected payoff which is

$$\sum_{j=1}^{n} A_j y_j \tag{4.1}$$

Over all Y (A_j is the jth column of matrix A). For simplicity, it can be written in matrix form as $-AY^T$ (superscript T means the transpose operator). In other words he is trying to minimize AY^T. Now player 1 tries to maximize his expected payoff based on what player 2 has done. So he needs to maximize the minimum of XAY^T over all X. The same logic works if we begin with player 2 and we can write the expected payoff of the players as:

Expected payoff for player 1: $\max_X(\min_Y(XAY^T))$

Expected payoff for player 2: $\min_Y(\max_X(XAY^T))$

When X is fixed $XA=(t_1,t_2,\ldots,t_n)$, where t_j is the expected value of the payoff for player 2 if he chooses pure strategy j. Since player 2 wants to minimize his payoff based on A (maximize his payoff based on $-A$) he chooses the minimum t_j value. Based on this argument, the expected payoff for player 1 can be expressed as:

$$\max_X(\min_Y(XAY^T)) = \max_X \min_j(XA) \tag{4.2}$$

The right-hand side of the equation can be written as a linear program as follows:

$$\max U \tag{4.3}$$

$$(XA)_j \geq U \quad \forall j \tag{4.4}$$

$$\sum_{i=1}^{m} x_i = 1 \tag{4.5}$$

$$x_i \geq 0 \quad \forall i \tag{4.6}$$

With the same logic the expected payoff for player 2 can be written as:

$$\min_Y(\max_X(XAY^T)) = \min_Y \max_i(AY^T) \tag{4.7}$$

and linearly modeled as:

$$\min V \tag{4.8}$$

s.t.:

$$(AY^T)_i \leq V \quad \forall i \tag{4.9}$$

$$\sum_{j=1}^{n} y_j = 1 \tag{4.10}$$

$$y_j \geq 0 \quad \forall j \tag{4.11}$$

Analyzing these two linear models show that they are dual. Based on the strong duality theorem, if one of these problems has an optimal solution then the other one will have an optimal solution and the value of their objective function is equal; namely $U=V$. Hence for the zero-sum two player games, the expected payoff for both players is the same value and is called the game value. Moreover, solving the MaxMin linear problem (equations (4.3) through ((4.6)) and the MinMax linear model (equations ((4.8) through ((4.11)) leads to the optimal solution for X and Y, respectively. In fact X and Y are the mixed strategy NE solution for the original game which are obtained from solving the linear program.

Finding a solution for most linear programs with continuous variables can be done in polynomial time as a function of the size of the problem. Hence solving the linear programming equivalent form of the zero-sum game is an efficient way of finding the NE for the game.

4.5. Proposed Risk Analysis Model in the Context of Game Theory

Providing the preliminaries to game theory, in this section we attempt to fit our sonar allocation problem in a game theoretic modeling framework. Since the concept of the security risk analysis problem is a competition between a defender (who wants to protect the critical infrastructures surrounding a port) and an attacker (who tries to attack the port through water), it seems reasonable to model the problem with a game between these entities. As a result the players of the game will be the defender firm and the terrorists who plan to invade an infrastructure (or a set of them) in the port.

In order to continue with the modeling approach, a number of simplifying assumptions are required to describe the methodology in a more straightforward way. Once the modeling technique is proposed, the simplifying assumptions (except one) are removed to reach the most general scale of the model. These assumptions are:

- The discrete nature of our sonar allocation problem is relaxed; it means that it is possible for each cell to allocate any proportion of total resources (although it may not be equivalent to an integer number of sonars) to it, such that the sum of all proportions adds up to one. Later in next Chapter, the integrality constraint is added to the model. It changes the optimization model from a linear program into a binary linear program.

- Multiple coverage of sonars is relaxed; as a result, there is no advantage for a cell to be covered by multiple sonars. This assumption is critical for the modeling process. Due to limitations in the structure of the game theoretic model, it is not possible to integrate it back into the model. However, based on the conclusions of Chapter 3, this feature is not beneficial in the case of real world scenarios. In such scenarios due to budget limitations, the sonar density is expected to be low and the investor is more interested in increasing the coverage area and consequently multiple coverage of sonars is minimal.

- A single type of sonar is allowed in the model. This assumption helps to decrease the size of the payoff matrix. This assumption is also removed in the next Chapter.

- The payoff matrix of the proposed game is zero-sum. First, the general-sum version of the payoff matrix is discussed. Then an approximation is made to make it zero-sum and use the features of the zero-sum games. In the next Chapter, the solution methodology for the general-sum game is presented as well.

- Multi-cell coverage of sonars and range-dependent detection probability are relaxed; hence each sonar can cover just one cell. This assumption is removed later in this Chapter, such that each sonar can cover the cell that it is located at, as well as the adjacent cells that fall within the coverage range of sonar. Range-dependent detection probability is also brought to the model at the same time.

Now, it is possible to present the simplest case of the modeling technique. After specifying the players, the next step is to define the set of possible actions for the players. As described in Chapter 2, the body of water is divided into cells. Assuming the total number of cells to be c, the set of possible actions for the defender can be defined as $Y=(y_1,y_2,\ldots,y_c)$, where $\sum_{j=1}^{c} y_j = 1$, $y_j \geq 0$ and y_j is the probability that defender chooses cell j to protect. The action set for the attacker is defined in the same fashion as $X = (x_1, x_2, \cdots, x_c)$, such that $\sum_{i=1}^{c} x_i = 1$, $x_i \geq 0$ and x_i is the proportion of attacker effort to invade cell i.

Defining the action sets for the player specifies the size of the payoff matrix. It will be a $c \cdot c$ matrix, for which the rows correspond to the actions of the attacker with respect to each cell and the columns refer to the actions of the defender with respect to each cell. Next the payoff values are needed to be defined. For simplicity, the cell index (i,j) is translated into a single index; that is instead of representing a cell with its coordinates, it is referred to by its assigned number; e.g. the hatched cell in Figure 4.4 is called cell number 1 instead of cell (2,2). Note that with this notation, we use index i, $i = 1,2,\ldots,c$ for referring to attacker related parameters and index j, $j = 1,2,\ldots,c$ to represent defender related parameters. When $i=j$ they both refer to the same cell, but with different perspectives. Now consider the grid of cells given in Figure 4.4. The payoff matrix for the three numbered cells will be discussed next.

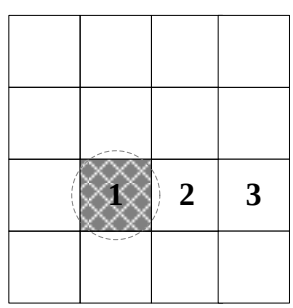


Figure 4.4. A grid with a sonar that covers one cell

In case the defender places a sonar in cell j, he undergoes cost $c_{d,j}$ (cost of defender in cell j) which includes the sonar price and installation and maintenance costs. Now consider the case where the attacker chooses the same cell to attack (i=j). Since the defender has placed a sonar in this cell, the attack will be recognized with detection probability dp. Hence the defender will only experience the additional expected cost $a_i \cdot (1 - dp)$ and his total payoff is $-c_{d,j} - a_i \cdot (1 - dp)$. If the attacker chooses to attack another cell ($i \neq j$), that cell is not under surveillance and the attack is assumed to be successful. The defender will experience the expected cost a_i and his total payoff is $-c_{d,j} - a_i$.

Now let us consider the attacker's payoff. Consider that the attacker chooses cell i to attack. He undergoes cost $c_{a,i}$ (cost of attacker to invade cell i). If the defender puts a sonar in the same cell, then the expected advantage for the attacker will be $a_i \cdot (1 - dp)$ and his payoff becomes $-c_{a,i} + a_i \cdot (1 - dp)$. If the defender does not invest on the same cell, the expected benefit for the attacker is a_i. The attacker's payoff becomes $-c_{a,i} + a_i$ in this situation.

		Defender		
		1	2	3
Attacker	1	$-c_{a,1}+a_1(1-dp)$, $-c_{d,1}-a_1(1-dp)$	$-c_{a,1}+a_1$, $-c_{d,2}-a_1$	$-c_{a,1}+a_1$, $-c_{d,3}-a_1$
	2	$-c_{a,2}+a_2$, $-c_{d,1}-a_2$	$-c_{a,2}+a_2(1-dp)$, $-c_{d,2}-a_2(1-dp)$	$-c_{a,2}+a_2$, $-c_{d,3}-a_2$
	3	$-c_{a,3}+a_3$, $-c_{d,1}-a_3$	$-c_{a,3}+a_3$, $-c_{d,2}-a_3$	$-c_{a,3}+a_3(1-dp)$, $-c_{d,3}-a_3(1-dp)$

Figure 4.5. Payoff matrix for a grid of three cells

Figure 4.5 shows the payoff matrix for a grid of three cells as defined above. The summation of payoff values for each combination of attacker-defender strategy (i,j) is $-c_{a,i} - c_{d,j}$. Clearly, this is not a zero-sum game. However, the cost of initiating an invasion by the attacker ($c_{a,i}$) and the cost of sonar placement in a cell by the defender ($c_{d,j}$) are much smaller than the expected

consequences of a successful attack in real world scenarios ($c_{a,i} \ll a_i, c_{d,j} \ll a_j$). Hence it is possible to ignore $c_{a,i}$ and $c_{d,j}$ compared to a_i and a_j. The payoff matrix can then be modified as shown in Figure 4.6. This game is indeed a zero-sum game.

		Defender		
		1	2	3
	1	$a_1(1\text{-}dp)$, $-a_1(1\text{-}dp)$	a_1, $-a_1$	a_1, $-a_1$
Attacker	2	a_2, $-a_2$	$a_2(1\text{-}dp)$, $-a_2(1\text{-}dp)$	a_2, $-a_2$
	3	a_3, $-a_3$	a_3, $-a_3$	$a_3(1\text{-}dp)$, $-a_3(1\text{-}dp)$

Figure 4.6. Revised payoff matrix for a grid of three cells

Defining the payoff matrix for the game concludes the process of modeling and makes it ready for writing the LP equivalent (as described earlier) and solving it for the optimal solution. The next step is to find a way to use the solutions of the LP problems for the sonar placement problem. The solution to MaxMin problem is not of much interest here and it can be used later to generate intelligent attack scenarios for testing and comparison of our various solution approaches. Yet the MinMax solution can be used as the guideline for the defender to find the optimal allocation of sonars to cells. As mentioned before, the y_j variable is the probability that defender chooses cell j to keep under surveillance. If we assume that the defender has a limited budget (b) in hand to protect the cells, the y_j probabilities can be used as the proportion of total budget that the defender assigns to cell j for its surveillance. This is the basic idea of using the MinMax problem solutions as the guideline for sonar allocation problem. Hence the basic strategic sonar allocation model becomes the model of equations (4.8) through ((4.11), where the matrix A is the payoff matrix as is described earlier.

However, this game theoretic model is pretty simple and there are issues with it; e.g. the sonar placement problem is a discrete resource allocation problem (the number of sonars is integer), while this model provides probabilities as the decision of defender. We need to modify the model or its solutions in order to be able to use its outputs. Another issue is that the current model lets each sonar to cover just one cell (the cell where it is placed). To be able to use this model for real world scenarios these simplifying assumptions and the others that are mentioned earlier in this Chapter are required to be removed. We deal with removing these assumptions in the next Chapter. Here a small example is presented to show the operation of the basic model.

Consider a grid with six cells. The configuration of this grid and the cell characteristic values are given in Figure 4.7. Assume that each sonar can cover one cell, the cell where it is placed. The detection probability for the cell that the sonar is located at is *dp*=0.7.

1	2	3	4	5	6
13	27	7	20	28	27

Figure 4.7. A six cell grid with characteristic values

With this information we can populate the payoff matrix for the attacker and the defender. Since the game is assumed to be zero-sum, we just present the attacker payoff values in the matrix as shown in Figure 4.8. The numbers are rounded to the nearest integer for simplicity. The defender payoffs can be easily calculated by multiplying attacker's payoff by -1. Solving the MinMax linear program of the defender with this payoff matrix results in Y = (0, 0.322, 0, 0, 0.356, 0.322). The objective function value becomes 20.88. This is the expected loss to the defender in the NE of the game. It can be interpreted as: allocate 32.2% of your resources to cell 2, 35.6% to cell 5 and 32.2% to cell 6. As mentioned earlier, this output cannot be used for sonar placement immediately and it requires modifications.

		Defender					
		1	2	3	4	5	6
Attacker	1	4	13	13	13	13	13
	2	27	8	27	27	27	27
	3	7	7	2	7	7	7
	4	20	20	20	6	20	20
	5	28	28	28	28	8	28
	6	27	27	27	27	27	8

Figure 4.8. Payoff matrix of the six-cell grid

These results are in accordance with what we expected from the model. To minimize the maximum loss, the defender chooses the cells with highest characteristic values to allocate his resources to. In this case his expected maximum loss is v=20.88 which is the value of the objective function of equation (4.8) for this example. Note that this value makes the inequality of equation (4.9) binding (active) for i=2,5,6.

4.6. Conclusion

A background on game theory is provided in this chapter, before proposing a game theoretic modeling methodology for sonar allocation in ports and waterways against terrorist attacks in Chapter 5. A few preliminary concepts and definitions in game theory are discussed to make the notations and definitions more clear. These definitions help to understand some technical phrases in the literature review section. In literature survey, the focus is on the articles that study the security risk problem with a game theoretic perspective.

Then the concept of NE in games is introduced. It is a powerful solution concept for games and is widely used in literature articles as well. Since finding the NE for games with large payoff matrices is not easy, the duality theory for zero-sum games is discussed next. Duality theory results in two dual linear programs, the solutions of which are the mixed strategy NE for the game. These two concepts help to find the solution of the game once we define it.

To simplify the modeling methodology, a few assumptions have been made. These assumptions are removed in the next Chapter to obtain the general game theoretic model that is comparable with probabilistic model in sense of the details both models include.

The elements of the game such as who the players are, what are possible actions for them and what is the payoff of each action are introduced later. Providing this information, the dual linear programs are constructed based on the duality theory. The defender's linear program is of interest for us. It is used as a guideline for the defender to allocate his resources. He assigns his resources to the cells proportional to the mixed strategy probabilities that he obtains from solving his linear program.

5. Extensions of Strategic Risk Model

5.1. Introduction

The game theory model for the sonar allocation problem is developed in Chapter 4. This model included several simplifying assumptions; e.g., multi-cell coverage of a sonar was relaxed and each cell was allowed just to cover the cell that it is placed at, range-dependent detection probability was ignored and all sections of the cell were protected with the same detection probability, the sonar allocation problem is assumed to be continuous (non-discrete) meaning that any proportion of total the resource can be assigned to a cell and just one type of sonar was allowed in the model.

In this Chapter we seek to relax these assumptions such that our model can be used in real world problems with minimal approximations and simplifications. Since multi-cell coverage of sonars and range-dependent detection probability are closely related, these two assumptions are added at the same time just by modifying the payoff matrix. Then the linear program of the defender is converted to a binary linear program so that it is possible to deal with the discrete nature of the sonar allocation problem.

Next, allowance of multiple sonar types is integrated into the model. It lets the model to choose not only where to place the sonars but also which types of sonar to use. To allow multiple sonar types in the model both the payoff matrix and the defender's binary program need to be modified. The size of payoff matrix becomes larger from $c \cdot c$ to $c \cdot k \cdot c$, where c is the number of cells and k is the number of sonar types to be allowed in the model. The decision variable in the binary program is redefined to allow multiple sonar types and the constraints of the problem also change.

With the scope of generality that we reach up to here, it is possible to build game theoretic models that are comparable to the probabilistic risk model of Chapter 2; that is, both models have similar features such as range-dependent detection probability, multi-cell coverage of sonars and they both allow multiple sonar types to be used. Accordingly, we plan to run the game theoretic

model on the test cases that are developed for the probabilistic model in Chapter 4 such that the results of the two approaches are comparable.

The last step to generalize our model is to allow the payoff matrix of the proposed game to be general-sum. Considering the zero-sum property for the payoff matrix lets us use the duality theory for zero-sum games as a modeling technique and solution mythology to reach Nash Equilibrium. However, this assumption decreases the generality of our model to such an extent that it cannot be used in most real world scenarios. In real world problems the payoff for defender and attacker may come from different sources and therefore do not necessarily add up to zero. Hence assuming a general-sum payoff matrix for our game model makes it more realistic and applicable. Clearly the duality theory cannot help as a modeling approach here. Instead, other solution techniques that can solve the general-sum games can be used from the literature.

This final step will be applied to the basic model of Chapter 4, for which none of the generalization steps of this Chapter are not effective. Hence, the generalization steps (removing simplifying assumptions) will be applied to the general-sum game model again. This step involves more work than the other generalization steps and it will be discussed in a new Chapter and is left for future work.

5.2. Multi-cell Coverage by Sonars

Single-cell coverage is one of the shortcomings of the basic model presented in Chapter 4. It is quite necessary for the sonars to be able to keep multiple cells under coverage in our model. Otherwise the size of cells should be selected so large that each sonar can cover one cell and a great amount of accuracy is going to be lost due to large cell size. Fortunately, it is possible to modify the payoff matrix and include the multi-cell coverage case of a sonar into the model. Also, we can bring the range dependency of detection probability to the payoff matrix at the same time. To explain this argument more clearly, consider the grid illustrated in Figure 5.1 on right. If placing a sonar in cell 1 leads to the coverage of an adjacent cell such as cell 2, then the payoff values of cell 2 for both attacker and defender need to be changed. Moreover, the detection probability for cell 2 is lower than cell 1; That is instead of using exactly the same detection probability (dp) for the adjacent cell (cell 2), it is possible to use a smaller detection probability ($dp^* < dp$).

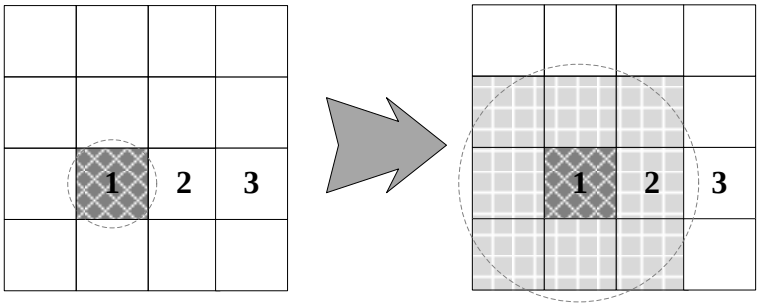


Figure 5.1. Two grids, one with a sonar that covers one cell (left) and the other with a sonar that covers nine cells (right)

To reproduce the payoff matrix with the new settings, assume that each sonar covers nine cells as shown in Figure 5.1. The cell that includes the sonar is covered with probability dp (dark highlight) and the surrounding cells (light highlight) with dp^*. Also consider cells 1, 2 and 3 to be located as numbered in Figure 5.1. Then the payoff matrix for these three cells can be rewritten as

the lower matrix in Figure 5.2. Let us use p_{ij} as the attacker's payoff if he chooses action i and the defender chooses action j. Since placing sonars in one cell still covers the same cell with highest detection probability, the p_{ij} values on the main diagonal (where $i=j$) of the payoff matrix do not change. To calculate p_{21} we need to think about the situation where the defender has allocated a sonar in cell 1, while the attacker chooses cell 2 to invade. Since the sonar covers the adjacent cells as well, it covers cell 2, too (as shown in Figure 5.1). However the detection probability at cell 2 is dp^* in this case. Thus the attack will be successful with probability $1\text{-}dp^*$ and the expected damage that the attacker will lead to is $a_2 \cdot (1\text{-}dp^*)$. In the same way we need to update the payoff values for all the cells that fall inside the coverage of sonars to obtain the lower payoff matrix in Figure 5.2. Since the game is zero-sum, the defender payoffs are of course $-p_{ij}$.

Defender

		1	2	3
Attacker	1	$a_1(1\text{-}dp), -a_1(1\text{-}dp)$	$a_1, -a_1$	$a_1, -a_1$
	2	$a_2, -a_2$	$a_2(1\text{-}dp), -a_2(1\text{-}dp)$	$a_2, -a_2$
	3	$a_3, -a_3$	$a_3, -a_3$	$a_3(1\text{-}dp), -a_3(1\text{-}dp)$

Defender

		1	2	3
Attacker	1	$a_1(1\text{-}dp), -a_1(1\text{-}dp)$	$a_1(1\text{-}dp^*), -a_1(1\text{-}dp^*)$	$a_1, -a_1$
	2	$a_2(1\text{-}dp^*), -a_2(1\text{-}dp^*)$	$a_2(1\text{-}dp), -a_2(1\text{-}dp)$	$a_2(1\text{-}dp^*), -a_2(1\text{-}dp^*)$
	3	$a_3, -a_3$	$a_3(1\text{-}dp^*), -a_3(1\text{-}dp^*)$	$a_3(1\text{-}dp), -a_3(1\text{-}dp)$

Figure 5.2. Simple (upper matrix) and modified (lower matrix) payoff matrix of the proposed game

Then the dual optimization problems for this payoff matrix can be rewritten and solved to achieve the optimal solutions. However the optimal probabilities for the defender are still continuous and cannot be directly used for sonar allocation. Hence, the next step is to discretize the defender solutions so that the results can lead to the sonar allocation.

5.3. Discrete Nature of Sonar Allocation Problem

As mentioned earlier, the sonar allocation problem has a discrete nature, meaning that it is not possible to assign a proportion of a sonar to a cell. This issue does not let us use the defender probabilities for sonar allocation immediately. A discretization process is required to make the probabilities useful. Since one sonar type is assumed to be used (for now), it is possible to find the number of available sonars. The value is $d = [b/s]$, where b is the budget, s is the sonar price and [ ] returns the integer part of the argument). Then it may seem reasonable to solve the linear MinMax problem and find the d highest probabilities and allocate sonars to the corresponding cells (since each probability is assigned to a cell). However, there are two issues with this heuristic method. First, the number of positive probabilities may be less than the number of sonars and we will have problems allocating the surplus sonars (as each cell can allocate at most one sonar to it in our model). The bigger problem is that the value of positive probabilities does not necessarily express any priority in selecting the corresponding cells for sonar allocation; that is sorting the positive probabilities decreasingly and selecting the d first ones does not guarantee the optimal solution. To elaborate more on these issues, consider the following example.

Think of a grid with six cells the same as the one in Chapter 4. The configuration of this grid and the cell characteristic values are given in Figure 5.3. Also assume that the sonar type to be used for coverage can cover three cells (the cell that it is located at and the left-side and right-side cells

in case they exist). The detection probability for the cell that the sonar is located at is *dp*=0.7 and for the two other cells is *dp*=0.4.

1	2	3	4	5	6
13	27	7	20	28	27

Figure 5.3. A six cell grid with characteristic values

The payoff matrix for this game can be constructed as shown in Figure 5.4. The numbers are rounded to the nearest integer for simplicity (these are the payoff values for the attacker). If we solve the MinMax problem, the optimal solution is: Y = (0, 0.432, 0, 0, 0.326, 0.242) with the objective value of 18.806 (note that the objective value gets better than the example of Chapter 4 due to multi-cell coverage of sonars). This means that the defender should allocate 43% of his resources to cell two, 33% to cell 5 and 24% to cell 6. Now if we decide to allocate four sonars to this grid, we will have problem where to put the fourth one. Now assume that our budget is limited and we can just put one sonar. Based on the above approach, since cell 2 has the highest probability, we have to allocate the sonar to cell 2. Since the highest characteristic value belongs to cell 5, we expect the sonar to be allocated to either of cells 4, 5 or 6 so that cell 5 is covered. Hence this solution methodology is not working as expected and we need to look for a more robust approach.

		Defender					
		1	2	3	4	5	6
Attacker	1	4	8	13	13	13	13
	2	16	8	16	27	27	27
	3	7	4	2	4	7	7
	4	20	20	12	6	12	20
	5	28	28	28	17	8	17
	6	27	27	27	27	16	8

Figure 5.4. Payoff matrix of the six-cell grid

Our goal is to make the solution of the MinMax problem compatible with the discrete nature of our resource allocation problem. This can be achieved through discretization of y_j probabilities; that is, we need to force y_j's to be either zero or *1/d*. In that case, equation (5.1) guarantees that *d* number of y_j probabilities will get the value of *1/d*. Then *d* sonars can be allocated to the corresponding cells.

$$\sum_{j=1}^{c} y_j = 1 \tag{5.1}$$

To make this argument more formal, a slight modification can be done to the model. We can multiply both sides of equation (5.1) with *d*. The right hand side becomes *d*. Taking the *d* on the left hand side into the summation results in

$$\sum_{j=1}^{c} d \cdot y_j = d \tag{5.2}$$

Since y_j is either zero or *1/d*, the term $d \cdot y_j$ is either zero or one. Hence a new binary variable can be defined as $z_j = d \cdot y_j$ and be used instead of y_j in the formulation. As d is a fixed positive number, multiplying it by the objective function will have no effect on the optimization model. To clean up the model, the term $d \cdot V$ is replaced by $V^{'}$. These changes convert our model into a standard form of a binary linear program as given in equations (5.3) through (5.6).

$$\min V^{'} \tag{5.3}$$

s.t.:

$$(AZ^T)_i \leq V^{'} \quad \forall i \tag{5.4}$$

$$\sum_{j=1}^{n} z_j = d \tag{5.5}$$

$$z_j \in \{0,1\} \quad \forall j \tag{5.6}$$

Solving the binary program for the six cell example with d=3 yields Z = (0, 1, 0, 0, 1, 1). This solution is the same as what we obtained from the heuristic. Putting d=4 results in Z = (1, 1, 0, 0, 1, 1) and d=1 gives Z = (0, 0, 0, 0, 0, 1). The two latter cases show the advantage of the binary program compared to the heuristic. In next section we will see that using the binary program model is the only choice for the case where multiple sonar types are allowed to be used.

5.4. Multiple Sonar Types

Using a single sonar type limits the capabilities of the resource allocation model to a great extent. Though in real world scenarios the firms may often choose a single resource (sonar) brand for allocation to reduce total cost, every brand of sonar have multiple types of sonars that differ in

many specifications. Hence it is necessary to allow multiple sonar types to be used in our methodology and let the model choose which types to use and how many of each. In this section we seek to remove the single sonar type assumption from our model so that it becomes a more realistic methodology that can be used real world problems.

In case of single sonar type, the binary variable z_j specifies if a sonar is placed at cell j or not. However, this information is not enough when multiple sonar types can be used. We need to know what type of sonar is assigned to cell j if any. To deploy this information z_j is replaced by a new variable z_{jk}, which is one when a sonar type k is placed at cell j and zero otherwise. Since the new variable has an additional index, all the equations of the binary program require modification.

First, we start with equation (5.5) that determines the number of sonars. Since different sonar types have various specifications, it is reasonable to assume a distinct price for each type. Then it will not be possible to calculate the number of sonars beforehand even though the budget is known. Instead we need to define a budget constraint that reflect the sonar prices and the budget directly; that is

$$\sum_{k=1}^{q}(s_k \cdot \sum_{j=1}^{c} z_{jk}) \leq b \tag{5.7}$$

Where s_k is the price of sonar type k and q is the number of sonar types. However, for the specific case of our problem $a_{ij} \geq 0$, $\forall i,j$ (as shown in the payoff matrix of Figure 5.2 for the attacker) and since the model of equations (5.3) through (5.6) is a minimization problem, the model decides to put $z_{jk}=0$ $\forall j,k$. Hence equation (5.7) needs to be modified. Changing the "less than or equal" sign to equality is not practical. Since z_{jk}'s are binary, this equation forces the model to find the combination of sonar prices that are exactly equal to the budget and narrows the feasible set to a

great extent (which is not desirable. Instead, one can change the sign to "greater than or equal to" and reduce a small value ε>0 from the right hand side. That is

$$b - \varepsilon \leq \sum_{k=1}^{q} (s_k \cdot \sum_{j=1}^{c} z_{jk}) \leq b \tag{5.8}$$

The choice of ε depends on the price of different sonar types. The maximum value of ε can be the highest sonar price s_{max}. If $\varepsilon=s_{max}$, then it is possible to add a sonar (with any price) to the solution and still the left hand side of equation (5.8) be less than or equal to b. One can run the model for a number of values of $\varepsilon\in[0,s_{max})$ starting from zero and stop as soon as the value of the left hand side of inequality (5.8) becomes less than or equal to b.

The next step is to modify the payoff matrix. Since the defender is the only player that deals with the sonar placement problem, only his actions set changes due to multiple sonar types (the attacker's set of actions does not change). The actions set of the defender for the single sonar type is defined as $Z=(z_1,z_2,\ldots,z_c)$, where $\sum_{j=1}^{c} z_j = d$, $z_j \in \{0,1\}$ where d is the number of sonars and z_j is the binary variable that specifies whether the defender chooses cell j to protect or not. Letting multiple sonar types in the model, the defender needs to choose among the sonar types as well; that is he needs to choose the cells that he wants to protect and also select the type of sonars to place in those cells. Hence the idea of the two index variable z_{jk} can be helpful here. The actions set can be redefined as $Z=(z_{11},z_{12},\ldots,z_{1c},z_{21},z_{22},\ldots,z_{2c},\ldots,z_{q1},z_{q2},\ldots,z_{qc})$ where $z_{jk} \in \{0,1\}$ and z_{jk} is defined as mentioned earlier. Note that inequality (5.8) limits the number of positive z_{jk}'s. It is obvious that the number of possible actions for the defender increases from c to $d\cdot c$ and the size of the payoff matrix becomes $c\cdot d\cdot c$. To illustrate the payoff matrix let us get back to the three-cell example for which the payoff matrix for single sonar type is already presented. Suppose that two sonar types are allowed in the model. The first type is exactly the same as the one in the previous example. It covers three cells; the cell that it is located at with detection probability dp, plus the

cells on its left and right with detection probability dp^*. The second type can cover two cells; the cell that it is located at with dp' and the cell on its right with dp'^* ($dp'^* \leq dp'$). Also assume that the cells labeled as cells 1, 2 and 3 are located as shown previously in Figure 5.1.

Defender

Attacker	1	2	3
1	$-a_1(1-dp)$	$-a_1(1-dp^*)$	$-a_1$
2	$-a_2(1-dp^*)$	$-a_2(1-dp)$	$-a_2(1-dp^*)$
3	$-a_3$	$-a_3(1-dp^*)$	$-a_3(1-dp)$

Payoff matrix for sonar type 1

Defender

Attacker	1	2	3
1	$-a_1(1-dp')$	$-a_1(1-dp'^*)$	$-a_1$
2	$-a_2$	$-a_2(1-dp')$	$-a_2(1-dp'^*)$
3	$-a_3$	$-a_3$	$-a_3(1-dp'^*)$

Payoff matrix for sonar type 2

Defender

Attacker	1	2	3	4	5	6
1	$-a_1(1-dp)$	$-a_1(1-dp^*)$	$-a_1$	$-a_1(1-dp')$	$-a_1(1-dp'^*)$	$-a_1$
2	$-a_2(1-dp^*)$	$-a_2(1-dp)$	$-a_2(1-dp^*)$	$-a_2$	$-a_2(1-dp')$	$-a_2(1-dp'^*)$
3	$-a_3$	$-a_3(1-dp^*)$	$-a_3(1-dp)$	$-a_3$	$-a_3$	$-a_3(1-dp'^*)$

Payoff sub-matrix for sonar type 1 | Payoff sub-matrix for sonar type 2

Figure 5.5. The payoff matrix for the three-cell example with a single sonar type (on top) and two sonar types (at the bottom)

The payoff matrices for the models with a single sonar type are presented in the top section of Figure 5.5. Note that the payoff values are just presented for the defender. Since the game is zero-sum, the attacker's payoffs can be calculated by removing the minus sign from the defender's payoffs. The top left matrix is for the game with sonar type 1 and it is exactly the same as the

bottom matrix in Figure 5.2. The top right is for the game that just allows sonar type 2. Since sonar type 2 just covers two cells, the payoff matrix is a bit different from the matrix on left. It is easy to check how these modifications are done.

The bottom payoff matrix is for the model that allows both sonar types to be used. It is clear that that the first three columns of the bottom payoff matrix are exactly duplicated from the top-left payoff matrix and the last three columns are the duplicates of the top-right matrix. In the same way the payoff matrix can be modified for as many sonar types that are allowed in the model. This payoff matrix is now conformable with the actions set or binary decision variable $Z=(z_{11},z_{12},\ldots,z_{1c},z_{21},z_{22},\ldots,z_{2c},\ldots,z_{d1},z_{d2},\ldots,z_{dc})$ that is introduced earlier. In other words, the number of columns of the payoff matrix A and the number of elements of vector Z are equal and while calculating AZ^T the columns of A are multiplied with the correct elements of vector Z. Hence the left side of inequality (5.4) is still legitimate.

We need to introduce a new set of constraints to conclude the integration of multiple sonar types into the model. A basic assumption in our approach is that we allow each cell to allocate at most one sonar to it. However with our current formulation up to q sonars can be assigned to a cell. To avoid this event we need to limit the number of sonars in each cell with a new constraint as:

$$\sum_{k=1}^{q} z_{jk} \leq 1 \quad \forall j \tag{5.9}$$

This means that at most one sonar type can be placed at location j. Putting these discussion points together yield our new binary linear program. This program can be written as:

$$\min V' \tag{5.10}$$

s.t.:

$$(AZ^T)_i \leq V' \quad \forall i \tag{5.11}$$

$$b - \varepsilon \leq \sum_{k=1}^{q} (s_k \cdot \sum_{j=1}^{c} z_{jk}) \leq b \tag{5.12}$$

$$\sum_{k=1}^{q} z_{jk} \leq 1 \quad \forall j \tag{5.13}$$

$$z_{jk} \in \{0,1\} \quad \forall j, k \tag{5.14}$$

This model allows multiple sonar types to be used in it. The level of generality of this model is almost equivalent to the full probabilistic model of equations (2.8) through (2.14) in Chapter 2. The only difference is that the strategic risk model does not consider multiple coverage of sonars over a cell. This minor shortcoming is not of great importance in real case scenarios. Due to budget limitations that apply in such situations, the density of sonars is low and the case in which multiple sonars cover the same cell becomes scarce. Hence there is no need to worry about this defect.

In future work, the test examples of Chapter 3 will be replicated for the strategic game model of equations (5.10) through (5.14) so that it becomes possible to compare the performance of both models. Moreover the zero-sum assumption for the payoff matrix will be removed and a more general case of the game called the general-sum game will be presented.

5.5. Comparison of SRM and PRM

This section seeks to exercise the strategic game model that is developed in this chapter. First, the game model will be tested with the test cases that were used for the probabilistic model in chapter 3. In the second step, the NE solutions to the MaxMin problem (attacker's problem) are used to build attack scenarios for the previous test cases. Then based on the sonar placement schemes

from both SRM and PRM, the expected damage of these attacks will be calculated and compared for the two approaches.

The test cases that are going to be used in this section are the same as the ones used for exercising the PRM. Hence, the two approaches can be contrasted and their performance can be compared. In order to make the comparison more reasonable, the final SRM of equations (5.9) to (5.14) in Chapter 5 will be used for modeling the test cases. This model is the most detailed game model and is closest to the PRM in terms of the assumptions that they make (these assumptions are described in Chapters 3 and 4).

The test case for which the results are shown in Table 3.2 is the one that is used for comparison. In summary, it includes five similar scenarios each having 300 cells (as shown in Figure 3.6). The results of these scenarios are averaged to reach more robust results. The same types of sonars (with 360˚, 180˚ and 90˚ coverages) are going to be utilized. The budget starts from zero (no sonar is placed) and increases in steps of 50 until it reaches 400. Going beyond 400 will result in long runtimes for both cases. Moreover, more than half of the cells (which are the ones with high a_{ij} values) are already covered by this budget. Increasing the budget will not cause much risk reduction for values above 400 for this test case.

For consistency, the Solver results from PRM will be compared with the Solver results from SRM (the heuristic results are not used). The 3% optimality gap is enforced in both cases so that the results can be reached in a reasonable time.

One measure of interest in comparing the two methods is the summation of risk over all cells after the sonar placement (objective function for PRM). This measure was used in Chapter 3 as well. However, here it is used with a slight change. The summation of risk over all cells is replaced by average risk per cell (via dividing the sum by the number of cells). It provides a better understanding for the range of characteristic values (a_{ij}) and conforms to the next performance measure (that is defined after this measure). The comparison of average risk per cell

is illustrated for SRM and PRM in Figure 5.6 for various budgets. As expected, in both cases, the average risk per cell decreases when budget increases. The PRM performs slightly better than SRM in decreasing average risk. This is due to the fact that the average risk per cell appears in the objective function for PRM while it is not the main focus in SRM.

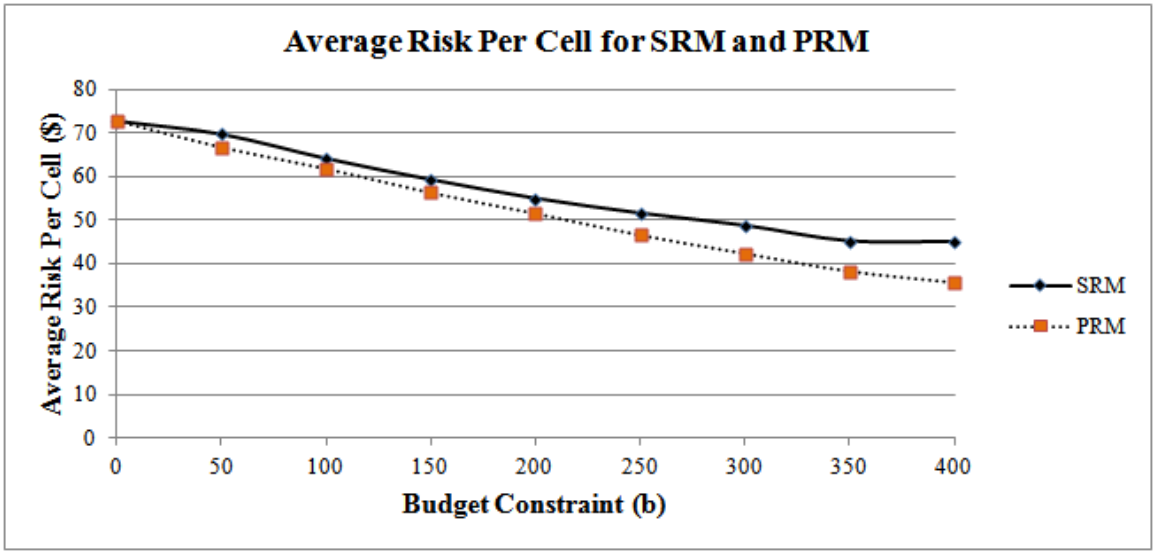


Figure 5.6. Comparison of average risk per cell for SRM and PRM

The solver results for this test case are provided in Table 5.1. The measures in the second and third row are the ones for the average risk per cell and the following two rows are for another performance measure that is discussed next.

Table 5.1. Solver results on Average risk per cell and maximum risk over all cells for SRM and PRM (for the test case with 300 cells)

Measure	**Model**	**Budget**								
		0	50	100	150	200	250	300	350	400
Average Risk per Cell	PRM	72.9	66.8	61.9	56.3	51.5	46.6	42.3	38.3	35.7
	SRM	72.9	69.8	64.2	59.4	55.0	51.6	48.7	45.1	45.0
Maximum Risk over all Cells	PRM	136	136	133.8	130	128	128	126.5	126	126
	SRM	136	129.2	126.8	123	117.8	115.6	114.4	113	112

In Chapter 5, the MinMax approach was utilized to reach the NE of the SRM. As suggested by the name, the MinMax approach looks for the highest payoffs of the attacker and provides strategies for the defender such that these payoffs (attacker's) are minimized. In essence, it minimizes the maximum payoffs of the attacker. Looking at payoff matrices of the proposed game models of Chapters 4 and 5 reveals that the highest attacker payoffs are related to the cells with largest a_{ij} values. That is, the MinMax approach finds the cells with highest a_{ij}'s and decreases the attacker's payoff by assigning sonars to these cells or in their vicinity. This is equivalent to say that the MinMax model minimizes the maximum a_{ij} value overall cells.

This argument leads to the next measure that is of interest for the comparison of the models. As mentioned earlier, the "average risk per cell" measure is the objective function in the PRM and hence favors it. Based on the last discussion we can use the maximum risk over all cells as a measure that has a close relationship with the SRM objective and compare the performance of the two models according to this measure. The results for this measure are provided in the last two rows of Table 5.1 and illustrated in Figure 5.7. Clearly, the SRM accomplishes better results for this measure.

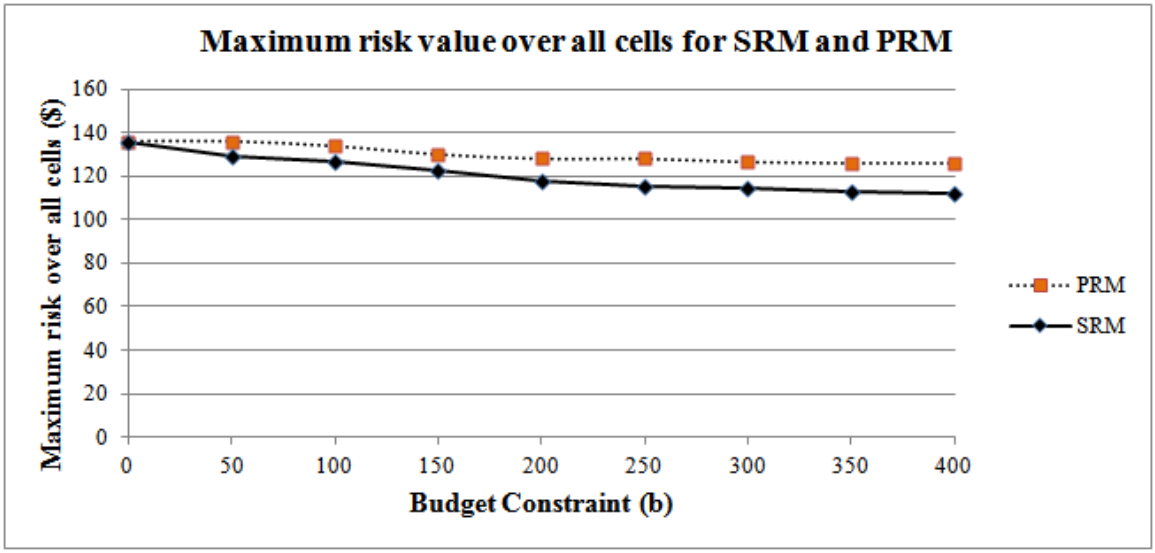


Figure 5.7. Comparison of maximum risk over all cells for SRM and PRM

To clarify the workings of PRM and SRM for these two performance measures, let us present the following example. Consider the cases provided in Figure 5.8 below. Assume that we seek to place a single sonar (that covers 4 cells as shown in Figure 5.8, with dp = 0.9) in a grid of cells that includes both case 1 and case 2. The numbers shown in the cells are the characteristic values and the empty cells are the ones with negligible a_{ij} values. The PRM places the sonar in the location that is shown in Case 1, due to the fact that Case 1 yields a higher risk reduction over all cells (0.9×[80+90+95+85]=315) compared to Case 2 (0.9×[65+70+75+110]=288). If we choose SRM to place the sonar it allocates the sonar to the point that is given in Case 2 as it covers the cell with the highest characteristic value (the cell with a_{ij} = 110). Thus, allocating the sonar to the point in Case 1 leads to a lower average risk per cell but higher maximum risk over all cells compared to the point in case 2. This is the rationale for the behavior of the two models against each of the average risk per cell and maximum risk over all cells measures.

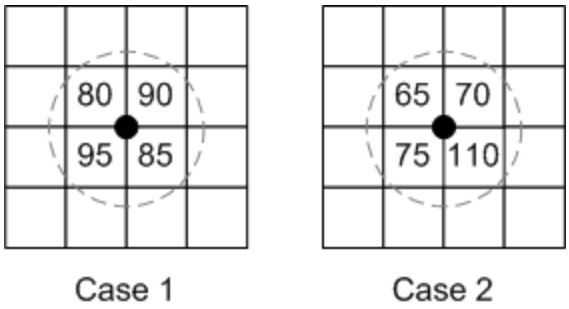


Figure 5.8. An example to compare SRM and PRM

These results do not convey a great advantage for any of the two models. PRM achieves slightly better results for the average risk per cell and SRM leads to somewhat lower maximum risk over all cells. Based on the interest of the modeler, one can benefit from the features that each of these models provide. The PRM is able to accommodate more features of the soars while the SRM can integrate the attacker's intelligence into the modeling effort.

A more interesting and realistic performance measure can be defined by considering the level of damage in case an attack happens. That is, we can generate attack scenarios and calculate the expected damage to the environment for various sonar placement schemes developed by PRM and SRM. To achieve the expected damage (as a performance measure) we need to specify how to generate attack scenarios and how to calculate the expectation.

As described in Section 4.5 of Chapter 4, two optimization models were developed as a result of the game theory approach; a MinMax model that leads to the NE strategies of the defender (and serves as a guideline for sonar placement) and a MaxMin model that solves the game for the attacker and provides his NE strategies. The MaxMin model and its results were not of interest in Chapter 4. However, they can be used here to generate attack scenarios. In other words, the outcome of solving the MaxMin model are the mixed strategy attack probabilities each corresponding to a cell of the environment. These probabilities are used by the intelligent attacker to choose potential regions of the environment among all cells for invasion.

To calculate the expected damage for a sonar placement scheme, the expected damage is conditioned on the cell to be attacked as given in equation (5.15).

$$E(Damage) = \sum_{i=1}^{number\ of\ cells} E(damage|cell\ i\ is\ attacked) \cdot p(cell\ i\ is\ attacked) \quad (5.15)$$

and then equation (5.16) is used to calculate the conditional expectation.

$$E(damage|cell\ i\ is\ attacked) = \begin{cases} a_i & if\ cell\ i\ is\ not\ covered\ by\ a\ sonar \\ a_i \cdot (1 - dp_n) & if\ cell\ i\ is\ covered\ by\ coverage\ of\ type\ n \end{cases} \quad (5.16)$$

Finally, the probability that cell i is attacked comes from the NE probabilities for the MaxMin model. In order to make this idea more clear, it is applied to a simple example. Consider the six-cell grid that was provided as an example in section 5.3 of Chapter 5. The arrangement of cells and their characteristic values are shown in Figure 5.9.

1	2	3	4	5	6
13	27	7	20	28	27

Figure 5.9. Arrangement of the six-cell grid

The sonar type that is used for this example is assumed to cover 3 cells, the cell that accommodates it and the ones to its left and right. The detection probability for the cell that the sonar is located at is *dp*=0.7 and for the two other cells is *dp*=0.4. These specifications lead to the payoff matrix (for attacker) shown in Figure 5.10 (and described in Chapter 5).

		Defender					
		1	2	3	4	5	6
Attacker	1	4	8	13	13	13	13
	2	16	8	16	27	27	27
	3	7	4	2	4	7	7
	4	20	20	12	6	12	20
	5	28	28	28	17	8	17
	6	27	27	27	27	16	8

Figure 5.10. Payoff matrix of the six-cell grid

Table 5.2 shows the results for solving the defender binary program for this example with different number of sonars (*d*).

Table 5.2. Six-cell example results

Number of Sonars (*d*)	**Placement Scheme (*Z*)**
1	(0, 0, 0, 0, 0, 1)
2	(0, 1, 0, 0, 0, 1)
3	(0, 1, 0, 0, 1, 1)
4	(1, 1, 0, 0, 1, 1)

It is possible to solve the linear program for the attacker at the same time. This linear program is the same as the one that was developed in equations (4.3) through (4.6) in Chapter 4. The results for this linear program are: X = (0, 0.445, 0, 0, 0.261, 0.294). Each x_i value represents the probability that cell is attacked by the attacker in NE. These probabilities can be used in the calculation of expected damage. Now we can calculate the expected damage for each of the sonar

placement schemes in Table 5.2. Before placing any sonar (*d*=0), the expected damage is calculated as:

$$E(Damage) = 13 \times 0 + 27 \times 0.445 + 7 \times 0 + 20 \times 0 + 28 \times 0.261 + 27 \times 0.294) = 27.261.$$

When *d*=1, a single sonar is placed in cell 6 that covers cell 6 with dp=0.7 and cell 5 with dp=0.4. The rest of cells are not covered and the expected damage becomes:

$$E(Damage) = 13 \times 0 + 27 \times 0.445 + 7 \times 0 + 20 \times 0 + 28 \times 0.261 \times (1 - 0.4) + 27 \times 0.294 \times (1 - 0.7) = 18.781$$

For *d*=2 the expected damage decreases to 10.370. All the cells that are going to be attacked (based on NE) are covered up to this point. Hence, adding extra sensors to the grid leads to multiple coverage and yields minimal expected damage reduction. As can be seen via this example, the expected damage is a metric that integrates the attack scenarios into account and can be a more comprehensive measure for model comparisons. Hence a new comparison of SRM and PRM can be performed utilizing the expected damage as follows.

Again, the same test case is going to be used with 300 cells as shown in Figure 3.6 and the budget constraint is relaxed step by step to compare the performance of the two models. As the results of Figure 5.11 suggest, the SRM outperforms the PRM. In fact, the SRM takes the intelligence of attacker into account and places the sonars accordingly, while the PRM seeks to minimize the overall risk in the field, regardless of where the attacker may choose to attack. This result reveals the advantage of SRM over PRM when the adversary is intelligent (which is the case in this study).

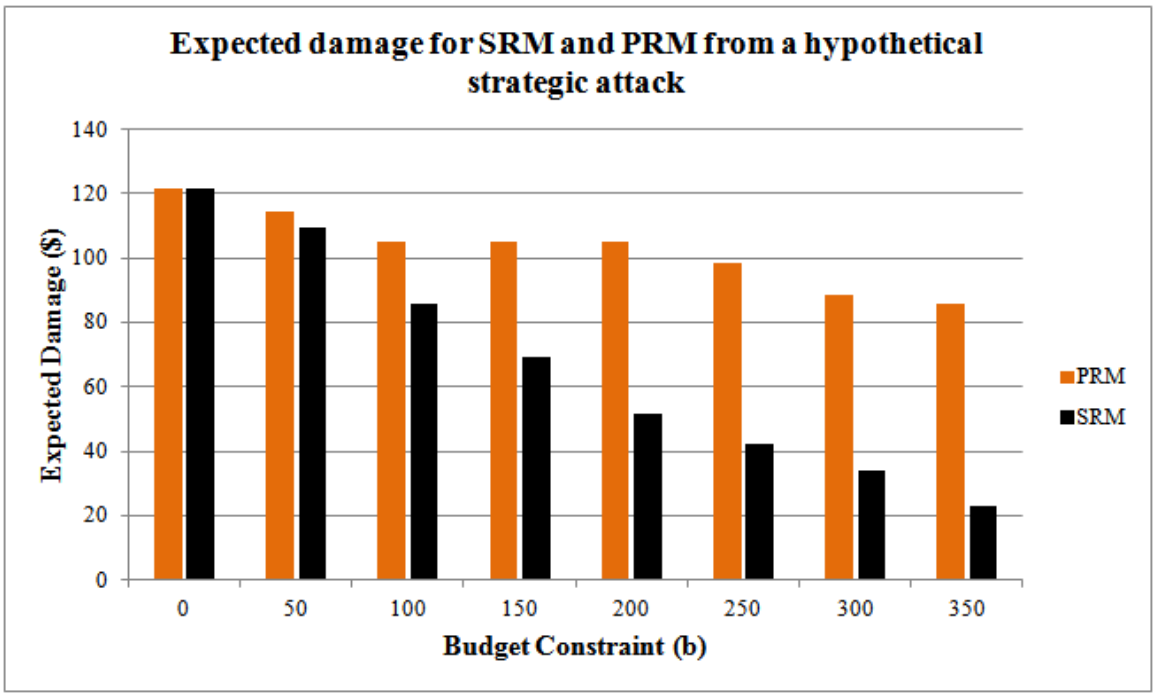


Figure 5.11. Comparison of expected damage for SRM and PRM

It is also interesting to compare SRM and PRM in a field with roughly different configuration. That is, we can use the New York Harbor example that was introduced in Chapter 3. Comparing Figure 3.2 and Figure 3.6 show how different these cases are from various perspectives (geography, distribution of a_{ij}'s and etc.).

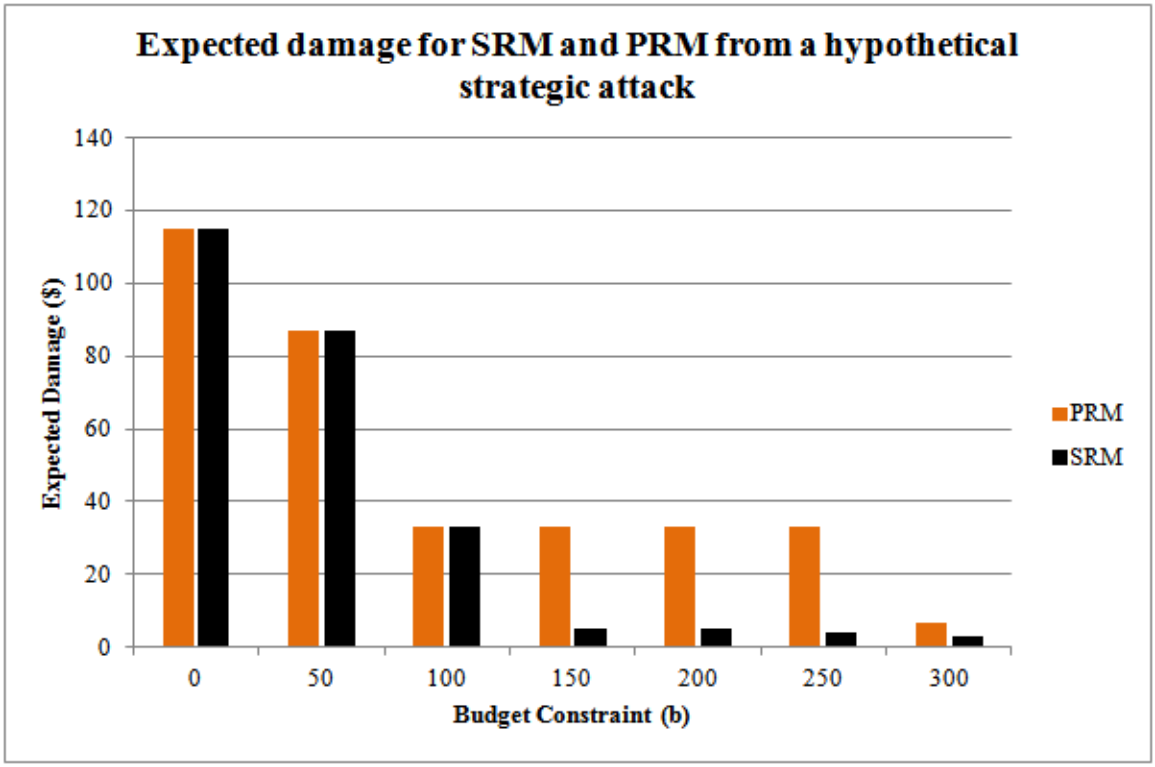


Figure 5.12. Comparison of expected damage for SRM and PRM in the New York Harbor example

Although the results of Figure 5.12 look a bit different from the ones in Figure 5.11, they both share the same results; the SRM suites the context of security risk analysis better than the PRM. For low budget values both models place the sonars around the dark cells in the top right section of Figure 3.6 and they almost decrease the expected damage to the same extent. However, when the budget exceeds 100, both models start to place the extra sonars in other parts of the environment and from this point the SRM provides better results than PRM. Finally, when the sonars become abundant in the field (budget constraint becomes more relaxed), most of the critical cells will be covered and consequently PRM's performance gets closer to SRM's.

The MinMax approach that we achieved while building the game theoretic framework has been widely used in the literature of risk analysis recently. In risk analysis, one of the significant

challenges is to assess the risk of various situations and find the ones peaking the risk. These peaks are then interpreted as the risk indicators. Then the goal is to investigate possible methods to mitigate the risk of such situations either through decreasing the probability of occurrence of such events or decreasing the consequences of them. In essence, the objective is to minimize the maximum risk values.

In the context of our study, the MinMax model does exactly the same thing. It finds the cells with maximum risk indicators and mitigates the risk by placing the sonar in the proximity of these cells. Hence, the SRM fits the literature of security risk analysis better.

Table 5.3 provides a short comparison of the SRM and PRM.

Table 5.3. Comparison of SRM and PRM

SRM	**PRM**
Integrates geographical details of the environment	Integrates geographical details of the environment
Allows multi-cell coverage of sonars	Allows multi-cell coverage of sonars
Features range-dependent detection probability	Features range-dependent detection probability
Allows multiple sonar types	Allows multiple sonar types
Does not allow multiple detection of sonars over a cell	Allows multiple detection of sonars over a cell
Minimizes the maximum risk over all cells	Minimizes the maximum risk over all cells
Considers intelligence of attackers	Ignores intelligence of attackers

5.6. Conclusion

The fundamental ideas for developing the SRM were presented in Chapter 4. Based on this introduction, the simple version of the strategic risk model is proposed in this chapter. This model is too simple to be used in large scale real world problems. Hence, it needs to involve more realistic assumptions. We focused our efforts on removing a number of simplifying assumptions

that were made in the beginning of this chapter. That is, we sought to include multi-cell coverage of sonars, range-dependent detection probability, the discrete nature of sonar allocation problem and allowance of multiple sonar types in the model.

Multi-cell coverage of sonars is added to the model just by modifying the payoff matrix. From previous Chapters we know that the detection probability of sonars decreases as the distance from the sonar increases. Hence, when a sonar covers multiple sonars, the cells which are farther from the sonar should receive less coverage (lower detection probability). We take this effect into consideration while modifying the payoff matrix and therefore add the range-dependent detection probability to the model at the same time.

Considering the discrete sonar allocation problem leads to the change in the nature of the defender's MinMax model. That is, we need to introduce a new binary decision variable instead of the continuous decision variable. This makes the model a binary linear program.

It required more effort to integrate multiple sonar types into the model. In this step, the model and the payoff matrix are required to be modified both. A new index is added to the decision variable to indicate sonar type. The constraint that defines the number of sonars is replaced by a budget constraint. A new constraint is added that limits the number of sonars per cell to at most one. Finally the payoff matrix becomes larger. In other words, for any new type of sonar, an extra $c \cdot c$ (c is the number of cells) sub-matrix is added to the old payoff matrix.

Removing these assumptions make the model more real and allow it to be used as a game theoretic modeling technique for sonar placement in ports and waterways in real world problems. This model now includes the same level of details as the probabilistic risk model of Chapter 2 does. Hence, we tried to provide a comparison between these models (SRM and PRM) using the test cases that were developed in Chapter 3. Three measures were proposed to quantify the performance of the models; namely average risk per cell, maximum risk over all cells and expected damage. The first and the second measure are the objective functions in SRM and PRM,

respectively. They just take the defender's view into consideration, while the last measure calculates the expected damage for the defender regarding the attacker's decisions. Hence, it is more reliable and more interesting.

None of the models showed a considerable advantage over the other for the first two measures. However, the SRM outperforms the PRM for the last measure. This shows the advantage of the SRM (over PRM) due to integration of attacker's intelligence in the model.

6. General-sum Game

6.1. Introduction

Two modeling approaches have been used to develop mathematical models for sonar placement in ports and waterways in Chapters 2 through 5. The probabilistic model of Chapter 2 is exercised in Chapter 3 with a number of test cases. In Chapter 4 an introduction to game theory is provided to provide the preliminaries for the SRM model. Then the simplest case of this game theoretic model is presented. This model is then generalized in Chapter 5 and the final version of it is compared with the PRM of Chapters 2 and 3.

One of the simplifying assumptions for the strategic game model in Chapter 5 was the zero-sum property of the payoff matrix for the game. This property allows the duality theory to be used for modeling the game. Based on duality theory, the zero-sum game becomes equivalent to a set of dual linear programs (one for the attacker and one for the defender). Solving each of the programs results in the NE strategies for the corresponding player. Since we are interested in the defender's strategy, we just solve his linear program and use the solution as a direction for sonar placement.

Despite having such nice properties, the zero-sum assumption is not realistic in many applications. In practice, the defender's and the attacker's interests may be entirely different. This means that the harm or damage to one of them is not necessarily the benefit of the other party and vice versa. A simple example of such a case is shown in the payoff matrix of Figure 4.5. As a result the sum of payoffs for various situations will not become zero. The final step to generalize the game theoretic model would be to make it general-sum. It means that the duality theory cannot be used anymore. Instead, other modeling and solution techniques are surveyed in the literature for the solution methodologies for finding NE for general-sum games. Moreover, since the modeling approach changes, the linear program of the defender will change as well and all the steps that we used to remove the simplifying assumptions (in Chapter 5) are needed again.

In this chapter, the literature of solution techniques for general-sum games is discussed. Then one of these techniques that suits the context of our study is utilized for modeling the attacker-defender game. Once the basic model is developed, it is modified (as the SRM was generalized in Chapter 5) in order to fit in the context of sonar placement problem. At the end, a test example is generated and run by the final version of general-sum SRM to illustrate the performance of this model.

6.2. Literature Review

Since Nash (1951) proposed the existance of NE for any game with finite set of actions, lots of efforts have been dedicated to finding efficient techniques to calculate the NE for different types of games. Since games are defined in variuos settings and with different assumptions, most solution techniques are proposed for a specific type of game and they work best for that specific type. Two player games are not excluded form this fact. However, since it is among the less complex type of games, the two player games have been studied to a great extent in the literature.

Various solution techniques are developed for this type of game. The most famous algorithm is proposed by Lemke and Howson (1964). They provide a constructive proof that a two player game has one equilibrium (at least). Since the proof is constructive, it provides a method for finding the equibrium. A number of other studies modify the Lemke-Howson algorithm to improve the performance for special cases. A survey of such studies is provided by Stengel (2002). Another group of researchers use Theorem 1 proposed by Nash (1951) and seek to find optimization techniques that satidfy the conditions of this theorem to reach the NE.

<u>Theorem 1 (Nash 1951)</u>:

Consider a two player general-sum game as:

$A_{m\times n}$: payoff matrix for player 1(attacker)

$B_{m\times n}$: payoff matrix for player 2 (defender)

x_i : NE probabilities for player 1 (i=1,2,…, m)

y_j : NE probabilities for player 2 (j=1,2,…,n)

Then for such a game has an NE $X = (x_1, x_2, \ldots, x_m)$ and $Y = (y_1, y_2, \ldots, y_n)$ if and only if for suitable values of U_{max} and V_{max} equations (6.1) through (6.8) hold.

$$\sum_{i=1}^{m} x_i = 1 \tag{6.1}$$

$$\sum_{j=1}^{n} y_j = 1 \tag{6.2}$$

$$U_{max} - \sum_{j=1}^{n} y_j \cdot A_{ij} \geq 0 \qquad \forall i \tag{6.3}$$

$$V_{max} - \sum_{i=1}^{m} x_i \cdot B_{ij} \geq 0 \qquad \forall j \tag{6.4}$$

$$x_i \cdot \left(U_{max} - \sum_{j=1}^{n} y_j \cdot A_{ij} \right) = 0 \qquad \forall i \tag{6.5}$$

$$y_j \cdot \left(V_{max} - \sum_{i=1}^{m} x_i \cdot B_{ij} \right) = 0 \qquad \forall j \tag{6.6}$$

$$x_i \geq 0 \qquad \forall i \tag{6.7}$$

$$y_j \geq 0 \qquad \forall j \tag{6.8}$$

These conditions define the Linear Complementarity Problem as defined by Cottle and Dantzig (1968). According to Cottle et al. (1992), there are various solution techniques for solving LCP. Sandholm et al. (2005) propose a Mixed Integer Linear Program (MILP) to solve the system of equation (6.1) through (6.8) . Since this model can accommodate the objective and features of our sonar placement problem, it is used as the baseline for our modeling effort as it will be described next.

6.3. Proposed Risk Analysis Model in the Context of Game Theory

As mentioned earlier, an MILP proposed by Sandholm et al. (2005) will be used as a foundation to build and solve our general-sum game. The feasible solutions of this MILP are the equilibriums of the game (there might be more than one NE for this game according to Lemke and Howson (1964)). Once the proposed model is described, it will be modified to adapt to the nature of our problem.

Basically, this MILP forces equations (6.3) through (6.6) to hold by defining new binary variables z_i and w_j as follows.

$$z_i = \begin{cases} 0 & if\ x_i > 0 \\ 1 & if\ x_i = 0 \end{cases} \tag{6.9}$$

$$w_j = \begin{cases} 0 & if\ y_j > 0 \\ 1 & if\ y_j = 0 \end{cases} \tag{6.10}$$

These variables are used in the following constraints to satisfy Complementarity.

$$x_i \leq 1 - z_i \qquad \forall i \tag{6.11}$$

$$U_{max} - \sum_{j=1}^{n} y_j \cdot A_{ij} \leq M_1 \cdot z_i \qquad \forall i \tag{6.12}$$

$$y_j \leq 1 - w_j \qquad \forall j \tag{6.13}$$

$$V_{max} - \sum_{i=1}^{m} x_i \cdot B_{ij} \leq M_2 \cdot w_j \qquad \forall j \tag{6.14}$$

Where M_1 and M_2 could be any large number, but to have a good formulation they are defined as:

$$M_1 = \max_{i,j} A_{ij} - \min_{i,j} A_{ij} \tag{6.15}$$

and

$$M_2 = \max_{i,j} B_{ij} - \min_{i,j} B_{ij} \tag{6.16}$$

M_1 and M_2 are input parameters to the model and they are calculated based on the payoff matrices A and B. Replacing equations (6.3) through (6.6) by the equations (6.11) through (6.14) converts the problem into a MILP. The model that we reach with these modifications is the one that is proposed by Sandholm et al. (2005). The rest of the development procedure in this chapter is the contribution of this research to the literature (unless stated otherwise).

For any specific NE of the game the corresponding values of U_{max} and V_{max} have contextual meanings here. For a solution, U_{max} is the expected payoff for player 1 and V_{max} is the expected payoffs for player 2.

Clearly, this MILP does not have an objective function. It is possible to solve it and reach the feasible solutions which are the NE of the game. However, in order to decrease the solution time and reach desirable solutins, it is strongly suggested by Sandholm et al. (2005) to define an objective function for the MILP. The objective function is arbitrary. It can be any measure of interest that can guide the decision variables (x_i's and y_j's) to a specific section of the feasible region. A suitable objective function for this study is to minimize the expected payoff for the player 1 (attacker) which is U_{max}. Hence our MILP becomes:

$$Min\ U_{max} \tag{6.17}$$

s.t.:

$$\sum_{i=1}^{m} x_i = 1 \tag{6.18}$$

$$\sum_{j=1}^{n} y_j = 1 \tag{6.19}$$

$$x_i \leq 1 - z_i \qquad \forall i \tag{6.20}$$

$$U_{max} - \sum_{j=1}^{n} y_j \cdot A_{ij} \leq M_1 \cdot z_i \qquad \forall i \tag{6.21}$$

$$y_j \leq 1 - w_j \qquad \forall j \tag{6.22}$$

$$V_{max} - \sum_{i=1}^{m} x_i \cdot B_{ij} \leq M_2 \cdot w_j \qquad \forall j \tag{6.23}$$

$$x_i \geq 0 \qquad \forall i \tag{6.24}$$

$$y_j \geq 0 \qquad \forall j \tag{6.25}$$

$$z_i \in \{0,1\} \qquad \forall i \tag{6.26}$$

$$w_j \in \{0,1\} \qquad \forall j \tag{6.27}$$

Equation (6.17) seeks to minimize the expected payoff for the attacker. Equations (6.18), (6.19), (6.24) and (6.25) assure that x_i's and y_j's are legitimate mixed strategy probabilities. Complementarity is satisfied by equations (6.20) through (6.23) as described before. Finally, equations (6.26) and (6.27) assert that z_i's and w_j's are binary. The solution to this model provides the NE of the game for which the expected payoff for the attacker is minimized. However, the solutions will not conform to the nature of our problem. The y_j probabilities are continuous and cannot be used as a guideline for sonar placement as described in Chapter 5. Hence a discretization process is required. We use the approach that was developed in Chapter 5.

6.4. Discretization Process Implementation

For now let us assume that just one type of sonar is allowed in the model. Hence, using d= [b/s] (where b is the budget and s is the sonar price) we can specify the number of sonars (d). Then, in order to be able to use the defender decisions (y_j's) as a guideline for sonar placement, y_j probabilities should be enforced to be either 0 or 1/d.

$$y_j \in \{0, \frac{1}{d}\} \tag{6.28}$$

Then we can define a new variable y_j' such that

$$y_j' = d \cdot y_j \tag{6.29}$$

and

$$y_j' \in \{0,1\} \tag{6.30}$$

Where y_j' is one if a sonar is placed at cell j and zero otherwise. Based on this modification, equations (6.18), (6.21) and (6.22) change to (6.31), (6.32), and (6.33), respectively.

$$\sum_{j=1}^{n} y_j^{'} = d \tag{6.31}$$

$$d \cdot U_{max} - \sum_{j=1}^{n} y_j^{'} \cdot A_{ij} \leq d \cdot M_1 \cdot z_i \qquad \forall i \tag{6.32}$$

$$y_j^{'} \leq 1 - w_j \qquad \forall j \tag{6.33}$$

This concludes the discretization process of the model. Now we can allow multiple sonar types in the model and modify the equations accordingly. Similar to what was done in Chapter 5, we will allow multiple sonar types in the model. Assume that q types of sonars are allowed to be used and each sonar of type k costs s_k. First, the variable y_j' is replaced by y_{j_k}' (note that k is an index for j or in other words the index j is replaced by j_k). y_{j_k}' is one if a sonar of type k is placed at cell j and zero otherwise.

Since it is not possible to specify the number of sonars anymore, we need to replace equation (6.31) with a budget constraint. That is

$$\sum_{k=1}^{q} (s_k \cdot \sum_{j=1}^{n} y_{j_k}^{'}) \leq b \tag{6.34}$$

But with the same rationale that we provided for equation (5.8), it needs to be converted to

$$b - \varepsilon \leq \sum_{k=1}^{q} (s_k \cdot \sum_{j=1}^{n} y_{j_k}^{'}) \leq b \tag{6.35}$$

Next, the payoff matrix needs to be modified. Since this process is exactly the same as what we did in Chapter 5, it is not going to be repeated here. Note that this process changes the size of the payoff matrices A and B from $m \times n$ to $m \times (n \cdot q)$ and the decision variable of the defender $Y^{'}$ becomes $Y' = (y_{1_1}', y_{2_1}', \dots, y_{n_1}', y_{1_2}', y_{2_2}', \dots, y_{n_2}', \dots, y_{1_q}', y_{2_q}', \dots, y_{n_q}')$. The parameters m and n are

the number of rows and columns of the payoff matrices, respectively. Since the number of cells in our study is *c*, then both *m* and *n* can be replaced by *c* in all equations. However, to maintain the generality of the model, *m* and *n* are kept in the formulations. Thus the final formulation becomes:

$$Min\ U_{max} \tag{6.36}$$

s.t.:

$$\sum_{i=1}^{m} x_i = 1 \tag{6.37}$$

$$b - \varepsilon \leq \sum_{k=1}^{q} (s_k \cdot \sum_{j=1}^{n} y'_{j_k}) \leq b \tag{6.38}$$

$$x_i \leq 1 - z_i \qquad \forall i \tag{6.39}$$

$$q_{max} \cdot U_{max} - \sum_{j=1}^{n} \sum_{k=1}^{q} y'_{j_k} \cdot A_{ij_k} \leq q_{max} \cdot M_1 \cdot z_i \qquad \forall i \tag{6.40}$$

$$y'_{j_k} \leq 1 - w_{j_k} \qquad \forall j, k \tag{6.41}$$

$$V_{max} - \sum_{i=1}^{m} x_i \cdot B_{ij_k} \leq M_2 \cdot w_{j_k} \qquad \forall j, k \tag{6.42}$$

$$\sum_{k=1}^{q} y'_{j_k} \leq 1 \qquad \forall j \tag{6.43}$$

$$x_i \geq 0 \qquad \forall i \tag{6.44}$$

$$y'_{j_k}, w_{j_k} \in \{0,1\} \qquad \forall j, k \tag{6.45}$$

$$z_i \in \{0,1\} \qquad \forall i \tag{6.46}$$

Equation (6.40) is the modified version of equation (6.32). Since the number of sonars is not specified in case of allowing multiple sonar types, the variable *d* in equation (6.32) becomes useless. Instead, we can use any large number in the right-hand-side of the equation, but to have a good formulation it is better to use the smallest possible value. Hence we define a new variable

q_{max} and use it instead of d. This variable is defined as $q_{max} = [b/s_{min}]$ (where b is the budget and s_{min} is the price for the cheapest sonar type) which is the maximum number of sonars allowed in the model. As a result of this modification, U_{max} is not the exact expected payoff for the attacker, but it intuitively expresses the same measure. Due to the addition of index k and the change in payoff matrix, equation (6.23) converts to (6.42). Equation (6.43) assures that no more than one sonar is placed in any cell and the rest of equations are the same as they were defined before.

This formulation concludes the modeling effort for building the most general SRM model. This model allows

1. a sonar to cover multiple cells with range-dependent detection probability
2. multiple type of sonars in the model
3. any payoff matrix (not necessarily zero-sum).

In the next section this model will be exercised to show its performance.

6.5. Test cases for General-sum Game

The first payoff matrix that we developed in Chapter 5 (Figure 4.5) led to a non-zero sum game. Assuming attacker's cost of initiating an attack (ca) and the defender's cost of sonar placement (cd) to be negligible, the game resulted to be zero-sum. However, these costs are not necessarily insignificant in many cases. In fact, they may change the players' decisions.

In the following test case we seek to check whether these costs can modify the players' decisions or not. To achieve this goal, we will use the same test case that was developed in Chapter 3 and used in Chapter 5. This test case is a grid of c = 300 cells (10 rows and 30 columns). In the test case, we assume that the cost of sonar placement for the defender is the same (zero) for all the cells of the environment , and just use different invasion costs (for the attacker) for the cells of the grid. To check the effect of different invasion costs on the player's decision variables, the

invasion cost for cells is defined in three settings. Invasion costs are assumed to be zero for all cells in the first setting. This results to the zero-sum game that we have studied in Chapter 5. In the second and third setting, we assume that attacking the cells with high characteristic values is more difficult (due to the special preventive actions that the defender may incorporate). In the third setting, the level of difficulty is such that it inverts the payoff for the attacker for all cells. Let us use a small example to make this argument clear. Figure 6.1 shows a grid of three cells with their characteristic values.

1	2	3
80	100	90

Figure 6.1. The six-cell grid

Let us assume that just one sonar that covers a single cell (with *dp* = 0.6) is going to be used for this example. The payoff matrix for the defender (Figure 6.2) is calculated exactly in the same way as Chapter 5.

	1	2	3
1	-32	-80	-80
2	-100	-40	-100
3	-90	-90	-36

Figure 6.2. Payoff Matrix for the defender for the three-cell example (the payoffs for the attacker are the same values but with positive sign for the zero-sum case)

Now assume that initiating an attack to cell 1 costs \$5 ($c_{a,1} = \$5$) for the attacker and $c_{a,2} = \$30$ and $c_{a,3} = \$18$. Using the formula shown in Figure 4.5 for calculating the payoffs for the attacker we get the matrix of Figure 6.3.

	1	2	3
1	27	75	75
2	70	10	70
3	72	72	18

Figure 6.3. Payoff Matrix for the attacker for the three-cell example (setting 3)

As seen in Figure 6.3, the attacker's preference in choosing a cell to invade is inverse the zero-sum case; meaning that the cells with higher characteristic values has become less interesting for the attacker (the order of preference for the attacker in the zero-sum case is cell 2, cell 3 and cell 1 from highest to lowest).

Now let us explain the second setting. This is a setting in between settings 1 and 3; e.g. presume that initiating an attack to cell 1 costs \$5 ($c_{a,1} = \$5$) for the attacker and $c_{a,2} = \$10$ and $c_{a,3} = \$25$. Figure 6.4 shows the new payoff matrix for the attacker.

	1	2	3
1	27	75	75
2	90	30	90
3	65	65	11

Figure 6.4. Payoff Matrix for the attacker for the three-cell example (setting 2)

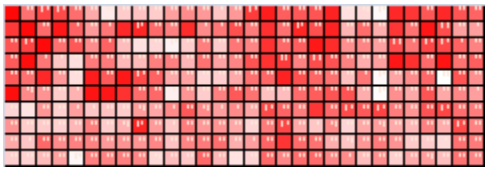
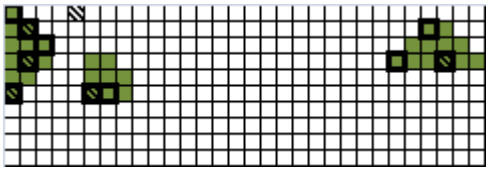
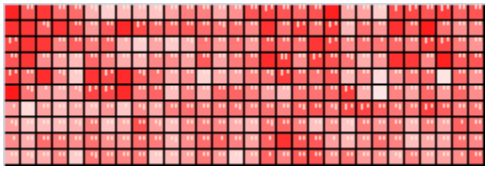
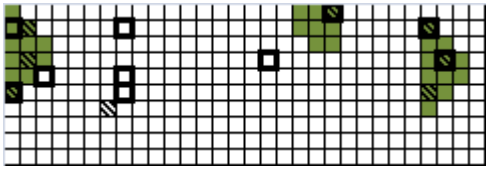
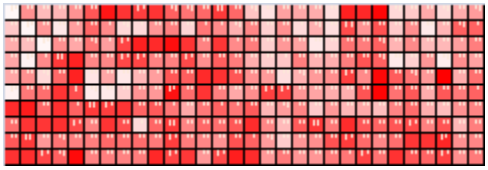
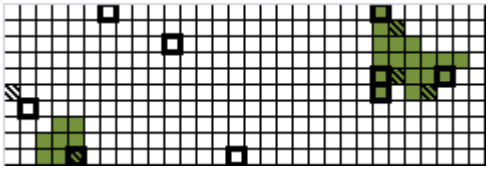
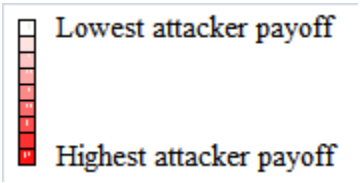
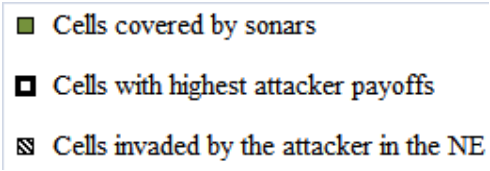
Setting 2 is an intermediate state in the sense that the preference for the attacker is the same as the zero-sum game in some cases and inverted in other situations (cell 2 has still the highest preference, but cell 1 and cell 3 has inverted their priorities).

The test case with 300 cells will be used for running these three settings for the formulation of equations (6.36) through (6.46). The settings described above cover a wide range of games and can show how sensitive the results of game can be to various payoff matrices. Since we are interested in the placement of sonars, we run the settings on a single scenario (rather than building 5 scenarios and averaging over scenarios) and compare the placement among settings. Moreover, the budget is fixed to $100 (which is almost equivalent to placing three sonars in the environment).

Table 6.1 illustrates the results for the three settings. The left panel in this table shows the heat map of the attacker's payoff for the three settings. The defenders payoff (which is the negative of attacker's payoff for setting 1) is the same for all three settings. Although setting 1 is zero-sum, it can be solved using the general-sum model as shown in Table 6.1. Note that results are not guaranteed to be optimal; however they are within 3% gap of optimality. In all three settings the sonars are placed such that they cover the cells with highest attacker payoffs. Since these 10 cells are located close to each other in setting 1, they are all covered. While in settings 2 and 3 these cells are distributed in the environment and hence cannot be covered by three sonars.

Since the objective function of our general-sum game seeks to minimize the attacker's expected payoff, most invaded cells are covered in all settings. It is also important to note that there might be other NE's for these settings; however, these ones are the ones that minimize the attacker's payoff.

Table 6.1. Comparison of sonar placement schemes for the three settings (the colored cells in left panel are the cells covered by sonars, the cells with thick borders are 10 cells with highest payoff for the attacker and the hatched cells are the cells that are invaded by the attacker)

	Heat map of the attacker's payoff	**Sonar placement scheme**
Original port		
Original port with noise		
Mirror image port		
Legend		

Comparing the sonar placement schemes in these settings reveals a notable fact. Although the defender's payoffs are the same for all the settings, the allocation of sonars in settings 1 and 3 is completely different. Hence, the attacker's interest is an important factor in the results of the game. Linking the heat maps (or cells with highest attacker payoffs) and sonar placement schemes in Table 6.1 indicates the sensitivity of the placement scheme to attacker's interest. This example clarifies how different the results of a zero-sum game and nonzero-sum game can be. Hence, modeling the SRM as a general-sum game provides more degrees of freedom in defining the payoffs and consequently provides more reliable and accurate results.

The significance of attacker's concern in driving the sonar placement scheme also magnifies the advantage of SRM over PRM. In PRM, it is not possible to include attacker's decisions into the modeling effort.

6.6. Conclusion

To wrap up the modeling approach, the last simplification assumption is removed from SRM in this chapter. That is, the game theory model is assumed to be general-sum rather than zero-sum. This allows the model to be used in a variety of general cases where the attacker and defender's points of interest will not necessarily coincide.

Since the game is not zero-sum, the duality theory becomes ineffective in this context. Instead, a number of other solution methodologies (that are developed in the literature) are utilized to model the general-sum game. Among all, a mixed integer linear program that can accommodate the features of our sonar placement problem is selected to be used. After introducing this model, it is modified to allow the discreteness of the sonar placement problem and also multiple sonar types.

The final model has the same features as the zero-sum game. It is then tested using an example to show the flexibility of general-sum game in modeling any payoff matrix for the players. This example also emphasizes the advantage of SRM over PRM in considering attacker's prospect in the modeling effort (by showing the notable change in sonar placement scheme as a result of variation in attacker's interest).

7. Conclusions and Future Directions

7.1. Summary

Two solution methodologies are proposed in this research study to mitigate maritime security risks. These models provide sonar placement schemes under the water (in ports and waterways) to detect anomalies such as divers, torpedoes and autonomous underwater vehicles. Both models integrate details about the geography of the environment, physics of sonars and etc. to a great extent. A number of test cases are developed and used though the study in order to text the proposed models and compare them.

The description of the main question in this study and its significance to a nation is followed by the introduction about the nature of the problem and its limitations in Chapter 1. Then the contribution of this manuscript to the literature of maritime security risk is explained. A few notes that clarify the scope of work in the study are presented at the end of this chapter.

The literature for the PRM is reviewed in the beginning of the second chapter. Introducing the discretization process allows us to get into more details of the model. Then the mathematical model for PRM is proposed in two steps. First, a simple version of the model is presented to show the underlying idea of the approach. Then the comprehensive optimization model follows. This model is revised to fit into a mixed integer linear program. A number of preprocessing steps are described afterwards. These steps explain the assumptions of our model and provide guidelines for preparing the parameters of the model.

The New York harbor test example is introduced in Chapter 3. The workings of the model are presented in this example. It also remarks the computational complexity of the model. This model is NP hard and cannot be solved for optimality using the solver packages in a reasonable amount of time (for large scale problems). A heuristic solution methodology is proposed to overcome this obstacle. This heuristic is greedy and iterative. It selects a sonar and places it in a cell of the grid that leads to maximum risk reduction in each iteration until the budget is exhausted. A few notes

about the complexity of the heuristic method are also discussed. The results from a competitive commercial solver package are compared to the ones from the heuristic for a number of test cases. These results illustrate the performance of the heuristic in various situations and make recommendations when to use heuristic. This concludes the development of the PRM and its solution methodologies.

The focus of research converts to the SRM form Chapter 4. A number of elements in game theory that are used later are defined and discussed in the beginning of this chapter. Then a literature review on the application of game theory in security risk is provided. The concept of Nash Equilibrium and its importance in game theory is presented in the next section. Since the SRM models in Chapters 4 and 5 are zero-sum, they can benefit from the link between duality theory and zero-sum games. Hence, the duality theory for zero-sum games is also argued. Setting a number of simplifying assumptions, allows us to present the simple version of the SRM for sonar placement. This model is then tested on a small test case.

The focus of Chapter 5 is on removing the simplifying assumptions of Chapter 4 and converting the SRM into reasonable model for practical purposes. That is, to integrate multi-cell coverage of sonars, range-dependent detection probability of sonars, discrete nature of sonar placement problem and multiple sonar types in the SRM. These extensions are brought into SRM step by step and some modifications are made into the model so that it can accommodate these features.

Reaching the final version of zero-sum SRM (which covers almost the same features as PRM) enables us to compare these models. Three measures are introduced to compare the performance of these models. These measures are compared for the test cases that were developed in Chapter 3. The SRM provides better results than the PRM considering the overall performance of these models for the three measures.

The final step in extending the SRM of Chapters 4 and 5 is to allow the payoff matrices to be general-sum (rather than zero-sum). This brings a great deal of flexibility to SRM as described in

Chapter 6. This chapter begins with a theorem from Nash (1951) that provides the conditions for reaching NE's of a general-sum game. Various methods are used in the literature in order to satisfy these conditions for any game and reach the NE's. Among all, a mixed integer linear program approach is selected. This model is able to integrate the features of the sonar placement problem into it. We extend this model step by step to integrate these features (as we did in Chapter5). Finally, the performance of the final general-sum SRM is tested via a test case. This test case illustrates the significance of attacker's intelligence and interest in the final solution of the game.

7.2. Research Contributions

This study contributes to the literature of maritime security risk analysis and management via proposing two modeling methodologies for underwater surveillance at ports and waterways against security threats. These models aim to find the optimal placement of sonars under the water such that the most critical sections of the port fall under surveillance of sonars. A budget constraint limits the number of sonars to be used. Both models integrate environment-specific features of the problem to a great extent.

The probabilistic risk model resembles the facility location problem, wireless sensor networks optimization or other general purpose sensor allocation problems to some extent. However, our probabilistic model incorporates many details that are specific to underwater sonar allocation, such as range-dependent detection probability, multiple coverage of sonars over a cell and angular coverage of sonars. The greedy heuristic that provides good solutions in linear time complexity is a great achievement in the study. Also, the test examples that are developed in Chapters 3, 5 and 6 are designed to test the models from various points of view. These test cases can be used in the literature as reference test cases for checking the performance of similar models.

To the best of our knowledge, the strategic risk model is the only game theoretic study for sonar allocation in ports and waterways in the literature. A similar game theoretic approach is proposed in the literature for sonar allocation in oceans. Yet, the assumptions of this study make it unusable for sonar allocation in ports. Moreover, the proposed model is just comparable to the basic game model of Chapter 4. Our SRM is quite comprehensive in the sense that it covers geographical features of the environment and physics of sonar place. Besides, it is presented in two versions; namely zero-sum and general-sum games. The zero-sum game is a special case that may not happen frequently in application. However, its solution methodology requires less effort.

The performance measures to compare the performance of PRM and SRM are among the novel contributions of this study. Specially, the last measure, expected damage, takes the attacker's action at NE into account is a very powerful measure as a result.

On the other hand, the general-sum game does not have the limitations of the zero-sum game. It is able to accommodate any payoff matrices for the players. The final general-sum SRM integrates exactly the same features as the zero-sum one and it is shown how powerful it is to capture the attacker's interests.

Also, preprocessing steps for the resource allocation problem, such as the discretization process and evaluation of relative significance of grid points are discussed in this manuscript. Guidelines for specifying these parameters are also provided. Most literature studies do not provide insight how to prepare model parameters for their studies.

7.3. Research Conclusions

A probabilistic risk model is developed in Chapter 2 for sonar allocation against security threats in ports and waterways. This model integrates the environmental and sonar-specific features to a great extent. The model is a mixed integer linear program and is proved to be NP hard in the literature. The parameters that add to the complexity of the model and their effect are discussed in

Chapter 3. It is shown that it is not time-feasible to solve this model for optimality even for medium size problems.

In order to use the model for real world large scale problems a heuristic algorithm is developed. The computational complexity of the heuristic is discussed as well. A number of test examples have been used to compare the performance of the heuristic with a solver package. These examples show the effect of parameter changes on the runtime and accuracy of results. The outcomes show that the results of the heuristic are so close to optimal when the density of sonars is low in the environment (due to low budget). This observation makes the heuristic useful for the real world scenarios. In reality the budget is usually low for the operator and he seeks to minimize the risk with his limited budgets. Hence, the density of sonars in the field of interest is low and the heuristic results are expected to be close to optimal.

A game theoretic model is developed in Chapter 4. This model is quite simple and does not feature much details of the sonar allocation problem. This model is then extended in Chapter 5 through a number of steps. These steps make the model more general. The generalized model is equivalent to the probabilistic model from the level of features the both integrate. Hence it is reasonable to compare the performance of these models. This task is done via introducing three performance measures; namely average risk of all cells, maximum risk over all cells and expected damage. The expected damage has an advantage over the other measures. It considers the attacker's action while doing the calculation and consequently is of more interest. It turns out that none of the models show any significant advantage over the other for the first two performance measures. However, the SRM performs much better than the SRM for the last measure.

The general-sum SRM of Chapter 6 clarifies the significance of attacker's concern in the sonar placement problem. Using this model, we showed that the sonar placement scheme is dragged towards the attacker's points of interest. Hence SRM has a great advantage over PRM in this regard. Moreover, the examples show the flexibility of general-sum SRM in capturing any pattern

for the payoff matrices, while in the zero-sum case the payoff values for attacker and defender should add up to zero. Consequently, it seems that the General-sum SRM is the most appropriate method to model the sonar placement problem of this study.

References

Aitsaadi, N., N. Achirt, K. Boussettat, and G. Pujolle. 2007. "Differentiated Underwater Sensor Network Deployment." *OCEANS* 1-6.

Aspnes, J., D. Goldenberg, and R. Y. Yang 2004. "On the Computational Complexity of Sensor Network Localization." Pp. 32-44 in *Algorithmic Aspects of Wireless Sensor Networks*. Springer Berlin / Heidelberg.

Bar-Noy, A., T. Brown, and S. Shamoun. 2010. "Sensor Allocation in Diverse Environments." *Distributed Computing in Sensor Systems* 6:379-392.

Barry, E. 2006. "Lost in the Dust of 9/11." Los Angeles Times. (http://articles.latimes.com/2006/oct/14/nation/na-cleaners14).

Berry, J. W., L. Fleischer, W. E. Hart, C. A. Phillips, and J. P. Watson. 2005. "Sensor Placement in Municipal Water Networks." *Journal of Water Resources Planning and Management* 131(3):237-243.

Bier, V. M., L. A. Cox, and M. N. Azaiez 2009. "Why Both Game Theory and Reliability Theory Are Important in Defending Infrastructure against Intelligent Attacks." in *Game Theoretic Risk Analysis of Security Threats*. Springer.

Bier, V. M., L. Samuelson, and S. Oliveros. 2007. "Choosing What to Protect: Strategic Defensive Allocation against an Unknown Attacker." *Journal of Public Economic Theory* 9(4):563-587.

Bram, J., J. Orr, and C. Rapaport. 2002. "Measuring the effects of the September 11 attack on New York City." Federal Reserve Bank of New York.

Brown, G. G., W. M. Carlyle, J. Salmeron, and K. Wood. 2005. "Analyzing the Vulnerability of Critical Infrastructure to Attack and Planning Defenses." *Tutorials in Operations Research, INFORMS* 102-123.

Brown, G. G., W. M. Carlyle, J. Smeron, and K. Wood. 2006. "Defending Critical Infrastructure." *Interfaces* 36(6):530-544.

Cavalier, T.M., W. A. Conner, E. Del Castillo, and S. I. Brown. 2007. "A heuristic algorithm for minimax sensor location in the plane." *European Journal of Operational Research* 183:42-55.

Chakrabarty, K., S. S. Iyengar, H. Qi, and E. Cho. 2002. "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks." *IEEE Transactions on Computers* 51(2):1448-1453.

Clouqueur, T., V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja. 2002. "Sensor Deployment Strategy for Target Detection." *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* 42-48.

Consultants, Planning a. M. 2002. "National Dredging Needs Study of U.S. Ports and Harbors.".

Cottle, R. W. and G. B. Dantzig. 1968. "Complementary Pivot Theory of Mathematical Programming." *Linear Algebra and its Applications* 1:103-125.

Cottle, R. W., J. S. Pang, and R. E. Stone. 1992. *The Linear Complementarity Problem*. San Diego: Academic Press.

Cox, L. A. 2008. "Some Limitations of "Risk = Threat × Vulnerability × Consequence" for Risk Analysis of Terrorist Attacks." *Risk Analysis* 28(6):1749-1761.

Dantzig, G. B. 1957. "Discrete-Variable Extremum Problems." *Operations Research* 5:266-288.

Dhillon, S. S. and K., I. S. S. Chakrabarty. 2002. "Sensor Placement for Grid Coverage under Imprecise Detections." *Proceedings of the Fifth International Conference on Information Fusion* 2:1581-1587.

DHS. 2011. "Department of Home land Security, Critical Infrastructures." Department of Homeland Security. (http://www.dhs.gov/files/programs/gc_1189168948944.shtm).

Dimitrov, N. B., D. P. Michalopoulos, D. P. Morton, M. V. Nehme, F. Pan, E. Popova, E. A. Schneider, and G. G. Thoreson. 2009. "Network Deployment of Radiation Detectors with Physics-Based Detection Probability Calculations." *Annals of Operations Research*.

Ezell, B. C., S. P. Bennett, D. Winterfeldt, J. Sokolowski, and A. J. Collins. 2010. "Probabilistic Risk Analysis and Terrorism Risk." *Risk Analysis* 30(4):575-589.

GAMS. 2012. "GAMS Homepage." http://www.gams.com/. (http://www.gams.com/).

Garrick, B. J., J. E. Hall, M. Kilger, J. C. McDonald, T. O'Toole, P. S. Probst, E. R. Parker, R. Rosenthal, A. W. Trivelpiece, L. A. Van Arsdale, and E. L. Zebroski. 2004. "Confronting the risks of terrorism: making the right decisions." *Reliability Engineering and System Safety* 86:129-176.

Golany, B., E. H. Kaplan, A. Marmur, and U. G. Rothblum. 2009. "Nature plays with dice – terrorists do not: Allocating resources to counter strategic versus probabilistic risks." *European Journal of Operational Research* 192(1):198-208.

Golen, E. F., S. Mishra, and N. Shenoy. 2010. "An underwater sensor allocation scheme for a range dependent environment." *Computer Networks* 54:404-415.

Heal, G. and H. Kunreuther. 2005. "IDS Models of Airline Security." *Journal of Conflict Resolution* 49(2):201-217.

IBM. 2011. "IBM ILOG CPLEX Optimizer." IBM ILOG CPLEX Optimizer. (www.ibm.com/software/integration/optimization/cplex-optimizer/).

Kaplan, S. and B. J. Garrick. 1981. "On The Quantitative Definition of Risk." *Risk Analysis* 1(1):11-27.

Kardes, E. 2005. "Robust Stochastic Games and Applications to CounterTerrorism Strategies." CREATE report.

Kim, H. S. and S. Park. 2006. "An Optimal Algorithm for the Sensor Location Problem." *Journal of Industrial Engineering & Management Science Joint Conference*.

Knauss, J. A. 2005. "Introduction to physical oceanography." P. 205 2nd ed. Prentice Hall.

Kunreuther, H. and G. Heal. 2002. "Interdependent Security." *Journal of Risk and Uncertainty*.

Lee, R. W. and J. J. Kulesz. 2008. "A risk-based sensor placement methodology." *Journal of Hazardous Materials* 158:417-429.

Lemke, C. E. and J. T. Howson. 1964. "Equilibrium points of bimatrix games." *Journal of the Society for Industrial and Applied Mathematics* 12:413-423.

Leung, M., J. H. Lambert, and A. Mosenthal. 2004. "A Risk-Based Approach to Setting Priorities in Protecting Bridges Against Terrorist Attacks." *Risk Analysis* 24(4):963-984.

Lin, F. Y. S. and L. P. Chiu. 2005. “A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks.” *IEEE Communications Letters* 9(1):43-45.

Makinen, G. 2002. “The Economic Effects of 9/11: A Retrospective Assessment.” Report for Congress.

McGill, W. L., B. M. Ayyub, and M. Kaminskiy. 2007. “Risk Analysis for Critical Asset Protection.” *Risk Analysis* 27(5):1265-1281.

Nash, J. 1951. “Non-Cooperative Games.” *The Annals of Mathematics* 54(2):286-295.

Paruchuri, P., J. P. Pearce, and S. Kraus. 2008. “Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games.” *In AAMAS*.

Paruchuri, P., J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. 2008. “Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games.” *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems* 2(895-902).

Pompili, D., T. Melodia, and I. F. Akyildiz. 2006. “Deployment Analysis in Underwater Acoustic Wireless Sensor Networks.” *WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks* 48-55.

Powell, R. 2007. “Defending Against Terrorist Attacks with Limited Resources.” *American Political Science Review* 101(3):527-541.

Sandler, T. and H. E. Lapan. 1988. “The Calculus of Dissent: An Analysis of Terrorists' Choice of Targets.” *Synthese* 76:245-261.

Simonoff, J. S., C. E. Restrepo, and R. Zimmerman. 2007. “Risk-Management and Risk-Analysis-Based Decision Tools for Attacks on Electric Power.” *Risk Analysis* 27(3):547-570.

Stengel, B. V. 2002. “Computing Equilibria for Two-person Games.” Pp. 1723-1759 in *Handbook of Game Theory with Economic Applications*. Amsterdam: Elsevier.

U.N. 2011. “Review of Maritime Transport.” UNITED NATIONS CONFERENCE ON TRADE AND DEVELOPMENT. (http://unctad.org/en/docs/rmt2011ch1_en.pdf).

Urick, R. J. 1983. in *Principles of Underwater Sound*. McGraw-Hill, Inc.

W., Cottle R. and G. B. Dantzig. 1968. “Complementary pivot theory of mathematical programming.” *Linear Algebra and its Applications* 1:103-125.

Wilhelm, W. E. and E. I. Gokce. 2010. “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.

Willis, H. H. 2007. “Guiding Resource Allocations Based on Terrorism Risk.” *Risk Analysis* 27(3):597-606.

Yates, J. T., R. Batta, I. Casas, and M. Karwan. 2011. “Allocating and Evaluating Defense Resources for the Protection of Critical Infrastructure from Terrorist Threats.” *Journal of Transportation Security* 4(2):145-169.

Zhuang, J. and V. M. Bier. 2007. “Balancing Terrorism and Natural Disasters—Defensive Strategy with Endogenous Attacker Effort.” *Operations Research* 55(5):976-991.

Zhuang, J. and V. M. Bier. 2010. “Reasons for Secrecy and Deception in Homeland-Security Resource Allocation.” *Risk Analysis* 30(12):1737-1743.