FACILITY PROTECTION OPTIMIZATION UNDER UNCERTAINTY

By

Patrick T. Hester

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

In

Mechanical Engineering

August, 2007

Nashville, Tennessee

Approved:

Dr. Sankaran Mahadevan

Dr. Gautam Biswas

Dr. David Dilts

Dr. Ken Pence

Dr. Mark Snell

To all of you who believed in me, especially my parents and my wife

ACKNOWLEDGEMENTS

I would especially like to thank Dr. Sankaran Mahadevan, my dissertation advisor, for his technical guidance and many hours reviewing and correcting my work in an effort to develop my writing ability. I would also like to express my gratitude to Dr. Gautam Biswas, Dr. David Dilts, and Dr. Ken Pence, members of my Dissertation Committee. Each of them has provided extensive guidance both academically and in a broader professional setting.

This work would not have been possible without the financial support of the National Science Foundation's Risk and Reliability Engineering and Management IGERT Program at Vanderbilt, the National Physical Sciences Consortium, and Sandia National Laboratories. This support is gratefully acknowledged. I would especially like to thank Dr. Mark Snell, Dr. Carla Ulibarri, and Dr. Robert Waters, my mentors at Sandia, for their continued efforts to keep my theoretical work grounded in reality.

No one has been more integral to my success in this process than my family. I would like to thank my parents for teaching me that all things are possible through education. Their encouragement has been a constant source of professional drive. Most importantly, I would like to thank my wonderful wife, Kasey, who has been there all along for me to vent when necessary, and who makes me a better person, both professionally and personally.

TABLE OF CONTENTS

Page

LIST OF TABLES

Name Page

LIST OF FIGURES

Name Page

LIST OF ABBREVIATIONS

The following abbreviations are used throughout the text:

CD	Connectivity Diagram
CDP	Critical Detection Point
CPS	Critical Path Set
FPO	Facility Protection Optimization
HTDF	Helper Team Detection Factor
HTTF	Helper Team Time Factor
ICP	Insider Compromise Point
PPS	Physical Protection System
STDF	Simultaneous Team Detection Factor
STTF	Simultaneous Team Time Factor

CHAPTER I

INTRODUCTION

1.1 Overview

The goal of this study is to develop a decision-making methodology which takes into account how multiple adversary (thief, saboteur, terrorist, etc.) teams may attack a facility and assist facility operators in designing protection systems to defend against such attacks. This results in the ability for the facility operator to assess relative facility and/or infrastructure safety, and make decisions regarding how to optimally allocate resources in physical protection elements to balance cost and performance. These physical protection elements enable the facility owner to prevent attacks through deterrence and to defeat the adversary (through detection, delay and response) if he or she chooses to attack.

1.2 Physical Protection Systems

A physical protection system (PPS) integrates people, procedures and physical safeguards to protect facilities against theft, sabotage, or other malevolent actions. The concept of physical protection systems was developed in 1972[55] and later revised by the International Atomic Energy Agency[56] and explored in great detail at Sandia National Laboratories[40]. PPS's include *detection*, *delay*, and *response safeguards* that help to protect against an adversary threat. *Detection* involves discovering an attempted or

actual intrusion in a facility and includes items such as exterior and interior sensors and entry control. In order to be successful, detection must be accompanied by assessment of the alarm, which is often provided through a mechanism such as closed circuit television, to indicate whether or not the alarm was genuine or false. *Delay* involves slowing adversaries until the response force can arrive, and it includes items such as barriers, fences, and walls, which inhibit the adversary. Delay time should be sufficient enough to allow for security personnel to respond in time to interrupt the adversary before completing his or her malevolent act. Finally, *response* involves the response force tasked with interrupting the adversary once his or her presence has been detected. Both on-site and off-site security personnel contribute to the ability to thwart potential attackers by timely response. As can be expected, a more rapid response is necessary in the case of sabotage (if the goal of the response force is to prevent the adversary from accessing the target) than theft.

Concepts of PPS were originally developed with nuclear facilities in mind, but they are equally applicable to any facility that requires protection from an adversary threat. The ultimate objective of a nuclear physical protection system is to prevent the theft of nuclear materials or sabotage of nuclear materials or facilities. Theft and sabotage can be prevented in two ways: by deterring the adversary such that an attack is not attempted or by defeating an adversary once an attack has begun. Deterrence is achieved by a thorough and balanced physical protection system that is viewed as too difficult to defeat by adversaries[40,54,55,56]. That is, the measures in place to defeat the adversaries are too numerous and powerful for the adversary to attempt an attack, rendering the target unattractive.

To prevent against theft, the objective is to protect against unauthorized individuals gaining access to the intended target and removing it from the facility. In sabotage, however, the goal of denial of access is to prevent the adversary from ever gaining access to the material, whereas the goal of denial of task is to prevent the adversary from committing an act of sabotage. Although the two objectives share similar concepts, the strategy to stop an adversary in each situation is different. To protect against some theft targets, one strategy is to use delay measures in close proximity to the target to stall the adversary long enough for the security personnel to call for help or to stop the adversary themselves. To protect against some sabotage targets, however, delay is necessarily far away from the target, to allow sufficient time for on-site security personnel to interrupt the attack and prevent the adversary from gaining access to the target.

In order to protect against these threats, the designer of a physical protection system should keep the following concepts in mind: *defense in depth*, *minimum consequence of component failure*, *balanced protection*, and *graded protection*. *Defense in depth* ensures that there are multiple safeguards in place to stop an adversary regardless of which path or attack strategy he or she chooses to employ. Defense in depth slows down an attacker by requiring they defeat multiple PPS elements to achieve their goal, ensuring facility protection in the event of the defeat of a single safeguard element.

Related to defense in depth is the concept of *minimum consequence of component failures*. Contingency plans must be developed which allow the security system to continue to operate in the event a safeguard within the system is lost. An example of this

would be an emergency backup generator that starts automatically once a primary power source is disabled. Incorporating minimum consequences of component (i.e. safeguard components or response force personnel) failure, however, can be costly as additional components add to the installation and operating costs of the facility.

Balanced protection refers to the concept that an adversary should be hindered by PPS elements independent of what attack strategy and path he or she chooses. A completely balanced system would require an equal amount of time to commit theft or sabotage independent of the route chosen. This is not likely to be possible due to budget and physical constraints, nor is it necessarily desirable. Certain elements within the system may have large inherent delays built into them, such as walls, while other components may not have as large of a delay time, such as a chain link fence. However, all elements should provide a sufficient level of security for the facility. The goal of a well-designed PPS is to provide adequate protection against all threats on all possible paths, while being mindful of practical constraints such as cost and reliability.

Graded protection refers to the concept that a facility should be protected in a level that is commensurate with its importance, or consequence of loss. Facilities whose loss, theft, or destruction would cause harm on a national level should be protected to a higher level of security than a facility that is of no national consequence. Determining which facilities are the most important involves ranking of the facilities according to their threat level and protecting them accordingly. This decision is made at an administrative level, such as in the Departments of State and Homeland Security. Further guidance on the topic of PPS is provided by IAEA[56] and Garcia[40].

Nuclear reactor facility operators typically use the Nuclear Regulatory Commission's (NRC) Design Basis Threat (DBT) as a method of determining what level of safeguards is sufficient. The DBT requirements[1,2] describe general adversary characteristics, including group size and capabilities, which nuclear facility operators must be prepared to defend against. These requirements are set by the NRC and include protection against radiological sabotage and theft of nuclear material. Facilities use the DBT to establish what safeguards are necessary for protection. Following the September 11, 2001 attacks, the NRC conducted thorough reviews of security procedures and standards to ensure that nuclear facility practices were adequate given the escalating terrorist threat. After its review, the NRC realized some of its requirements were inadequate. In 2003, the NRC issued an updated DBT guideline[2] which stressed the need for enhanced security procedures at critical facilities. After two years of implementation and observation, the NRC felt that these rules should be more generically imposed to include more facilities on certain license classes. This decision was based on adversary trend analysis and input from Federal law enforcement and the intelligence community. In 2005, a new report was issued by the NRC which reflected these observations, including enhanced adversary capabilities[79]. Improvements deemed necessary by the NRC report included giving consideration to "the potential for attack on facilities by multiple coordinated teams of a large number of individuals"[79] and attacks from multiple locations.

However, since the current DBT guidelines[1,79] do not account for a multiple team attack threat, current facility operators are not required to design for this threat. Thus, their current PPS designs are may prove to be insufficient to handle the increasingly

complicated scenarios when multiple adversary teams operate in collaboration. This study develops a methodology to defend against such scenarios.

1.3 Game Theory and PPS

The above section discussed the basic concepts of physical protection systems without consideration for the impact that adversarial and response force decision making has on PPS design. Analyzing how the facility operator needs to design the PPS for a real world situation, where the conflict between adversarial and response forces is involved, can use the concept of game theory[80]. Research in the area of game theoretic network interdiction done by Washburn and Wood[102] is used to develop the methodology in this study. Assuming that the adversary and the facility operator are all-knowing with regards to a networked facility, (that is, they know the arc travel times, detector locations, etc.) the adversary will always choose the path which optimizes his or her objectives (e.g. minimum travel time through the facility, minimum probability of detection, etc.), that is, the optimal path. This assumption is based on the idea of game theory for a repeated game, and it assumes that the adversary's and facility owner's strategies are static and do not change with time. Overall, the adversary wants to maximize his or her objective, while the interdictor wants to minimize that same objective. Washburn and Wood[102] show that these conflicting objectives result in a zero sum game situation in a network in which the objective of interest for the interdictor is to minimize the maximum flow. This zero sum game strategy can easily be adapted to any objective, e.g. maximizing travel time or maximizing the probability of effectiveness in defeating the adversary. The

problem of maximizing the utility of the security system (as defined in Section 3.2) is addressed in the example problems throughout this study.

The reasoning behind choosing the optimal path is that the adversary knows where all the safeguards have been placed throughout the facility, so it is to his or her best advantage to take the route which best achieves his or her goal. One may argue that if the interdictor knows the adversary will take the optimal path, he or she will spend his or her money on safeguarding the optimal path. This would, however, result in a new path being optimal, and, the adversary would not choose the previous optimal path for travel. Since the equilibrium state between the two forces is for the adversary to travel along the optimal path, it is the interdictor's goal, then, to ensure that the adversary's objective function value along the optimal path is above a threshold value either to deter the adversary from attempting an attack, or to ensure the adversary will be defeated if he or she chooses to attack.

It is in the best interests of the interdictor to attempt to ensure that all possible paths through the facility have objective function values above the threshold value. This concept is further discussed by the IAEA[56] and Garcia[40], as well as in the previous section as the theory of balanced protection.

1.4 Research Objectives

The background presented in Sections 1.1-1.3 provides motivation for this study, which consists of the following four objectives in order to ultimately develop a methodology to assist facility operators in protecting facilities against multiple adversary team threats:

1. Develop a conceptual framework for computing system effectiveness.
2. Develop an optimization under uncertainty methodology for facility protection against a single adversary team attack.
3. Extend the single adversary methodology to handle multiple team attacks.
4. Demonstrate the usefulness of the completed methodology by applying it to problem of large scale complexity.

Knowing that adversarial threats are real and escalating, facility owners must seek ways to develop an effective PPS to protect their critical facilities. *Objective 1* of this study, pursued in Chapters II and III, establishes a problem framework that allows a facility owner to develop a PPS for a new facility, perform trade-off analyses to determine the relative merit of differing PPS upgrades, and analyze existing systems to determine if a facility's PPS is performing as expected. Incorporating these three tasks simultaneously is an improvement of this study as current methods focus on either facility design or analysis. Chapter II provides a discussion of current metrics for system effectiveness and potential approaches for evaluating facility system effectiveness. The purpose of Chapter III is to discuss the features of the facility protection optimization problem and develop a solution framework for the problem.

Once a mathematical formulation has been developed, *Objective 2*, pursued in Chapter IV, develops the details of solving the complete facility protection optimization under uncertainty problem which can handle a single adversary team attack. The methodology utilizes an efficient first-order reliability method based approach to incorporating stochastic behavior, allowing for a larger number of random variables to be incorporated than in previous studies. Additionally, the methodology presented in

Chapter IV incorporates conflict between adversarial and facility guard forces and timely detection, concepts that are previously omitted from facility protection methodologies. Finally, the approach presented in Chapter IV presents a novel approach to decoupling adversary critical path selection and safeguards optimization. All of the contributions of this study allow for realistic (larger scale) problems to be analyzed.

Following the conceptual and mathematical development of a single adversary facility protection methodology, *Objective 3*, pursued in Chapter V, extends the single adversary team methodology developed in *Objectives 1* and *2* to handle multiple team attacks. This is a significant development of this study as a mathematical methodology to combat multiple adversary team attacks against a facility does not currently exist. The purpose of this task is to ensure the capabilities of the developed methodology are commensurate with the expected threat level present in the world currently.

An underlying goal of this study is to develop a methodology that can handle analysis of a large scale problem. Garcia[41] has provided a detailed description of a realistic facility which can be used to evaluate the methodology developed in *Objectives 1-3*. *Objective 4*, pursued in Chapter VI, investigates the usefulness of the complete facility protection methodology by applying it to a real world problem. Demonstrating the capability to analyze a large scale problem is also an important development in this study as current methods require significant computational effort that is prohibitive to analyzing these problems.

Chapter VII summarizes the conclusions of the study and provides recommendations for future research.

CHAPTER II

SYSTEM EFFECTIVENESS

The first objective of this study relates to measuring whether or not a facility's PPS is providing an adequate level of protection against an adversarial attack. This measure is referred to as system effectiveness. Although there is no universally accepted definition, system effectiveness is defined in this study as the degree to which a system, defined as a set of interrelated components working together to achieve a common goal, meets their overall designed purpose. This definition is intentionally vague as the system being analyzed helps to dictate what measure of effectiveness is appropriate. System effectiveness can also be thought of as a performance measure (metric). Section 2.1 discusses the metrics for system effectiveness used in this study, including a discussion on combining conflicting objectives. Section 2.2 describes current methods for computing system effectiveness, including their advantages and disadvantages. The chapter ends with a summary in Section 2.3.

2.1 Typical System Effectiveness Metrics

Any metric on which a system's performance can be evaluated can be used to assess system effectiveness. Depending on the application, a specific metric may be preferred. For example, facility throughput and efficiency may be important measures to a line manager at a manufacturing facility while reduction of casualties and minimum

exposure to harm may be important to the commander of a military operation. While the objective of a specific system dictates which measure is most important, cost and system performance often act as the two most prominent metrics in PPS design[54]. These two metrics are naturally competing as increased cost is typically required in order to achieve improved system performance. As a result, the goal of the designer is to develop a system which optimally balances cost and performance.

Metrics may be employed at the system, component, or group of components level. For example, the manager of a facility may wish to know the cost of the facility's entire physical protection system, whereas the security force manager may focus only on the costs of the security personnel.

Further discussion of cost and performance is provided in Sections 2.1.1-2.1.2, respectively.

2.1.1 Cost

Cost is often a driving factor in many organizations and it can therefore have a direct impact on system effectiveness. Throughout this study, costs are treated as life cycle costs for the safeguards, including related installation, operation, and maintenance costs. These costs are assumed to be deterministic.

When making decisions regarding safeguards' expenditures, the analyst must decide if additional safeguards are cost effective. That is, is the additional cost of added safeguards a justified investment? In order to make this decision, the analyst must perform a tradeoff between cost and performance, which is discussed in the next section.

2.1.2 Performance

Performance may be proportional to cost (at least initially, if the safeguards' budget is spent wisely, until the point of diminishing returns), thus, the objective of maximizing performance competes directly with minimizing cost. Performance can be measured in many different ways, but for the purposes of PPS design, performance is considered in terms of the probability of security system effectiveness (P_E) and the probability of security system failure ($P_{FS} = 1\text{-}P_E$). This failure can be measured in terms of specific failure of a component of the system, as total system failure, or as failure to meet some performance threshold. Any of these measures provides insight to the system designer and operator in designing/operating the system. The following section discusses P_E and its components further.

2.1.2.1 Probability of System Effectiveness

There are many strategies available to an adversary to achieve his or her goal, which is to travel his or her chosen attack path with the least likelihood of being defeated, or the highest likelihood of succeeding[40]. The adversary can choose a brute force attack to minimize the time required to complete his or her actions, with little regard to the probability of being detected. Conversely, the adversary may choose a stealth attack to minimize detection probability along a path without regard to travel time.

Effectiveness measures are available to deal with either of the adversarial approaches. To deal with a brute force attack, the cumulative delay time along the path (T_{min}) is compared with the guard's response time (T_G). An adequate PPS allows for the guards to have ample time to respond to an incident. In this case, the minimum delay along the remaining segments of the path (T_R) should be greater than or equal to the

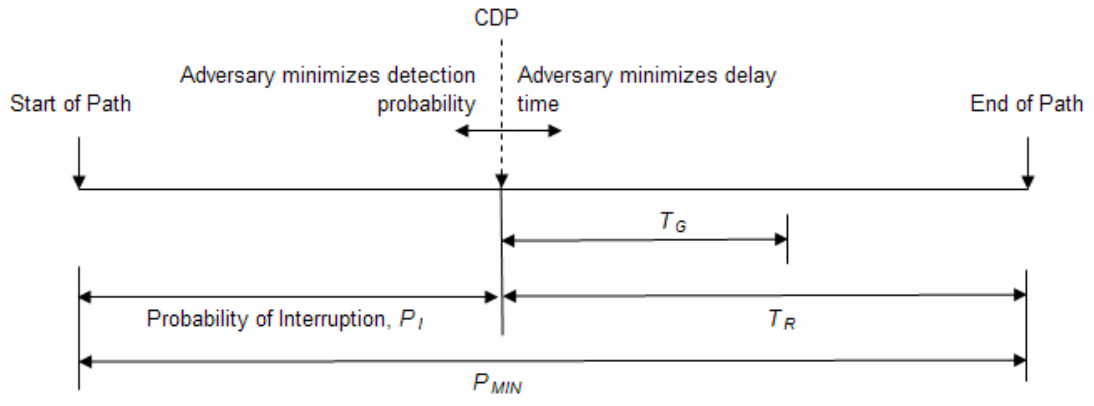
guard response time. This can be achieved by either reduced guard response times (by assigning a larger guard force to the facility) or by increasing the delays along the adversary's path (by installing barrier safeguards). The disadvantage of this approach is that it does not adequately address the issue of detection. Without detection, the delay is meaningless, as the response force will not be alerted to the adversary's presence.

To deal with a stealth attack, the cumulative probability of detection (P_{min}) of the adversary is analyzed. An effective system results in a P_{min} that is above an acceptable threshold level. This can be achieved by installing detectors throughout the facility. The disadvantage of this approach is that it does not consider the resulting delay time necessary for guard force response. Without delay elements to slow the adversary, detection is meaningless as the guard force may not have adequate time to interrupt the adversary.

Referring to the aforementioned methods, Garcia states: "Due to the deficiencies of each of these measures, neither delay time nor cumulative probability of detection alone is the best measure of system effectiveness. A better measure of effectiveness is timely detection, which combines P_{min}, T_{min}, and T_G."[40]

The point where the remaining minimum delay along the adversary path just exceeds the guard response time is referred to as the critical detection point (CDP). The probability of interruption (P_I) is the cumulative probability of detection from the path's start up to the CDP, and it serves as the basis of Sandia National Laboratories' Estimate of Adversary Sequence Interruption (EASI) model[10,40] for calculating system effectiveness. P_I is noted to be different than P_{min} as it does not represent the detection across the entire path. The timelines over which P_I and P_{min}, as well as T_G and T_R, are

calculated are illustrated in Figure 2-1. P_I itself is often used as a measure of system effectiveness as it represents timely detection. There is a limitation to this approach as timely detection only includes detection, delay, and guard response time. It does not include any resulting conflict between the guard force and the adversaries, represented through the probability of neutralization, P_N. P_N refers to the probability that the facility's guard force will defeat the adversary if a physical confrontation occurs (i.e. if the adversary does not surrender once interrupted). Simulation tools[99] and analytical approximations[15] exist to determine the results of such conflicts. Determination of P_N is discussed in detail further in Section 3.2.1.3. Garcia includes detailed discussion on the calculation of P_I, along with some examples[40]. In the methodology presented in this dissertation, P_N is incorporated, however the time required for neutralization is assumed to be negligible.



Note: CDP is defined such that $T_G \leq T_R$

Figure 2-1: Illustration of Timely Detection

It is worth noting that the location of the CDP is influence by the adversary's objectives. If the adversary's goal is to simply enter a facility and commit an act of sabotage such as a suicide bomber, the CDP is earlier in the path as the adversary task time is reduced. Conversely, if the adversary's goal is theft, his or her total task time increases and the security force has more time to respond to his or her detection, thus the CDP is later.

Garcia also uses P_I as part of another measure of system effectiveness[40], that is dependent on the adversary being identified as an outsider or an insider. Outsiders are individuals who have no special insight into the operations of the facility, while insiders (e.g. facility workers) have some intimate knowledge of, access to, and authority at the facility which gives them an advantage when attacking the facility. If the attacker is an outsider, system effectiveness, P_E, is calculated as $P_{E\ outsider} = 1-(1-P_I*P_N)^{103}$. If the attacker is an insider, the equation becomes more complicated. That is because the insider typically does not act abnormally when trying to commit theft or sabotage until he or she is detected or feels threatened, at which point he or she acts like an outsider. System effectiveness for insiders is calculated as $P_{E\ insider} = P_{ICP}*P_{N,NV} + (1-P_{ICP})*P_{E\ outsider}$. P_{ICP} is the probability of interruption up to the insider compromise point (ICP). Until the ICP, the insider acts like any other individual associated with the facility so as not to create suspicion of his or her intentions. It is worth noting that the ICP is an insider-dependent point. Some insiders will feel threatened earlier on and thus become violent whereas others will stay calm for a longer period of time[40]. The latter makes his or her probability of interruption much lower as his or her motives are not immediately obvious. $P_{N,NV}$ is the probability of neutralization for a nonviolent (NV) individual until

the ICP. Until the ICP, the insider is likely to be nonviolent if his or her motives are discovered. $P_{E\ insider}$ and $P_{E\ outsider}$ offer a structured approach to determining whether or not a PPS provides sufficient protection against an adversarial threat.

Insider attacks are considered outside the scope of this study, and therefore, only outsider attacks are considered. All equations are developed with this simplification in mind. Future work should consider extending this methodology to include insider attacks.

Previous facility protection work has concentrated on adversaries choosing the path with the least probability of interruption[40] and probability of detection[81]. The former study lacks a formal mathematical methodology to analyze a facility and the latter lacks the inclusion of timely detection. Both methods lack inclusion of the probability of neutralization. Inclusion of neutralization is important when dealing with a multiple team attack in order to account for scenarios in which the adversary attacks the facility with a large number of individuals, thereby overwhelming the guard response force. Since no previous mathematical methodology exists which deals with a multiple team attack, it is natural that previous methodologies ignore the inclusion of P_N, as it adds complexity to the analysis. Simply analyzing this type of scenario by looking at P_I is inadequate as the response force may be able to *interrupt* the adversary force, but if they are greatly outmatched, they will not be able to *neutralize* them. Neutralization can no longer be assumed to be successful with a multiple team attack as the size of the adversarial force may be very large. As a result, P_E is chosen over P_I as a superior measure of the performance of the PPS system.

2.2 Methods for Computing System Effectiveness

Section 2.1 discussed the system effectiveness metrics for evaluating the PPS. The purpose of this section is to review current practices for analyzing system effectiveness. The goal of any analysis of security system effectiveness is to meet the three capabilities outlined in Section 1.4, that is, to be able to design a new system, perform an upgrades analysis (which can be thought of as a new system design with a reduced budget), and analyze the performance of an existing system. Following is a discussion of the different methods available for computing system effectiveness. Some of these methods are currently utilized for system effectiveness analysis, while others can be adapted to be used for this purpose. Each method includes a brief introduction, as well as a discussion of its ability to meet the outlined objectives, and any advantages or disadvantages specific to the method.

2.2.1 Attack Trees/Graphs

The fault tree method is a graphical enumeration of the ways in which component failures can lead to a system failure[23,35,95,100]. Fault trees are often used to analyze critical facilities, such as nuclear reactors[100]. Further, Schneier[95] coined the term "attack tree" to mean a specific type of fault tree in which the failure in question is a malevolent attack by an adversary. A reliability block diagram[3,52,91] is a particular type of fault tree where the goal is to evaluate overall system reliability. These techniques involve a top-down evaluation of a particular facility. The analysis begins by generating a list of possible attacker goals. Each goal is the root (top) node of a separate tree. Next, the analyst enumerates all likely possible attacks against the goals and populates the trees. Once an

attack tree has been constructed, the likelihood of the top event can be computed. If the top node is formulated as P_E, then system effectiveness can be calculated.

This method is advantageous due to its simplicity of implementation and the large amount of previous work using attack and fault trees[23,35,95,100]. Additionally, many potential attack strategies can be captured in a single attack tree and the resulting analysis can identify the most vulnerable path as that path with the lowest value of system effectiveness. There are two distinct disadvantages, however. This method appears to be best suited for PPS analysis of a set of potential attack strategies, but it is not appropriate for security system design. Further, while attack trees have been used to solve similar problems, they have not been utilized for P_E calculation. Inclusion of multiple teams, conditional probabilities, timely detection, etc., results in an inability to use attack trees.

2.2.2 Adversary Path Analysis

The approach explored by Garcia[40] focuses on the existence of adversary paths. An adversary path represents an ordered series of actions which, if completely successfully, execute an act of theft or sabotage. Figure 2-2 shows a sample adversary path to destroy a pump in an industrial facility.

The adversary path analysis method repeats the P_I (or P_E) calculation for many paths in the facility and compares these values to one another to find the path of the greatest system vulnerability (i.e. the path with the lowest P_I (or P_E)). The analysis can be quite cumbersome without a formal methodology for which paths to analyze. If the lowest P_I (or P_E) path is deemed unacceptable, system upgrades must be explored in

order to improve the system. Garcia also includes numerous examples for analyzing identified system upgrades[40].

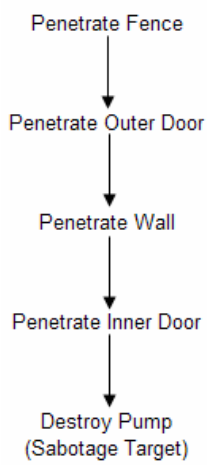


Figure 2-2: Sample Adversary Path[40]

The advantage of this method is that it provides a methodology for analyzing facility system effectiveness and is in fact currently used in this manner[40]. The disadvantage of this method is that it requires separate enumeration of candidate adversary paths. As a result, this method provides an appropriate analysis tool, but it does not provide design capabilities.

2.2.3 Network Representation

Network (or graph) representations involve translating a facility into a network much the way it is done for attack trees, although in this case the nodes typically represent physical locations within the system (rather than states) and the arcs some measure of connectivity between the locations (rather than actions to transition between

states) such as travel time or cost (in dollars) to travel between nodes. The network is then analyzed simultaneously from the perspective of the adversary, who is trying to defeat the facility, and the facility owner, who is trying to protect the facility. Well-established mathematical approaches exist for handling network representations of facilities[3]. Network representations provide for a structured approach to analyzing networks and making decisions, and they have been employed extensively to analogous problems as the design and analysis of security systems, although not using the same effectiveness metrics as are desired in this study[26,38,46,59,60,77,81,87,94,104].

Network representations are an ideal choice for adversary path selection as they have been utilized to solve analogous problems such as shortest path and minimum flow routing through networks[3]. Currently, problems in which maximization of minimum P_E is the desired objective have not been explored using these methods. As a result, they appear well suited for PPS design and analysis if previous studies can be extended to utilize this metric.

2.2.4 Dynamic systems

While the methodology presented in this study does not include dynamic behavior, it is important to realize that this is a potential extension to this dissertation. Petri nets[61,74,84,85,86] and agent based simulation[34,105] may be used to address dynamic aspects of the facility. A Petri net is an abstract, formal model of information flow. They model distributed systems as a directed bipartite (or separable) graph. Advantages of Petri nets include the ability to handle dynamic behavior and multiple simultaneous events[84], which may be useful in this methodology. Their application, however, appears

to be better suited for analysis than for design. Agent-based systems also provide a technique for analyzing dynamic systems. In these systems, agents are computer systems that are programmed with logic (i.e. rules concerning beliefs, desires, etc.) and they are capable of making independent decisions based on this logic. Adversary and response force actions could be simulated using agents. Similarly to Petri nets, however, agent-based systems appear to be best suited for analysis and not for design. Additionally, dynamic systems are outside the scope of this study.

2.2.5 Method Choice

All of the methods of computing system effectiveness mentioned in Sections 2.2.1-2.2.4 offer particular advantages and disadvantages for system effectiveness calculation. A summary of these methods is provided in Table 2-1.

Ultimately the goal of this chapter is to select or create a problem representation framework which is able to analyze an existing PPS, design a PPS for a new facility, and perform a PPS upgrades analysis using a user-defined effectiveness metric. No one current method meets all of these criteria, and thus, the only way to achieve this goal is to develop a method which is a combination of the approaches found in Sections 2.2.1-2.2.4. Attack trees, adversary path analysis and dynamic systems are inappropriate for security system design, while graph theory representation, attack trees, and dynamic systems methods do not currently utilize P_E as a metric for system effectiveness. As a result, the methodology presented in this study incorporates a combination of the security system analysis capabilities of adversary path analysis with the design capabilities of graph theory. Adversary path analysis is a natural choice for incorporation in this

methodology given its current use as an analysis tool for physical protection systems. Graph theory complements adversary path analysis by providing a methodology that is capable of identifying vulnerable adversary paths (given the analysis capabilities of adversary path analysis). Further, the time-expanded aspect of dynamic systems is outside the scope of this study and therefore unnecessary, and attack trees do not appear useful given their limitations. Future work may include developing a methodology that combines the methodology developed in this dissertation with dynamic systems. This potential extension is discussed in Chapter VII.

Table 2-1: Summary of System Effectiveness Methodologies

Problem Representation	**Advantages**	**Disadvantages**
Attack Trees/Graphs	- Many potential attack methods can be captured in one tree	- Best suited for analysis and not design - Inclusion of multiple teams, conditional probabilities, timely detection, etc., results in inability to use attack trees
Adversary Paths	- Designed specifically for existing security system effectiveness evaluation and therefore seems most ideal for this application	- Best suited for analysis and not design - Only appropriate when potential adversary attack path is known
Networks	- Suitable for new system design or existing system analysis	- No methods currently exist which utilize P_E as a measure of system effectiveness
Petri Nets/Agent Based Simulation	- Able to handle dynamic systems and multiple simultaneous events - Useful for complicated event analysis	- More appropriate for analysis than design - Dynamic systems not included in scope of study - No methods currently exist which utilize P_E as a measure of system effectiveness

2.3 Summary

This chapter provided an overview of both system effectiveness metrics and problem representation frameworks. Probability of system effectiveness and cost were identified as primary metrics for facility system effectiveness measurement. Several types of approaches to represent the problem were investigated for use in calculating the system effectiveness metrics, and networks and adversary paths were found useful. Adversary paths were found to be useful for analysis since they are currently utilized for this purpose, while networks were found to be useful for their versatility in handling different optimization objectives and capabilities (design and analysis). These concepts will be used in the following chapter, where the problem features of facility protection optimization are discussed and utilized to develop a problem solution framework.

CHAPTER III

PROBLEM SOLUTION FRAMEWORK

The previous chapter discussed system effectiveness measures and potential frameworks to evaluate these measures. The purpose of this chapter is to discuss the features of the facility protection optimization problem and to develop a general solution framework for this problem. The chapter begins with the development of a network representation in Section 3.1. A general problem formulation utilizing this network representation is then discussed in Section 3.2. Section 3.3 discusses how network interdiction concepts can be utilized to solve this problem formulation. A discussion of the approach for handling stochastic behavior in the formulation is provided in Section 3.4. Finally, a summary of the chapter is provided in Section 3.5.

3.1 Network Representation

A graphical representation of the relationship between the elements or actions in the system, referred to as a connectivity diagram (CD), is developed first in order to facilitate calculation of the objective discussed in the previous chapter. Throughout this study, the terms network and CD are used interchangeably. For this step, the idea is to analyze the facility through the eyes of the adversary and determine how he or she would attempt to achieve his or her objectives. If a subject matter expert for the system is available, which is often the case when an upgrades' analysis is being performed rather

than a new system design, or the system is not overly complicated, a task-based CD should be constructed. If no subject matter expert is available and the system is overly complicated, a location-based CD should be constructed, as this formulation allows for greater flexibility at a cost of lost accuracy (since the information incorporated involves greater uncertainty). The following discussion of CD development is provided as a brief overview but it is not intended to be exhaustive. That is, complete development of a CD is outside the scope of this study and is understood to be undertaken by a facility operations' subject matter expert[40].

3.1.1 Task-Based Connectivity Diagram

For a task-based CD, the diagram should be constructed with arcs representing actions and nodes between them representing milestones or physical locations. Actions should reflect how to transition from one milestone or location to another. If two rooms in a facility are represented by nodes on a CD, the arcs between them should indicate the links between the nodes (i.e. a door between the rooms or a hallway). The final node should be the milestone of adversary mission success. An example of a task-based CD is shown in Figure 3-1. The values shown on the arc represent the expected value for the action, in terms of the effectiveness metric. In this example, the numerical values in Figure 3-1 are in terms of detection probability. This example is based on the scenario presented in Figure 2-2 in Section 2.2.2, with the addition of "bribe guard" as a way to transition from the node of outside the facility to the node of inside the fence.

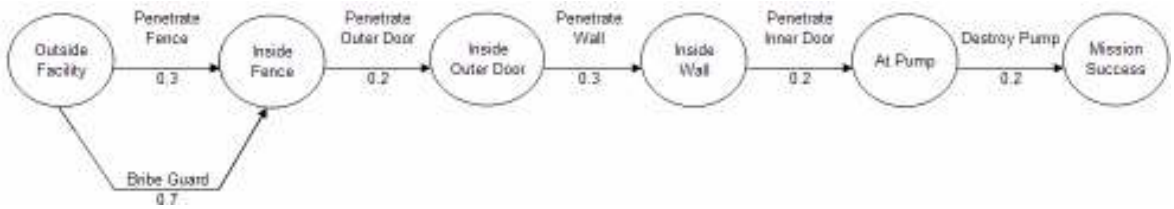


Figure 3-1: Task-Based Connectivity Diagram

3.1.2 Location-Based Connectivity Diagram

For a location-based CD, the diagram is constructed with nodes representing physical locations within the facility. The existence of an arc between two nodes indicates that there is a method by which to transition from one node to the other, i.e. there would be an arc between two nodes if there is a physical connection between them. The arc values represent the best-known estimate for the transition (e.g. travel time) between nodes. Since location-based CD's are often constructed based on less information than task-based CD's (since they are for new projects and have no subject matter experts), these estimates can be very approximate. If no other data is available, an estimate of arc values can be made based on relative distances between nodes. In order to do this, the largest physical distance between two nodes in the CD is given a value of 1 and the smallest distance, 0, and all intermediate values are interpolated based on distance of that particular arc vs. the longest distance. This method is appropriate for travel time and detection probability (P_D), quantities which are needed for safeguards' analysis. Additionally, it should be noted that no specific method for transitioning between nodes is specified, as in task-based CD's. That is to say, if there are two transition measures between two specific nodes, only one arc connects these nodes. For instance, rather than having two arcs transitioning between "outside facility" and "inside

fence" as in Figure 3-1, the average of the two detection probabilities from this task-based CD has been used in the location-based CD shown in Figure 3-2. Additionally, other approaches such as the minimum value or the geometric mean could be used. Also, since location-based CD's are based on less information than task-based CD's, in the example shown in Figure 3-2 it is assumed that less data is available for the transitions between "inside outer door" and "inside wall" and between "inside wall" and "at pump." Since less information is available, the values in this example between these nodes are assumed to be more conservative than those present in Figure 3-1. In general, location-based CD's will be more conservative as the values used to generate them are typically less certain than equivalent task-based CD's and conservative assumptions are made to generate data to populate the CD.

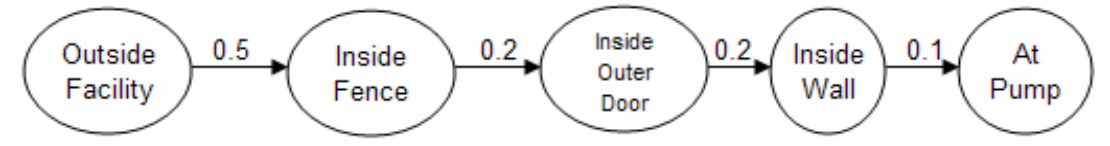


Figure 3-2: Location-Based Connectivity Diagram

Since actions cannot be reflected in a location-based CD, the values reflected in a location-based CD are likely to be less accurate than a task-based CD. For instance, the probability of detection for transitioning from inside the wall to being adjacent to the pump (and sabotaging it) is only reflected as one value in the location-based CD (Figure 3-2), whereas the task-based CD (Figure 3-1) reflects the actions of both penetrating the door to arrive at the pump and destroying the pump. This distinction is important to keep in mind when deciding to use either a task-based or location-based CD.

3.1.3 Hybrid Connectivity Diagram

Another option for creating a CD involves a hybrid of task-based and location-based CD's to describe the facility in question. This is often a useful alternative when the designer has some data, but not enough to accurately create a task-based CD, but he or she wants more detail and accuracy than a location-based CD provides. In order to implement this method, the designer collects as much data as is available and populates a CD with this information. The gaps in the tasks present in the CD are represented as they would be in a location-based CD, with the best estimate available for the arcs. An example based on Figure 3-1 is shown in Figure 3-3. In this case, data were not available for the transitions between "Inside Outer Door" and "Inside Wall" and "Inside Wall" and "At Pump," so they were replaced with best-known estimates, as in Figure 3-2.

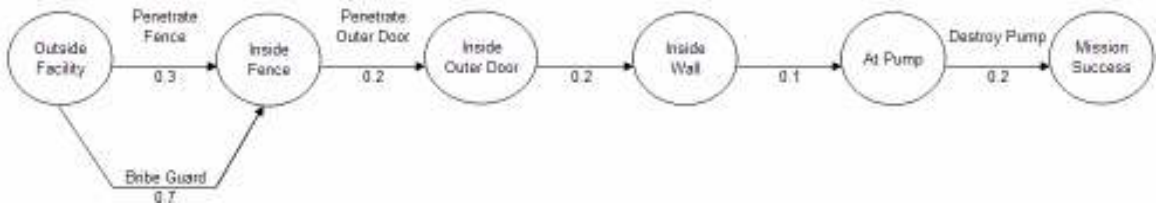


Figure 3-3: Hybrid Connectivity Diagram

This concept of connectivity diagrams allows for a simple representation of a facility and/or adversary actions which facilitates the methodology developed in this dissertation. In addition to identifying the objectives and constraints and constructing a CD, several other quantities must be specified in order to completely define the problem prior to undertaking a PPS design. These additional inputs are discussed in the following section.

3.1.4 Problem Inputs

While the CD is helpful for representation purposes, it needs to be translated into an arc-node incidence matrix for use in analysis and design. This matrix is a sparse matrix, with directed arc names as column headings, node names as row headings, a 1 in the matrix if the arc leaves from the corresponding node, a -1 if the arc enters the corresponding node, and 0 elsewhere. Additionally, a node balance matrix must be constructed. This is a matrix with arc names as row headings, a 1 at the origin of the CD, a -1 at the destination, and 0 elsewhere. Figure 3-4 shows a sample directed network graph. The nodes in the graph (and all other network graphs in this study, unless otherwise noted) are numbered in no particular order, with "O" representing the origin of the facility (instead of 1) and "D" representing the destination in the graph (instead of the terminal node number). Table 3-1 and Table 3-2 show the arc-node incidence matrix and node balance matrix, respectively, corresponding to Figure 3-4. More information on these concepts can be found in Ahuja[3].

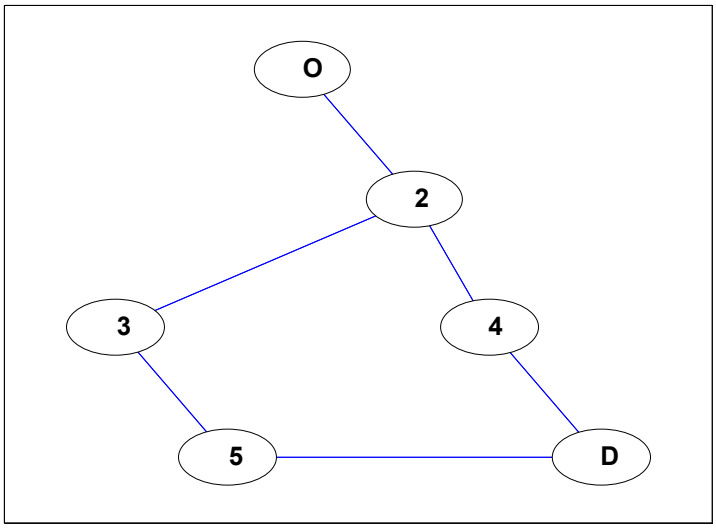


Figure 3-4: Sample Network

Table 3-1: Sample Arc-Node Incidence Matrix

	O-2	*2-3*	*2-4*	*3-5*	*4-D*	*5-D*
O	1	0	0	0	0	0
2	-1	1	1	0	0	0
3	0	-1	0	1	0	0
4	0	0	-1	0	1	0
5	0	0	0	-1	0	1
D	0	0	0	0	-1	-1

Table 3-2: Sample Node Balance Matrix

O	1
2	0
3	0
4	0
5	0
D	-1

The next step in defining the problem is to identify the safeguards' options available for the PPS. This definition includes the cost, detection probability, and travel time associated with the individual safeguard if it were to be installed. Safeguard definition also includes identifying whether or not the safeguard is a response force safeguard. This is an important distinction as guards help to increase both P_I and P_N.

Also, the uninterdicted (or initial, that is, including all installed safeguards, if an upgrades analysis or current system analysis is being performed) P_D (detection probability) and T_t (travel time) for each arc, an adjacency matrix of travel times between arcs, baseline guard response time (this corresponds to on-site guards if they exist, or off-site response force if not), and the number of potential origins and destinations in the problem must be specified. Typically the number of origins and destinations is one, but

in the event the problem includes multiple origins and destinations, this must be identified prior to analysis.

The final piece of information required to completely define the problem is the number of adversary teams considered in the analysis and, if the number of adversary teams is greater than one, their effect on one another. The number of teams should reflect the threat the facility is accurately expected to face and not simply an overly-conservative estimate. The reason for this is the safeguards' cost associated with protecting against adversary teams increases dramatically as the number of teams does. Experiments showing these results are presented in Sections 5.2 and 5.3. If the number of adversary teams is greater than one, the helper team detection factor (*HTDF*), helper team time factor (*HTTF*), simultaneous primary team detection factor (*STDF*), and simultaneous primary team time factor (*STTF*) must be specified. These four factors help to describe the influence multiple adversary teams have on one another and they must be specified at the outset of the problem. They are discussed in detail in Section 5.1.2.

The following section discusses how these inputs are translated into a mathematical problem formulation.

3.2 Problem Formulation

For the remainder of this study, $G = (O , A)$ denotes a directed network with node set O and arc set A. Arcs are referred to by a single index, i or j, where $i,j \in A$. Throughout, s and t (with $s \neq t$) indicates origin and destination, respectively. With regards to safeguards, S represents the set of safeguard types, l.

A methodology is needed to simultaneously optimize *Cost* and P_E objectives since they are both considered important measures of system effectiveness and their associated design objectives are conflicting. In order to combine the metrics present in the facility protection problem (cost, in dollars and P_E, in probability), the objectives must be non-dimensionalized. As mentioned earlier (see Section 2.1.2), probability of failure, $P_{FS} = 1 - P_E$ is used here, so that cost and performance are both minimization objectives. Then, Cost and P_{FS} can be combined as follows:

$$Utility_{PPS_n} = 1 - \sqrt{\left(\frac{Cost_n - \min\limits_N(Cost_n)}{\max\limits_N(Cost_n) - \min\limits_N(Cost_n)} \right)^2 + \left(\frac{P_{FS_n} - \min\limits_N(P_{FS_n})}{\max\limits_N(P_{FS_n}) - \min\limits_N(P_{FS_n})} \right)^2} \qquad \textbf{(3-1)}$$

where n is an index representing the current *Cost* and P_{FS} values, and N is the set of all possible n's (i.e. Pareto points). This equation calculates a non-dimensionalized distance from the origin for both *Cost* and P_{FS} (the expression beneath the square root) such that the point closest to the origin represents the optimal value of $Utility_{PPS}$ (see Figure 3-5, where P_{FS}* and *Cost** represent non-dimensionalized P_{FS} and *Cost*, respectively) and then subtracts it from unity so that the above objective is a maximization objective. This combined metric for system effectiveness will be utilized throughout this study.

The general optimization problem to be solved in this study, then, is to maximize $Utility_{PPS_n}$, as defined in Eq. (3-1), as:

$$\begin{aligned} &\max_{n \in N} Utility_{PPS_n} \\ &\text{s.t.} \min_{n \in N}(Cost_n) \le Cost \le \max_{n \in N}(Cost_n) \\ &\text{s.t.} \min_{n \in N}(P_{FS_n}) \le P_{FS} \le \max_{n \in N}(P_{FS_n}) \end{aligned} \qquad \textbf{(3-2)}$$

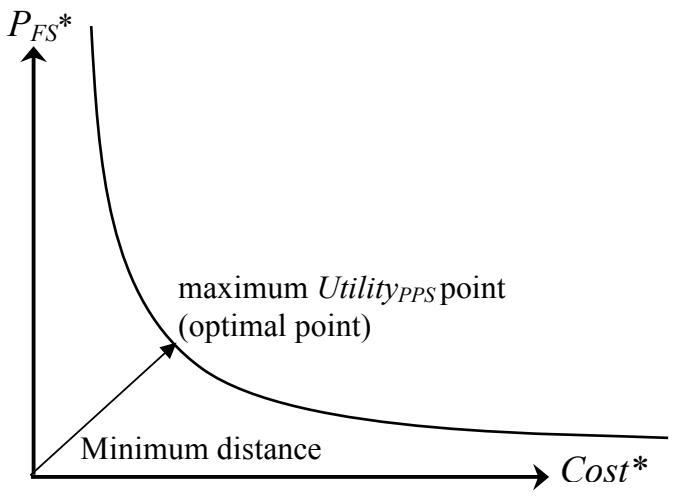


Figure 3-5: Illustration of maximum $Utility_{PPS}$ point

Considering the stochasticity in variables such as time to traverse each arc, guard response time, detection probabilities of the safeguards, delay time increases as a result of the safeguards, and the location in the path where the adversary is detected, the problem in Eq. (3-2) may be re-stated probabilistically as

$$\begin{aligned} &\max_{n\in N} E[Utility_{PPS_n}] \\ &\text{s.t.} \min_{n\in N}(Cost_n) \le Cost \le \max_{n\in N}(Cost_n) \\ &\text{s.t.} \min_{n\in N} E[P_{FS_n}] \le E[P_{FS}] \le \max_{n\in N} E[P_{FS_n}] \end{aligned} \quad \textbf{(3-3)}$$

where the performance constraint and objective are no longer deterministic; that is, they now include stochastic behavior, and cost is assumed to be deterministic. Assuming cost is deterministic and linear is a limitation of this methodology as discounts may apply for purchasing multiple safeguards of a particular type together. Additionally, costs will vary over time as safeguards deteriorate and require additional maintenance.

The problem in Eq. (3-3) looks similar to the problems solved in the field of reliability-based design optimization (RBDO) for mechanical systems. Research in such

problems over the past two decades has resulted in several techniques for probabilistic analysis and optimization that are much more efficient than the Monte Carlo-based optimization methods typically used in the stochastic network interdiction literature. Therefore, this study combines the concepts of network interdiction with reliability-based design optimization techniques, to allow for a PPS design which efficiently and effectively incorporates uncertainty.

Further discussion of the calculation of P_{FS} and *Cost* is provided in the next section. A discussion of uncertainty analysis techniques is provided in Section 3.4, while a complete solution methodology to solve this formulation is discussed in detail in Chapter IV. The following section discusses the calculation of *Cost* and P_{FS}, quantities that occur in the objective and constraints in Eqs. (3-2) and (3-3).

3.2.1 Calculation of P_{FS} and *Cost*

Cost is defined as:

$$Cost = \sum_{l \in S} C_l \sum_{i \in A} y_{li} \tag{3-4}$$

where C_l is the cost of l^{th} safeguard and y_{li} is a binary indicator variable indicating whether or not a safeguard of type l is installed on arc i (1 if installed, 0 if not).

Calculation of P_{FS} follows the following general outline:

1. Calculate detection probability.
2. Calculate guard response time.
3. Calculate probability of timely detection, which is conditional on detection because timely detection cannot occur unless detection occurs first.

4. Calculate probability of adversary neutralization which is conditional on timely detection because neutralization cannot occur unless timely detection has occurred. Thus, the adversary cannot be neutralized if he or she is not first caught (detected in a timely manner) and this cannot happen unless his or her presence in the facility is first detected.
5. Calculate P_{FS} as the complement of P_E, which is further defined as the sum of the per-arc product of detection, timely detection, and neutralization[40]. Thus, P_{FS} for path k is defined as:

$$P_{FS_k} = 1 - P_{E_k} = 1 - \sum_{i=1}^{n} \left[P_{D_{ik}} P_{TD_k|D_{ik}} P_{N_k|TD_k} x_{ik} \right] \quad \textbf{(3-5)}$$

where P_{FS_k} is the probability of security system failure for the k^{th} path, P_{E_k} is the probability of system effectiveness for the k^{th} path, n is the number of arcs on the path, $P_{D_{ik}}$ is the detection probability across arc i on the k^{th} path, $P_{TD_k|D_{ik}}$ is the timely detection probability on path k given that detection at arc i on path k (D_{ik}) occurs, $P_{N_k|TD_k}$ is the neutralization probability on path k given that timely detection on path k (TD_k) occurs, x_{ik} is a binary variable indicating whether or not flow is traveling across arc i on path k.

The following sections discuss details of calculating the quantities in Eq. (3-5).

3.2.1.1 Probability of Detection

The formulation used to calculate $P_{D_{ik}}$ is based on the Estimate of Adversary Sequence Interruption formulation created by Bennett[10] and developed in Garcia[40]. The formula calculates the total detection probability due to the uninterdicted detection probability and the installed safeguards on a particular arc. Then, the cumulative probability of detection of all arcs leading up to the current arc is calculated. The current

total detection probability is multiplied by the cumulative probability of non-detection, or 1-P_D, for the previous arcs. The purpose of doing this is to ensure that the total detection probability does not exceed 1. The cumulative probability of detection can be thought of as the probability that the adversary is detected on any of the arcs. For example, if P_D = .5 for arc 1, and .6 for arc 2, the cumulative $P_D = P(D_1 \cup D_2) = 0.5 + 0.6 - (0.5*0.6) = 0.8$. This per-arc value is then normalized by the probability that the adversary has not been detected until the current arc. $P_{D_{ik}}$ is thus defined as:

$$P_{D_{ik}} = [1-(1-P_{D_O})\prod_{l\in S}(1-P_{D_{li}}y_{lik})]\prod_{j=1}^{i-1}(1-P_{D_j}x_{jk}) \quad \textbf{(3-6)}$$

where P_{D_O} is the arc's uninterdicted detection probability, $P_{D_{li}}$ is the detection probability of the l^{th} safeguard on the i^{th} arc, y_{lik} is a binary variable indicating whether or not a safeguard of type l is installed on arc i of path k, P_{D_j} is the probability of detection across arc j, and x_{jk} is a binary variable indicating whether or not flow travels across arc j on path k. The part of Eq. (3-6) that falls within the square brackets corresponds to the current arc. This value is then adjusted by the part of Eq. (3-6) to the right of the square brackets, the probability of non-detection on previous arcs.

P_D values are assumed to be provided by the manufacturer of the individual safeguard as a performance metric for the particular item. These values are readily available for use in analysis.

3.2.1.2 Probability of Timely Detection

$P_{TD_k|P_{D_{ik}}>0}$ is calculated differently depending on whether or not the analysis is deterministic or stochastic. The stochastic formulation of $P_{TD_k|D_{ik}}$ is discussed in Section

3.4.1. Timely detection given detection at an arc includes both the current arc and future arcs the adversary will travel on. For example, timely detection at arc 1 includes both the timely detection that occurs at arc 1 plus any potential downstream timely detection that may occur further in the adversary's path. $P_{TD_k|D_{ik}}$, then, can be thought of as the complement of the non-timely detection probabilities, as follows:

$$P_{TD_k|D_{ik}} = 1 - \prod_{j=i}^{n} (1 - P_{TD_{jk}|D_i}) \quad \textbf{(3-7)}$$

where $P_{TD_{jk}|D_i} = 1 \ni g_{jk} \le 0, 0$ otherwise

where $g_{jk} = T_{G_{jk}} - \sum_{m=i+1}^{j} [T_m + \sum_{l \in S} T_l y_{lmk}] + \theta \sum_{m=i}^{i} [T_m + \sum_{l \in S} T_l y_{lmk}]$

where g_{jk} may be referred to as the limit state function for timely detection on the j^{th} arc on the k^{th} path (The limit state function is formulated such that $g_{jk} < 0$ indicates failure (for the adversary), $g_{jk} > 0$ indicates success, and $g_{jk} = 0$, the boundary between failure and success is referred to as the limit state. $\theta = 1$ indicates that detection occurs at the beginning of the task and $\theta = 0$ indicates that detection occurs at the end of the task.), T_m is the travel time across arc m, $T_{G_{jk}}$ is the minimum guard response time to arc j on path k, and θ is the location in the current task at which detection occurs. This formulation is based on Bennett[10] and Garcia[40], and includes the assumption that guards have an arc that they are assigned to patrol. For this reason, the timely detection probability is calculated on a per-arc basis based on detection at arc i and then aggregated, as in Eq. (3-7).

Throughout this methodology, the term "guard" is used in general terms to refer to any response personnel. That is, a guard could mean an armed, specialized SWAT team or an unarmed on-site security guard. Guards are assumed to be willing to engage in battle with adversaries and these assumptions will be further discussed in the following

section. If desired, differing guard types can be reflected as different safeguard types with specific delay, detection, and neutralization capabilities.

In order to facilitate the analysis in Eq. (3-7) (and in the next section), the analyst must also develop a strategy for dispatching guards throughout the facility to respond to the detection of an adversary. While the approach used is not necessarily the optimal dispatch of guards, it is thought that the split-second decision required of the guards to respond to a facility intrusion would not allow for the guards to allocate themselves optimally. Guards are assigned to a particular arc. Then, a heuristic approach is used to dispatch each guard to the nearest expected adversary location, which is path dependent. For example, if a guard is assigned to arc 4-D in the network shown in Figure 3-4, he or she will remain on 4-D if the adversary attacks through path O-2-4-D. On the other hand, if the adversary attacks through path O-2-3-5-D, the guard must travel to the nearest arc on that path to attempt to neutralize the adversary. Thus, the guard travels to arc 5-D. While this approach may be optimistic as it assumes that the guards can determine where the adversary will travel, it is a first attempt at developing a guard response strategy.

Consequently, guard response time, $T_{G_{ik}}$, must be calculated. $T_{G_{ik}}$ is calculated using the Floyd-Warshall all-pairs shortest path algorithm[3,37,101]. The algorithm uses an adjacency matrix of a weighted, directed graph to determine the shortest path between the guard's stationed location and the assigned arcs in the adversary's path. The weight of each arc represents the travel time across the arc (this is always T_i for the guards, as it is assumed they are not impeded by safeguards). The algorithm computes $T_{G_{ik}}$ for each set of nodes (in this case, the times between the guard's stationed location and all other nodes in the facility), which is defined as the minimum travel time path between them.

This assumes that the guards will not travel throughout the facility to interrupt or neutralize an adversary. While this is a conservative assumption, it is reasonable in order to make the analysis feasible. Additionally, the presence of multiple teams implies that any single team could be seen as a decoy employed by the adversary. If the guard were to leave his or her post to travel to any location within the facility, the post would be vulnerable to attack by subsequent adversaries.

3.2.1.3 Probability of Neutralization

$P_{N_k|TD_k}$ is typically calculated using complicated simulation tools[99], but it can also be estimated using analytical approximations[15]. The approximation used in this analysis is based on mathematical formulas developed by Lanchester[68] for calculating attrition rates of combating military forces. Lanchester determined that the power of an army with widespread attacking capabilities (i.e. it can attack multiple targets at once) is proportional to the square of the number of units in the army. This relationship is reflected in Lanchester's Square Law, defined as follows for combat between guard and adversary forces:

$$G_O^2 - G_F^2 = E(A_O^2 - A_F^2) \quad \textbf{(3-8)}$$

where G_O is the initial number of individuals in the guard force, G_F is the final number of individuals in the guard force, E is the exchange rate of weapon efficiency (relative weaponry/skill level between forces), A_O is initial number of adversaries, and A_F is the final number of adversaries.

Equation (3-8) is reformulated by Brown[15], based on all potential final scenarios of guard and adversary forces that result in adversary forces being completely eliminated. This is the definition of neutralization, as the security force neutralizes the adversary

when the adversary force is depleted completely (or the adversary surrenders). Brown's work assumes that *E* in Eq. (3-8) is 1; that is, the forces are equally capable and matched in weaponry. An example use of this equation is a problem in which both guard and adversary forces start with 2 individuals. Assuming that simultaneous kills cannot occur (a limitation of Brown's work), the end states that result in a victory for the guard team are the guard team with 1 individual and adversary team with 0 or the guard team with 2 individuals and adversary team with 0. An additional limitation of Lanchester and Brown's derivation is the lack of a surrender option. A surrender option is included in the methodology presented here through a weighted sum of the number of adversary forces versus the total number of adversary and guard forces. If the adversary is severely outnumbered, he or she is likely to surrender. If he or she feels that victory is achievable, he or she is more likely to engage in battle. The inclusion of surrender assumes that the adversary is rational. This is a significant assumption and can be eliminated by changing $P_{surrender}$ to 0. $P_{N_{ik}|TD_k}$, then, is calculated as follows:

$$P_{N_{ik}|TD_k} = P_{surrender\,ik} + (1 - P_{surrender_{ik}})P_{G_{ik}A_{ik}} \quad \textbf{(3-9)}$$

$$\text{where } P_{surrender_{ik}} = \frac{G_{ik}}{G_{ik} + A_{ik}}$$

$$\text{where } P_{G_{ik}A_{ik}} = \sum_{j=1}^{G_{ik}} \frac{(-1)^{G_{ik}-j}\, j^{G_{ik}+A_{ik}}}{[(G_{ik} - j)!(A_{ik} + j)!]}$$

$$\text{where } G_{ik} = \sum_{l \in S} y_{lik} I_l$$

where $P_{surrender\,ik}$ is the probability of adversary surrender on arc *i* of path *k* once he or she is interrupted, G_{ik} is the number of guard force individuals on arc *i* of path *k*, A_{ik} is the number of adversaries on arc *i* of path *k*, $P_{G_{ik}A_{ik}}$ is the probability that G_{ik} guards will neutralize A_{ik} adversaries given in Brown's work[15], and I_l is a binary variable indicating

whether or not the l^{th} safeguard type is a guard. Brown's summation[15] ($P_{G_{ik}A_{ik}}$) calculates the various end scenarios of battle between adversary and guard forces that result in the adversary being completely neutralized (i.e. 2 guards, 0 adversaries or 1 guard, 0 adversaries for a two adversary, two guard battle).

Once $P_{N_{ik}|TD_k}$ is obtained for all arcs of a particular path using Eq. (3-9), these values must be combined in order to include the location of detection. The reason for this is that complete neutralization on an arc includes both the current arc and future arcs the adversary may travel on. For example, neutralization at arc 1 includes both the neutralization that occurs at arc 1 plus any potential downstream neutralization that may occur further in the adversary's path. $P_{N_k|TD_k}$, then, can be thought of as the complement of the non-neutralization probabilities, as follows:

$$P_{N_k|TD_k} = 1 - \prod_{j=i}^{n} (1 - P_{N_{ik}|TD_k}) \qquad \textbf{(3-10)}$$

where all variables are as before.

Eq. (3-10) assumes that timely detection and neutralization are calculated independently at every arc. This formulation then combines the values at each arc, as in Eq. (3-5). If timely detection is equal to 0, neutralization does not matter as P_{FS} will equal 1. If neutralization is equal to 0, regardless of the value of timely detection, P_{FS} will equal 1. Since neutralization is conditional on timely detection being greater than 0, the two values can be separated.

The following section discusses the utilization of previous work exploring similar network-based problems to solve this problem formulation.

3.3 Network Interdiction

Facility protection optimization (FPO) is a process by which a facility "network" is disrupted by increasing the path length or lowering the capacity across an arc in the facility in one form or another. An analogous form of the FPO problem, network interdiction, has been studied in detail in previous literature. Network interdiction performs the same actions as FPO intends to perform, except for an entire network such as a communications network or an interstate system. The following sections discuss deterministic and stochastic network interdiction literature, respectively.

3.3.1 Deterministic Network Interdiction

Deterministic network interdiction has been studied by McMasters and Mustin[77], Phillips[87], and Wood[104], focusing on military applications, and on the interdiction of illegal drugs and their contributing chemicals. The maximum flow approach to network interdiction is often used for problems such as drug networks, where the interdictor, presumably a government body, is trying to interrupt the flow of drugs through a network by creating choke points in the network which severely decrease the allowable material flow through the network.

The optimization of $Utility_{PPS}$ (as defined in Eq. (3-2)) is analogous to the problem of maximizing the travel time of the shortest path, which has been addressed as a deterministic problem in detail in Fulkerson and Harding[38], Golden[46], Israeli[59], and Israeli and Wood[60]. In this study, the goal of the interdictor (facility owner) is to increase the $Utility_{PPS}$ to as large a value as possible by interdicting arcs in the network, potentially increasing their travel times (with barriers) and detection probabilities (with detectors).

This interdiction seeks to deter the adversary from attempting to attack the facility or to ensure he or she is interrupted and neutralized if he or she chooses to attack. This concept was discussed in more detail in Section 1.3.

3.3.2 Stochastic Network Interdiction

It is necessary to include stochastic elements in facility protection optimization in order to perform a real-world analysis. The stochastic elements considered in this study are the times to traverse each arc, the probabilities of successful performance of various safeguards, uninterdicted arc detection probabilities, guard response time, location (within an arc) of the adversary's detection (θ), and the travel time increases as a result of the safeguards. Safeguard costs are assumed to be deterministic (life cycle costs) in order to simplify the problem analysis.

Stochastic network interdiction is addressed in Cormican, Morton, and Wood[26] and Sanchez and Wood[94] with an objective of minimizing the maximum flow through the network. Pan, Charlton, and Morton[81] address the objective of minimizing the adversary's maximum probability of success in traversing the network. Israeli[59] explores the objective of maximizing the shortest path.

The traditional approach to solving non-deterministic network interdiction problems has been to use a two-stage stochastic program with recourse. The two-stage stochastic programming approach, developed by Birge and Louveaux[11], requires discretization and decomposition, numerical simulation, or enumeration in order to include the randomness in the problem. In this type of problem, a first-stage decision (e.g. safeguard location and types) is made to optimize an objective (e.g. maximize the

adversary's shortest travel time), taking into account the second stage decision (e.g. the adversary's choice of paths). All of the variables within the problem involve some level of uncertainty and should be addressed accordingly. Sanchez and Wood[94] use Monte Carlo sampling to screen and select the best critical paths (e.g. shortest paths). The random element in their model is whether or not the attempted interdiction succeeds. Cormican, Morton, and Wood[26] explore the use of both interdiction success and arc capacities as random parameters, assigning them a number of finite values. Pan, Charlton, and Morton[81] handle the adversary's chosen origin and destination pair as an unknown event. The approach used in this study is significantly different than the expected value approach used by Israeli[59] or the typical sequential approach used by Cormican, Morton, and Wood[26] and Sanchez and Wood[94]. These two approaches use a sequential approximation algorithm combined with Monte Carlo simulation to create bounds on a solution and not an exact solution. While this approach is valid, it requires significant computational effort in order to achieve accurate results. Convergence of Monte Carlo to an accurate result usually requires at least several thousand iterations (for realistic failure probabilities of the order < 0.001) until the distribution of the random variables begins to emulate the actual distribution of the variables and yields meaningful results.

As mentioned in Section 3.2, the problem formulation given in Eq. (3-3) looks similar to the problems solved in the field of reliability-based design optimization (RBDO) for mechanical systems. (The probabilistic constraint in Eq. (3-3) is referred to as a reliability constraint in mechanical systems design). Research in such problems over the past two decades has resulted in several techniques for probabilistic analysis and

optimization that are much more efficient than the Monte Carlo-based optimization methods typically used in the stochastic network interdiction literature. Therefore, this study combines the concepts of network interdiction with reliability-based design optimization and first order reliability method (FORM) techniques, to allow for a FPO methodology which efficiently and effectively incorporates uncertainty. The next section briefly reviews reliability analysis and reliability-based optimization techniques, both of which are then used in subsequent sections to solve the formulation in Eq. (3-3).

3.4 Uncertainty Analysis

Reliability-based design optimization (RBDO) is concerned with finding a set of design variables for a given engineering system such that a given objective function is optimized (e.g., minimum cost, minimum weight, etc.) and the design requirements (strength, durability, etc.) are satisfied with high probability. As mentioned in Section 3.2 and the previous section, the problem formulation for FPO in Eq. (3-3) is similar to reliability-based design optimization (RBDO).

There are two steps in solving Eq. (3-3). Step 1 is reliability analysis, i.e., evaluation of the probability constraint. Step 2 is optimization. Step 1 is discussed in the following section, focusing on a first-order approximation to calculate the probabilistic constraint in Eq. (3-3). Methods under step 2 are reviewed in Section 3.4.2, and the approach utilized in this study is discussed in detail in Chapter IV.

3.4.1 Reliability Analysis

Analytical calculation of probability of timely detection, $P_{TD_k|D_{ik}}$, (refer to Eq. (3-7)) which for the remainder of this section is denoted as P_{TD} for notational simplicity, requires the evaluation of the integral of the joint probability density function (pdf) of all the random variables over the failure domain, as:

$$P_{TD} = P(g_{jk}(d,x) \le 0) = \int_{g_{jk}(d,x)\le 0} f_x(x)dx \qquad \textbf{(3-11)}$$

where g_{jk} is as defined in Eq. (3-7). The failure domain can be thought of as the region in which timely detection occurs, so timely detection in this sense is a failure event. This integral poses computational hurdles as it can be difficult to formulate the joint probability density explicitly and integration of a multidimensional integral may be difficult. Therefore, numerical integration methods such as Monte Carlo simulation or analytical approximations such as first-order reliability method (FORM) or second-order methods (such as the second-order reliability method, SORM) are commonly used in mechanical systems reliability analysis. Monte Carlo simulation requires multiple runs of the deterministic system analysis and can be very costly. On the other hand, analytical approximations such as FORM and SORM are very efficient, and have been shown to provide reasonably accurate estimates of the probability integral for numerous applications in mechanical and structural systems. Detailed descriptions of these methods and computational issues are provided in Ang and Tang[5], Haldar and Mahadevan[52], and Ditlevsen and Madsen[31].

In FORM, the variables, x, which may each be of a different probability distribution, and may be correlated, are first translated to equivalent uncorrelated standard normal variables u. For uncorrelated normal variables, this transformation is

simply $u_i = \frac{x_i - \mu_i}{\sigma_i}$. (Later, this concept is expanded to include variables that are non-normal and/or correlated). The limit state and the failure and safe regions are shown in Figure 3-6, in the equivalent uncorrelated standard normal space *u*.

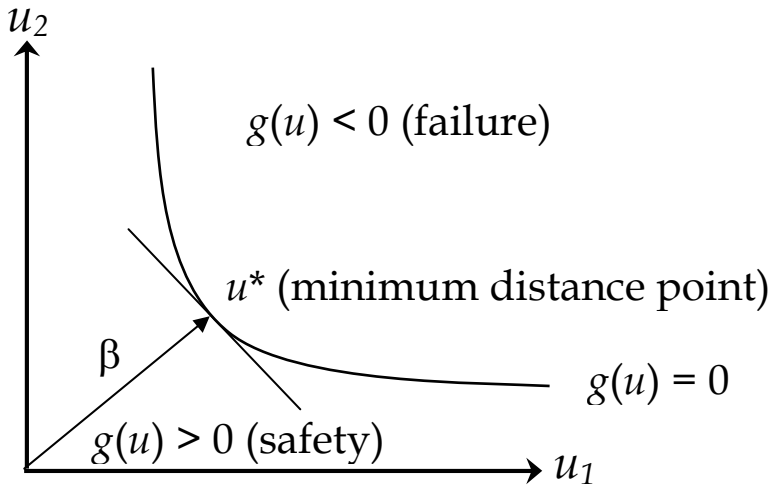


Figure 3-6: Illustration of limit state and failure and safe regions

The failure probability is now the integral of the joint normal pdf over the failure region. The FORM replaces the nonlinear boundary $g = 0$ with a linear approximation, at the closest point to the origin, and calculates the P_{TD} as follows:

$$P_{TD} = P(\, g_{jk}(d,x) \leq 0\,) = \Phi\,(-\beta(d,u)) \qquad \textbf{(3-12)}$$

where Φ is the cumulative distribution function (CDF) of a standard normal variable and $\beta(d,u)$ is the minimum distance from the origin to the limit state. Thus, the multidimensional integral of the joint pdf is now approximated with a single dimensional integral as in Eq. (3-12), the argument of which (i.e., $\beta(d,u)$) is calculated from a

minimum distance search. The minimum distance point on the limit state is also referred to as the most probable point (MPP), since linear approximation at this point gives the highest estimate of the failure probability as opposed to linearization at any other point on the limit state. (A second-order approximation of the failure boundary is referred to as SORM[14,66,98], where the failure probability calculation also requires curvatures of the limit state).

The minimum distance point (or MPP) u^* is found as the solution to the problem:

$$\begin{aligned} &\min \beta(d,u) \\ &\text{s.t. } g_{jk}(d,x) \le 0 \end{aligned} \quad \textbf{(3-13)}$$

A Newton-based method to solve Eq. (3-13) was suggested by Rackwitz and Fiessler[90]. Other methods such as sequential quadratic programming[70] (SQP) have also been used in the literature[32,107].

The minimum distance point may also be found using a dual formulation of Eq. (3-13) as

$$\begin{aligned} &\min g_{jk}(d,x) \\ &\text{s.t. } \| u \| = \beta_t \end{aligned} \quad \textbf{(3-14)}$$

where all variables are as before. This dual problem may be referred to as inverse FORM.

For non-normal variables, the transformation to uncorrelated standard normal space is $u_i = \dfrac{x_i - \mu_i^N}{\sigma_i^N}$, where μ_x^N and σ_x^N are the equivalent normal mean and standard deviation, respectively, of the x variables at each iteration during the minimum distance search. Rackwitz and Fiessler[90] suggested the solution of μ_x^N and σ_x^N by matching the

PDF and CDF of the original variable and the equivalent normal variable at the iteration point. Other transformations are also available[73,92].

If the variables are correlated, then the equivalent normal variables are also correlated. In that case, these are transformed to an uncorrelated space through an orthonormal transformation of the correlation matrix of the random variables through eigenvector analysis or a Cholesky factorization[52]. The minimum distance search and first-order or second-order approximation to the probability integral is only carried out in the uncorrelated standard normal space. Further guidance on the overall reliability analysis procedure is provided by Haldar and Mahadevan[52].

The minimum distance search typically involves five to ten evaluations of the limit state (and thus system analysis), and then the probability is evaluated using a simple analytical formula as in Eq. (3-12). Compared to this, Monte Carlo simulation may need thousands of samples if the failure probability is small, thus making Monte Carlo methods prohibitively expensive for solving large scale stochastic optimization problems.

FORM has been found to be very accurate for linear limit states with normal variables. Although the limit state in the evaluation of P_{TD} is nonlinear (although it is close to linear) and the random variables are not normal (although they are truncated normal variables), FORM is found to be of sufficient accuracy in this problem. This assumption is evaluated further in Sections 4.5 and 4.6 as FORM results are compared with Monte Carlo results to show there is no noticeable gain in accuracy to justify the additional computational expense of Monte Carlo simulation.

3.4.2 Reliability-Based Optimization

In many reliability-based optimization studies[25,32,71,97,106,107], a probability constraint has been replaced by a quantile equivalent, i.e., by a minimum distance constraint. In the current problem, this leads to

$$\begin{aligned} &\min E[Utility_{PPS_n}] \\ &\text{s.t. } \beta_i \geq \beta_t \end{aligned} \quad \textbf{(3-15)}$$

where $\beta_t = -\Phi^{-1}(P_t)$, P_t is a target probability value, and β_i is the minimum distance computed from Eq. (3-13). Alternatively, the dual formulation has also been used, based on Eq. (3-14), as

$$\begin{aligned} &\min E[Utility_{PPS_n}] \\ &\text{s.t. } g_{jk}(d,x) \leq 0 \end{aligned} \quad \textbf{(3-16)}$$

where $g_i(d,x)$ is computed from Eq. (3-14).

Since the reliability constraint evaluation itself is an iterative procedure (see Eqs. (3-13) and (3-14)), the number of function evaluations required for reliability-based optimization is considerably larger than deterministic or safety factor-based optimization. A simple nested implementation of RBDO (i.e., reliability analysis iterations nested within optimization iterations, as in Figure 3-7) tremendously increases the computational effort, and as a result, several approaches have been developed to improve the computational efficiency, typically measured in terms of the number of functional evaluations required to reach a solution for RBDO methods. In decoupled methods[32,93,107] the reliability analysis iterations and the optimization iterations are executed sequentially, instead of in a nested manner (refer to Figure 3-8, where OL represents optimization loop and RL represents reliability loop). This is done by fixing

the results of one analysis while performing the iterations of the other analysis. Single loop methods[67,71,98] perform the optimization through an equivalent deterministic formulation which replaces the reliability analysis constraint with the equivalent Karush-Kuhn-Tucker conditions at the minimum distance point on the limit state. Several versions of decoupled and single loop methods have been developed, based on whether direct or inverse FORM is used for the reliability analysis step. Note that FORM is used in all of these efficient RBDO techniques, although the method developed by Zou and Mahadevan[107] does not require FORM. Further information on the various RBDO formulations is provided in Chiralaksanakul and Mahadevan[24]. Further discussion on the application of the chosen RBDO methodology is provided in Section 4.3.

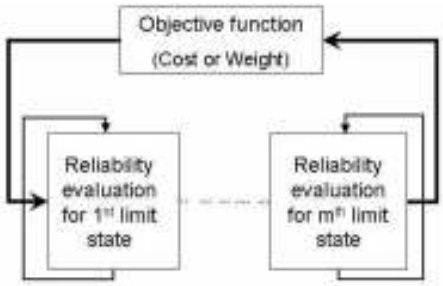


Figure 3-7: Illustration of Nested RBDO Method

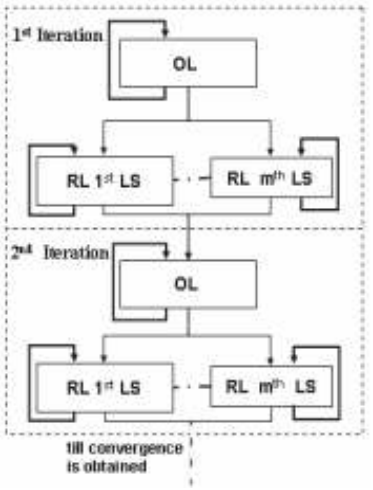


Figure 3-8: Illustration of Decoupled RBDO with Inverse FORM

3.5 Summary

This chapter presented the general problem framework of the facility protection optimization problem, utilizing a network representation. A complete problem statement was introduced utilizing the metric of $Utility_{PPS}$, which combines *Cost* and P_{FS}. Given computationally intensive methods used in previous literature, an efficient method that combines RBDO with FORM was introduced to handle uncertainty and solve the problem statement efficiently. Chapter IV provides a complete solution methodology for the general problem framework presented in this chapter for solving single adversary team problems.

CHAPTER IV

SINGLE ADVERSARY TEAM METHODOLOGY

This chapter pursues the second objective of this study, i.e. to develop an approach to solve the problem formulated in Chapter III in order to handle a single adversary team attack. The following simplifications and assumptions have been made in developing the solution approach:

- The adversary is assumed to be attacking the facility as a single entity. An extension of this approach to address multiple teams is provided in Chapter V.
- The adversary is assumed to be an outsider. Refer to Section 2.1.2.1 for an overview of outsider characteristics.
- All random variables are assumed to be uncorrelated, rather than assuming values of correlation for the problem analysis, which would be more controversial than eliminating correlation entirely. Correlation can be incorporated without a great deal of additional work.
- Response force actions are simplified by assuming the guards are assigned to a particular patrol area, as discussed in Section 3.2.1.3. Recall that guards are stationed on a given arc and they are allowed to respond to nearby arcs when the adversary's strategy necessitates this.
- The calculation of P_E is through a first-order approximation, as in Section 3.4.1.

- The connectivity diagrams for the example problems are location-based, as discussed in Section 3.1. The diagrams represent the physical layout of the facility.
- Cost is assumed to be deterministic and linear. That is, one value is used for the cost of each safeguard (assumed to be life cycle cost) and this cost per safeguard does not change regardless of how many of a particular safeguard are purchased. That is to say, there are no discounts for safeguard bundles.
- It is assumed throughout this methodology that only one of each safeguard type can be installed on each arc. This constraint is enforced by restricting variables to binary variables, and it can be removed and the values can be restricted to higher integer (or non-integer) values if necessary.
- It is assumed that sabotage requires access to the actual target. This assumption will result in a different security system than if the adversary was able to sabotage the target from a distance. This surrounding area control may be explored in future work.

Section 4.1 discusses the identification of set of critical paths for inclusion into the optimization, while Section 4.2 addresses safeguards optimization. Section 4.3 then discusses the chosen optimization strategy. Section 4.4 follows with a discussion of method efficiency. Sections 4.5 and 4.6 apply the developed methodology to explore new facility design, upgrades analysis, and existing facility analysis for two example problems. Section 4.7 demonstrates the methodology on a simple practical example. The chapter concludes in Section 4.8 with some remarks about the methods.

4.1 Determine Critical Path Set

Identification of the critical path set is necessary in order to determine which paths in the facility are the most vulnerable (and subsequently defend these paths with optimal safeguards placement), as discussed in Section 1.3 and Chapter II. The critical path set is determined directly from the connectivity diagram and the parameters identified in Section 3.1 and it is based on the concept of determining which paths in the facility are the most vulnerable.

One approach to determining the critical path set is to simply enumerate *feasible paths* through the facility. This approach ignores the objective being considered and merely attempts to find paths which satisfy the arc-node incidence matrix and node balance constraints (as discussed in Section 3.1.4). This approach involves solving a linear system of equations and therefore, is computationally very efficient. The drawback, however, is that this approach often enumerates more paths than are necessary for inclusion into the safeguards analysis. As a result, it can be computationally inefficient for larger problems.

Alternatively, *optimal path selection* can be utilized to determine the critical path set. First, the path with the optimal value of P_E is calculated (in terms of the adversary's viewpoint, since the adversary is only concerned with defeating the PPS, his or her goal is based entirely on the path with the lowest P_E).

If the analysis is intended to evaluate the current PPS, the optimal path selection stops with the adversary's optimal path identification as the facility owner is only concerned about the worst case adversary path (as this is the likely path of attack, based on the discussion in Section 1.3). If however, an upgrades analysis or new system design

is being performed, the next k best *unique* paths without repeated arcs through the facility are calculated. More information on k-shortest paths problem formulations is provided in a survey by Eppstein[33]. Uniqueness in this case means that any path cannot contain all the arcs of a previously identified path. The path identification continues until the P_E of the k^{th} best unique path is above some predefined threshold or until the budget for safeguards is exhausted (in the case of an upgrades analysis). This set of paths is referred to as the *critical path set* (CPS) and it is used during the safeguards analysis. In the absence of constraints (as in feasible path selection), the CPS would be equivalent to the set of all unique paths without repeated arcs in the facility. Details of this process are shown in Section 4.1.2, where a formal definition of CPS is provided.

The importance of analyzing all paths below a threshold value is to ensure that the interdictor does not leave a vulnerable path, that is, an uninterdicted route through the network which the adversary can exploit to his advantage. The reason for including only unique paths is that any additional path would be redundant and represent objective levels above those of the critical paths. For instance, the critical paths for the simple location-based undirected CD shown in Figure 4-1 (traveling from A-F, where values shown are travel times) are *ABCEF* and *ABDEF*. Any additional paths would be unnecessary. Path *ABCEDBCEF* is a valid path to travel from *A* to *F*, but it is not a unique path in that it contains all the arcs found in the path *ABCEF*. Since the longer path contains all arcs which are present in *ABCEF*, any safeguards applied to the shorter path in turn affects any longer paths as well, and it is therefore unnecessary to include this non-unique path in the analysis. Additionally, *ABCEDBCEF* contains the repeated arcs of *BC* and *CE*. For these reasons, path *ABCEDBCEF* is not included in the safeguards optimization.

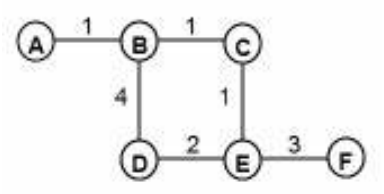


Figure 4-1: Located-Based Connectivity Diagram #2

As both the feasible path selection and optimal path selection approaches to critical path enumeration are new, previous critical selection work is addressed in the following section.

4.1.1 Previous Critical Path Selection Research

Several approaches are available in the literature for solving network interdiction problems but they are either computationally expensive or result in redundant paths which are unnecessary for inclusion in the analysis. Sanchez and Wood[94] develop an algorithm called BEST which enumerates all critical paths with an objective value that falls within predefined bounds. This could result in redundant paths which are unnecessary for inclusion into the analysis in all but a completely directed network. Cormican, Morton, and Wood[26] enumerate all critical paths, further increasing the computational effort of the approach that Sanchez and Wood employ. Enumerating only the unique paths without repeated arcs (subject to budget and performance constraints) would save computational effort since the unnecessary paths are not included in the safeguards optimization.

Since previous techniques tend to enumerate a larger number of paths, these methodologies tend to focus effort on efficiently performing the interdiction analysis

through decomposition techniques (Israeli[59]; Israeli and Wood[60]), whereas the focus in the methodology proposed in this study is on restricting the number of paths analyzed and decoupling the critical path enumeration from the safeguards optimization. As the safeguards optimization is a costly integer mathematical program, reformulating the problem so that it only needs to be solved once saves significantly on computational effort. Further discussion of the computational efficiency of this approach is provided in Section 4.4.

The approach followed in this methodology is to utilize feasible path analysis for simple problems and to use optimal path analysis for more complicated problems. If the network is simple, as in Figure 4-1, feasible path analysis can be undertaken to save computational effort. If the network is more complicated, as is often the case with real world problems, then computer-based optimization techniques must be used to determine the CPS. A detailed mathematical formulation of critical path selection follows.

4.1.2 Mathematical Formulation of Critical Path Selection

If the analysis of the network does not require *optimal path selection* (i.e. if the problem is not large scale), then *feasible path selection* is advised. Feasible path selection is of the form:

$$\text{Find } x_{ik} \in X \tag{4-1}$$

$$s.t. \sum_{out} x_{ik} - \sum_{in} x_{ik} = \begin{cases} 1 \text{ if } x \text{ is a supply node} \\ -1 \text{ if } x \text{ is a demand node} \\ 0 \text{ for all intermediate nodes} \end{cases}$$

$$s.t. \sum_{i \in A} x_{ik} < Card_k \forall k = 1,2..k-1$$

$$x_{ik}(x_{ik}-1)=0 \;\forall\, i$$

where x_{ik} is a binary variable indicating whether or not flow is traveling across arc i on path k, X is the set of all feasible x_{ik}'s, i is an index representing arcs in the path, and $Card_k$ is the cardinality of the set of arcs in the k^{th} critical path.

The first constraint in Eq. (4-1) represents standard flow balance constraints for network problems. They ensure that any flow (in this case, the adversary's path) that enters the network must exit it and the flow balance at all intermediate nodes (non-origins and destinations) is zero.

The second constraint in Eq. (4-1) ensures path uniqueness by forcing the sum of the x_{ik}'s to be less than the cardinality for the paths of the previous iterations. For example, the shortest path for the network in Figure 4-1 (if the values shown are delay times) is *ABCEF*. To find the 2nd shortest unique path, a constraint is added which says: $x_{AB,1} + x_{BC,1} + x_{CE,1} + x_{EF,1} < 4$. This formulation finds the shortest path through the network that does not include the path of *AB-BC-CE-EF*.

The final constraint in Eq. (4-1) ensures that paths with repeated arcs are prevented. This constraint restricts flow values to either 0 (no flow) or 1 (single flow). In order for travel across a particular arc to be repeated, the flow would have to be allowed to be an integer value above one.

The overall algorithm to find the critical path set (CPS) using feasible path selection, then, is as follows:

Step 0: Initialize CPS = 0.

Step 1: Solve for set of feasible x_{ik}'s using Eq. (4-1). Add solution to critical path set.

Step 2: Repeat Step 1 until infeasible (no feasible paths remain).

If, however, *optimal path selection* is necessary, then the procedure for identifying the CPS is more involved. In this step, the analyst is looking for the most vulnerable set of paths, defined in this case as those with the lowest P_E. This begins by evaluating the minimum P_E path through the network and continues to find the next k best unique paths through the facility. If either an upgrades analysis or new system design is being undertaken, a complete CPS must be calculated through the following formulation:

$$\min_{k} P_{E\,k} = \sum_{i=1}^{n} \left[P_{D_{ik}} P_{TD_k|D_{ik}} P_{N_k|TD_k} x_{ik} \right] \tag{4-2}$$

$$s.t. \sum_{out} x_{ik} - \sum_{in} x_{ik} = \begin{cases} 1 \text{ if } x \text{ is a supply node} \\ -1 \text{ if } x \text{ is a demand node} \\ 0 \text{ for all intermediate nodes} \end{cases}$$

$$s.t. \sum_{i \in A} x_{ik} < Card_k \forall k = 1,2..k-1$$

$$x_{ik}(x_{ik} - 1) = 0 \; \forall \; i$$

$$s.t. \; P_{E_k} \le P_{E_{\text{limit}}}$$

$$s.t. \; MCS_k \min_{l}(C_l) \le budget \; \forall \; l$$

where $P_{E\,\text{limit}}$ is the minimum acceptable P_E for the paths, MCS_k is the minimum cut set of safeguards for all the enumerated paths (that is, the minimum number of safeguards required to interdict each path in the CPS at least once), and all else is as before. Details of calculating $P_{E\,k}$ are provided in Section 3.2.1.

The first, second, and third constraints are identical to those in Eq. (4-1).

The fourth and fifth constraints in Eq. (4-2) represent performance and budget constraints, respectively. They determine a stopping point for the CPS enumeration based on performance limits and the available budget. If the analyst has a performance limit above which he or she feels path enumeration is no longer necessary, the fourth constraint terminates the CPS enumeration. If the analyst has a budget for additional safeguard expenditures ($0 in the case of an existing facility analysis, an analyst-defined amount if an upgrades analysis is being undertaken, or total cost of complete safeguards enumeration if a complete new system design is being undertaken), the fifth constraint allows for enumeration of paths until each path cannot be interdicted with at least one safeguard. In other words, if the paths all share one common arc, then the minimum cut set is one and therefore interdiction only requires a budget of one safeguard. If, however, there are no shared arcs among two paths, then the minimum cut set is two and the budget must allow for installation of two safeguards or the CPS enumeration will terminate.

The deterministic problem formulation of Eq. (4-2) can be modified to a stochastic formulation as:

$$\min \mathrm{E}(P_{E_k}) = \sum_{i=1}^{n} \left[E(P_{D_{ik}}) P_{TD_k|D_{ik}} E(P_{N_k|TD_k}) x_{ik} \right] \forall k = 1,2..N \qquad \textbf{(4-3)}$$

$$s.t. \sum_{out} x_{ik} - \sum_{in} x_{ik} = \begin{cases} 1 \text{ if } x \text{ is a supply node} \\ -1 \text{ if } x \text{ is a demand node} \\ 0 \text{ for all intermediate nodes} \end{cases}$$

$$s.t. \ \mathrm{E}(P_{E_k}) \le P_{E_{\text{limit}}}$$

$$s.t. \ MCS_k C_l \le budget \ \forall \ l$$

$$s.t. \sum_{i \in A} x_{ik} < Card_k \forall k = 1,2..k-1$$

$$x_{ik}(x_{ik}-1)=0 \;\forall\, i$$

where $P_{TD_k|D_{ik}} = P(T_{G_j} - \sum_{m=i+1}^{j}(T_m + \sum_{l\in S} T_l y_{lm}) + \theta\sum_{m=i}(T_m + \sum_{l\in S} T_l y_{lm}) \le 0)$ and is calculated as described in Section 3.4.1, and all other variables are as before. Eq. (4-1) can be reformulated in a manner similar to Eq. (4-3) if desired.

The major difference between the formulations of Eqs. (4-2) and (4-3) lies in the evaluation of the objective. In Eq. (4-2), all values in the equation are considered deterministic.

The overall algorithm to find the critical path set (CPS) using optimal path selection, then, is as follows:

Step 0: Initialize CPS = 0.

Step 1: Solve for set of x_{ik}'s using Eq. (4-3). Add solution to critical path set.

Step 2: Repeat Step 1 until either the 4th or 5th constraint are violated or no feasible paths remain.

Once the critical path set is generated (through either feasible path selection or optimal path selection), it is retained for further analysis in the safeguards optimization step, detailed in the next section.

4.2 Design Optimization of the Safeguards System

The purpose of this section is to develop a mathematical model for optimum design of the safeguards system. A multiobjective optimization approach is required to perform this optimization, as discussed in the following section. The safeguards optimization formulation is then discussed.

4.2.1 Multiobjective Optimization

While $Utility_{PPS}$ (refer to Eq. (3-1)) is a single objective, it contains both *Cost* and P_{FS} objectives which must be optimized simultaneously. Since neither *Cost* nor P_{FS} can be expressed explicitly in terms of one another (i.e. $Cost \neq f(P_{FS})$), optimization of $Utility_{PPS}$ requires a multiobjective approach.

The relative importance of both objectives is not known with certainty, otherwise the conflicting objectives can easily be combined into a single objective using a weighted sum. The set of variables that produces the optimal outcome for this type of problem is referred to as the Pareto optimal set[19,20,28,83] and it yields a set of possible answers for the multiobjective optimization. A set of points is said to be Pareto optimal if, in moving from point A to another point B in the set, any improvement in one of the objective functions causes the value of at least one of the other objective functions to worsen. This concept is shown graphically in Figure 4-2. This is, of course, for a function in which both f_1 and f_2 are being minimized. Theoretically, the Pareto optimal set yields an infinite set of solutions that the analyst can chose from. This set reduces to a finite number of points in the event one of the variables can be discretized.

In order to choose from the solutions located on the Pareto optimal curve, an objective must be developed which combines minimization of cost and maximization of P_E into one objective so that the optimal solution can be chosen.

The maximum value of $Utility_{PPS}$ represents the best balance of cost and performance possible in the system. Many approaches are available to generate the set of *Cost* and P_{FS} values which populate the Pareto optimal set. The most popular are weighted sum[20], goal programming[21,57], and ε-constraint[20]. Although the remainder of

this section focuses on the bi-objective problem considered in this study, these techniques may be generalized to problems where more than two objectives are being optimized.

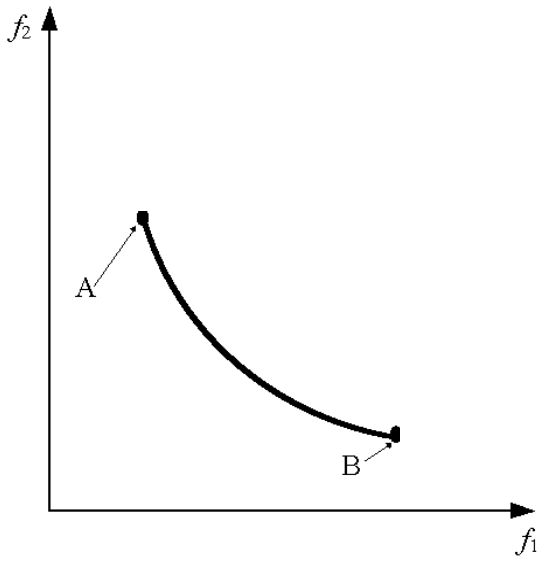


Figure 4-2: Graphical Illustration of the Pareto Optimal Curve[20]

In the weighted sum approach[20] a function with two objectives is combined into a single objective by constructing a weighted sum of all the objectives. The problem can then be optimized using standard optimization techniques. The difficulty arises in assigning weights to the different objective functions. If the combined problem is convex, then a complete set of non-inferior (Pareto) solutions can be found. However, if the problem is not convex, generation of the entire Pareto set is not guaranteed.

In the second method, goal programming[21,57], the analyst must construct a set of goals (realistic or not) that should be attained (if possible) and include all the goals in the objective function through a penalty function. Although this method is simple and easy

to use, the possibility of solutions that are not Pareto efficient and the subjectivity in the penalty formulation have been cited as potential weaknesses[42].

The ε-constraint method[20] does not have a problem with convexity. In this formulation, all but one of the objectives are transformed into constraints and the problem is optimized for the remaining objective as follows:

$$\begin{aligned} &\min_{x} f_r(x) \\ &s.t. \\ &f_i(x) \le \varepsilon_i \forall i = 1,2..n \ne r \end{aligned} \qquad \textbf{(4-4)}$$

where $f_i(x)$ is the i^{th} objective function, ε_i is the limit on the i^{th} constraint.

For a bi-objective problem, this procedure is straightforward as ε_i is set to the minimum feasible value for f_i, the problem is solved, and then ε_i is incrementally increased until it is set to the maximum feasible value for f_i. Theoretically, this method allows the designer to determine the complete Pareto set of optimal points, but only if all possible values of ε_i are used. In order to ensure this, the problem can be solved with P_E as the objective to minimize and the value of *Cost* in the constraint. Since *Cost* can only take on a finite number of discrete values and safeguards decisions are deterministic and binary (i.e. a safeguard cannot be partially installed), the complete Pareto set is generated.

In order for a true representation of the decision-maker's preferences to be taken into account, the decision-maker must create a complete table of his or her preferences and satisfaction levels for a range of possible objective value combinations. A multi-objective optimization technique must then be able to find a solution which satisfies these preferences. It is assumed for this study that the facility owner is indifferent towards *Cost* and P_E. That is, the facility owner simply wants the best safeguards plan which balances these two objectives optimally. Given this preference, as well as the simplicity

of implementing it since *Cost* is a discrete variable, multiobjective optimization is performed using ε-constraint optimization in this study. The optimization is repeated multiple times with optimization of P_E as the objective and *Cost* as a constraint until all feasible values of *Cost* are explored.

If the computational burden required to develop the entire Pareto set is prohibitive, a novel approach can be taken that reduces this burden. In this limit-constrained multiobjective optimization, the analyst must calculate the end points of the Pareto set, as illustrated by A and B in Figure 4-2 and discussed earlier in this section. These points will provide the $\max_N(Cost_n), \min_N(Cost_n), \max_N(P_{FS\,n}), \text{and} \min_N(P_{FS\,n})$ necessary for evaluating Eq. (3-1). The problem can then be solved as a unconstrained maximization, with Eq. (3-1) as the objective. This formulation will provide only the maximum $Utility_{PPS}$ point and not the entire Pareto set.

The following section discusses incorporating this multiobjective optimization framework into a deterministic and then a stochastic safeguards optimization.

4.2.2 Optimization Formulation

The baseline safeguards optimization problem is deterministic. All variables are assumed static and the problem, therefore, is significantly simpler than its stochastic equivalent. The safeguards optimization is undertaken as a repeated optimization of P_E with increasing cost constraint limits in order to facilitate multiobjective optimization (as outlined in the previous section). The formulation for this problem is very similar to the formulation for optimal path selection (Eq. (4-2)), as follows:

$$\max \min_{k} P_{E_k} \quad \textbf{(4-5)}$$

$$\text{where } P_{E\,k} = \sum_{i=1}^{n} \left[P_{D_{ik}} P_{TD_k|D_{ik}} P_{N_k|TD_k} \right]$$

$$\text{s.t.} \sum_{l \in S} C_l \sum_{i \in A} y_{li} \leq budget$$

where k is the index of the critical paths, and $P_{D_{ik}}$, $P_{TD_k|D_{ik}}$, and $P_{N_k|TD_k}$ are as defined in Eqs. (3-6), (3-7), and (3-10) respectively and the y_{ik}'s present in Eqs. (3-6), (3-7), and (3-10) are the design variables.

The overall algorithm to find the Pareto optimal set, then, is as follows:

Step 0: Initialize *budget* to minimum budget value (0 for new system design, current PPS cost for upgrades analysis or existing system analysis).

Step 1: Solve for set of y_{ik}'s using Eq. (4-5).

Step 2: Compute $Utility_{PPS}$ and store solution, along with set of y_{ik}'s.

Step 3: Increase budget by 1 unit (corresponding to cost of smallest safeguard).

Step 4: Repeat Steps 1-3 until budget is equal to maximum budget.

The set of y_{ik}'s which yields the highest value of $Utility_{PPS}$ is the optimal safeguards configuration. This value should be recommended to facility management as the chosen solution for a PPS configuration. This formulation is limited, however, in that it does not include uncertainty.

The deterministic problem formulation of Eq. (4-5) can be modified to a stochastic formulation as:

$$\max \min_{k} \mathrm{E}(P_{E_k}) = \sum_{i=1}^{n} \left[E(P_{D_{ik}}) P_{TD_k|D_{ik}} E(P_{N_k|TD_k}) \right] \quad \textbf{(4-6)}$$

$$\text{s.t.} \sum_{l \in S} C_l \sum_{i \in A} y_{li} \leq budget$$

where $P_{D_{ik}}$, $P_{TD_k|D_{ik}}$, and $P_{N_k|TD_k}$ are as defined in Eqs. (3-6), (3-12), and (3-10) respectively.

The following section discusses the optimization strategies used to solve both the critical path selection and safeguards optimization problems.

4.3 Optimization Strategy

Since both the critical path selection and safeguards optimization problems are composed of discrete decision variables, branch and bound[16,69], cutting plane methods[49,50,51] and problem reformulations, heuristics[45,47,64,75], and dynamic programming[8,62] were explored as solution algorithms early on in the analysis procedure. These four methods are very useful in solving complicated nonlinear optimization problems, as in these problems. Following is a description of each method.

Given that the optimization problems involved discrete decision variables, branch and bound is a natural choice for optimization. Branch and bound is a general partial enumeration technique which splits the main optimization problem up into smaller solvable subproblems, retaining the best found objective and eliminating the need to explore all possibilities. In the subproblems, constraints restricting the variables to discrete are removed and the problem is solved as a continuous optimization problem. This optimization is repeated while tightening bounds on the variables by constraining individual variable values to integers. Eventually, the branch and bound procedure yields an optimal integer solution, without requiring complete solution enumeration, which, for large real-world problems, is impractical. More information on branch and bound is provided in Brusco and Stahl[16] and Lawler and Wood[69].

Cutting plane methods[49,50,51] and problem reformulations work by adding constraints to linear programs until the optimal basic solution takes on integer values. These methods involve reformulations of the original problem in order to make the overall problem solve more efficiently, as well as enhancements such as Benders' decomposition[9], which efficiently solves complicated problems by decomposing the problem into smaller subproblems and utilizing cuts to shrink the feasible solution space. Cutting plane methods and problem reformulations are prominent in previous network interdiction work[26,60,81,94,104]. However, due to the implicit nature of the objective function utilized in this study, these methods proved unsuccessful.

Heuristic approaches (genetic algorithms, tabu search, and simulated annealing) were explored for optimization due to their potential for solving complicated optimization problems. Genetic algorithms (GA) work by emulating natural selection and evolution to find the optimum solution of an optimization problem. They begin with a random population and evolve over generations into a better solution, based on the fitness of current solutions. More information on GA's is provided by Man, Tang, and Kwong[75] and Goldberg[47]. A GA was tested for solving this problem. Given that the GA returned inconsistent results with inferior objective values than the branch and bound solver (and in some cases, could not converge to a solution), their results are not reported. Mention of them is provided solely to demonstrate what approaches were used to solve this problem. Tabu search[45] (which operates by forbidding or penalizing moves that take the solution to points in the solution space already visited) and simulated annealing[64] (where the objective function is interpreted as the internal energy of a system whose state search space is compared to a physical system; the problem objective is minimization of

internal energy) are two other well-known heuristic methods for obtaining the global optimum of a complicated objective function. Heuristic methods do not guarantee global optimality and they can be difficult to implement when variables are discrete. As a result, tabu search and simulated annealing were not explored further.

Dynamic programming[8] was explored given its usefulness for solving repeated problems (as is the case with the safeguards analysis) through its approach of partial enumeration. The basic idea of dynamic programming can be illustrated by exploring the knapsack optimization problem[62]. In the knapsack problem, an individual wants to maximize the amount of *N* items that can fit in a knapsack with a total weight of *M*. In order to solve this problem via dynamic programming, the problem is solved for all possible weights up to *M*. This can be done rather quickly by using prior solutions to build up to larger solutions. For example, in order to evaluate the solution for a weight of 2, the program needs to look at solutions which build up from a weight of one or zero. In order to get from zero to two, only an item with weight of two can be used. In order to get from one to two, the program uses the solution for a weight of one and then adds an item with a weight of one. As the program continues, this approach becomes more efficient as fewer inferior solutions are explored.

After exploring the different methods for solving the optimization problems, two differing techniques were chosen for the critical path generation and safeguards optimization. For critical path generation, branch and bound was chosen to calculate the individual critical paths, due to its effectiveness in solving integer optimization problems. Dynamic programming was chosen for the safeguards optimization as it involves a repeated optimization process where cost increases incrementally.

The subproblem optimizations within branch and bound were performed in Matlab[29,76] using a combination of routines from the Matlab Optimization toolbox, publicly available code[18,65,96], and author-developed content. The underlying nonlinear optimization routine in Matlab is a sequential quadratic programming (SQP) algorithm[70], which is designed to achieve quick, reliable results for nonlinear problems. At each major iteration, the SQP method chooses an updated search direction by solving a quadratic programming (QP) subproblem. An estimate of the Hessian of the Lagrangian is updated at each iteration using the Broyden-Fletcher-Goldfarb-Shanno formula[36,48]. A line search is performed to determine the new search direction using a merit function similar to those proposed by Han[53] and Powell[88,89]. The QP subproblem is solved using an active set strategy similar to that described in Gill, Murray, and Wright[43]. Further information on the algorithm is provided in the Matlab user's manual[76].

For the purpose of both the critical path set enumeration and the safeguards optimization in this study, a nested method of RBDO was implemented. Since the limit state function in this problem is not linear (although it is close) and involves non-normal variables, a FOSM approach cannot be utilized for reliability analysis. The limit state is nearly linear, however, so FORM can be utilized, thereby resulting in a computationally inexpensive reliability loop of the RBDO process. Therefore, the nested method is sufficient and single loop or decoupled approaches are not necessary. If the problem size or the computational expense of the reliability analysis prevents large scale problems from being solved, it would be advisable to implement a single loop or decoupled approach to the problem.

The next section provides a detailed discussion on the efficiency of the chosen methodology for critical path selection and safeguards optimization.

4.4 Methodology Efficiency

Instead of concentrating effort on decomposing the problem as in previous network interdiction studies[26,60,81,94,104], this methodology decouples the critical path selection and safeguards optimization completely so that the safeguards optimization only needs to be performed once for each cost level. Additionally, decoupling is used in the safeguards optimization to separate the reliability analysis from the optimization procedure. Finally, computational savings are realized through enumeration of a reduced critical path set.

For identifying the k-shortest paths in a network, the best known bound for the number of function evaluations required for a directed graph with m vertices and n arcs is $O(n + m \log m + k)$[33]. When multiple adversary teams are considered, this number is multiplied by the number of teams, as the size of the network grows linearly as the number of teams increases. If the method described in this dissertation is utilized, the number of paths enumerated, k, will be less, as discussed in Section 4.1.1. Therefore, the critical path selection process will require less function evaluations that previously identified methods. The extent of this computational savings will depend on the network being analyzed.

The safeguards optimization requires significant computational effort. For total safeguards enumeration, there are 2^{n*S} potential safeguards plans ($n*S$ represents the total number of arcs, n, multiplied by the total number of safeguards types, S) as each

combination of arc and safeguard type is a binary decision variable indicating whether or not a safeguard of a particular type is installed on a particular arc. Each of these plans must be calculated for each budget level, critical path, and adversary team ($C*k*R$ times, where R is the number of teams, k the number of critical paths, and C, the number of cost levels). Assuming total solution enumeration, the total worst case number of functions required for the complete safeguards optimization, then, is $C*k*R*2^{n*S}$. Using dynamic programming, there are significant computational savings, as the number of function evaluations for the methodology in this study reduces to $\frac{kRS^2nC}{2}$ since the analysis now must be undertaken an average of $\frac{kRSC}{2}$ times at each arc, of which there are $n*S$ arcs. If analysis is performed using the approach in previous methodologies, this complexity increases to $\frac{k^2RS^2nC}{2}$, as the safeguards optimization must be repeated each time a new critical path is generated (k times).

As a result, this methodology results in a reduction of computational effort for both critical path selection and safeguards analysis. Additional savings can be seen if the critical paths are generated via feasible path enumeration, as detailed in Section 4.1.2.

Computational savings are also realized by utilizing reliability-based design optimization with a first-order reliability approach to analyzing uncertainty. This dual layer of decoupling (decoupling of the critical path selection and safeguards optimization; and decoupling of the reliability analysis and safeguards optimization), plus the use of FORM instead of Monte Carlo simulation, results in significant computational savings that allow large scale problems to be solved.

Following are twelve example problems. The first six represent two separate network topologies, each being analyzed as a current facility analysis, an upgrades analysis, and a new facility PPS design. The final six are a demonstration of the use of this methodology on a simple practical example. Following each set of example problems is a table summarizing the CPU times of each method.

4.5 Example Problem 1

The examples presented in this section are based on the simple network shown in Figure 4-3. In this network, the origin is marked with an "O", the destination with a "D" and all other intermediate nodes are numbered in no particular order. All arcs have arrows to indicate the direction of flow across them since this is a directed graph. The example is simple in order to demonstrate the capabilities of the developed methodology.

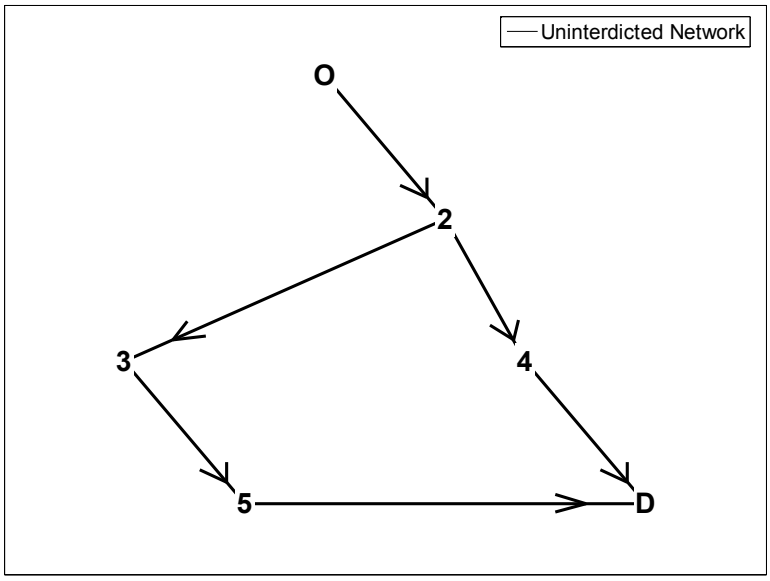


Figure 4-3: Network Graph, Example Problem 1

The input variables for the network are summarized below (quantities are identical for all arcs):

Table 4-1: Variable Statistics, Example Problem 1

Variable Description	*Statistics*
Baseline guard response (T_G)	N(5,1) min.
Uninterdicted travel time (T)	N(1,0.1) min.
Uninterdicted detection probability (P_D)	0
Detection location (θ) *within arc*	N(0.5, 0.05)

There are three types of safeguards at the facility owner's disposal: a barrier, a detector, and a guard. Their properties are summarized below:

Table 4-2: Safeguard Properties, Example Problem 1

No.	*Safeguard Description*	*Delay Time Statistics (mins)*	*Detection Probability Statistics*	*Cost ($K)*
1	*Barrier*	N(1,0.1)	-	5
2	*Detector*	-	N(0.2,0.02)	10
3	*Guard*	N(5,1)	N(0.5,0.05)	100

The stochastic optimization is carried out using both FORM and Monte Carlo analysis.

For Monte Carlo analyses, the analyses began by using 1,000 samples whenever P_{FS} was required, unless otherwise noted. Since Monte Carlo simulation (MCS) is computationally intensive, the intent of keeping the number of samples low is to decrease the resulting computational burden. The problem with this approach, however, is that it does not provide consistent results for complicated problems (i.e. new system design)

when performing dynamic programming. The reason for this is simple. If the number of samples is small, there is a large likelihood that the MCS approach will retain an inferior security system at a low cost value in the analysis (as the inherent variability in MCS means that running the optimization several times yields slightly different results and therefore, differing safeguards plans at lower costs). As a result, more samples are required for solution stability, further increasing the computational requirements of MCS. This inconsistency is eliminated by using a FORM-based approach, since it is an analytical approximation rather than a numerical simulation and it therefore yields identical results every time an optimization is performed. This computational issue becomes magnified as the problem increases in complexity. As a result, MCS-based optimization is considered impractical for all but the simplest of problems due to large computational requirements.

4.5.1 Existing Facility Analysis

An existing facility analysis is useful if a facility owner is trying to decide whether or not his or her facility's current security system provides adequate protection against an adversary threat.

For this example, a random safeguards plan was generated for the network shown in Figure 4-3. The network corresponding to this random safeguards plan is shown in Figure 4-4.

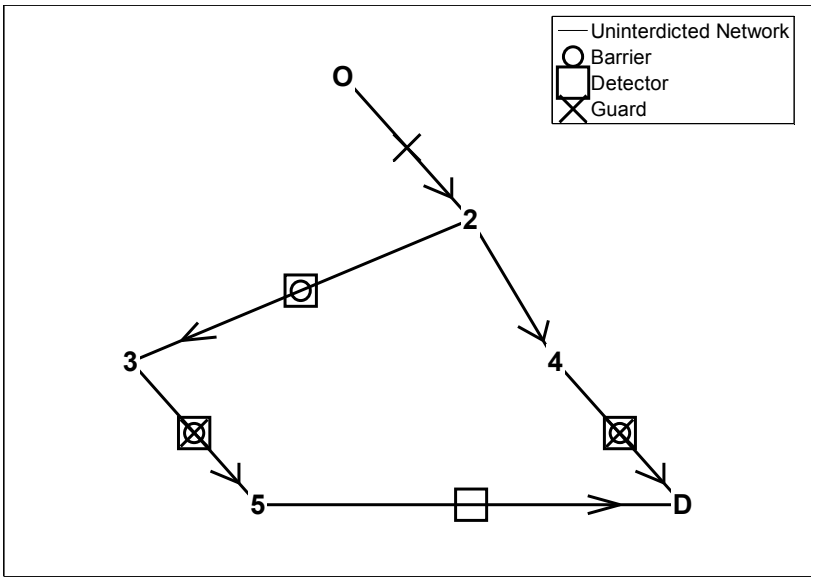


Figure 4-4: Randomly Generated Network, Example Problem 1

The only results that are generated from an existing facility analysis are the *Cost*, $355K, and P_{FS}, .2385, which were identical for both the FORM and Monte Carlo (with 1,000 samples) analyses.

4.5.2 Upgrades Analysis

An upgrades analysis is useful in the case that the facility owner performs an existing facility analysis and decides that the level of facility protection provided by the current security system is inadequate. In this case, the facility owner specifies an allowable budget for upgrades and an upgrades analysis is performed to find out what level of performance improvement can be achieved with a limited budget.

For this example, a safeguards configuration is randomly generated (for illustration purposes) as shown in Figure 4-5. This random safeguards configuration can be taken as the baseline scenario, with an upgrades budget of $250K. All other data are

as before. Figure 4-6 shows the results of the upgrades analysis for both FORM and 1,000 samples of Monte Carlo (their results are identical).

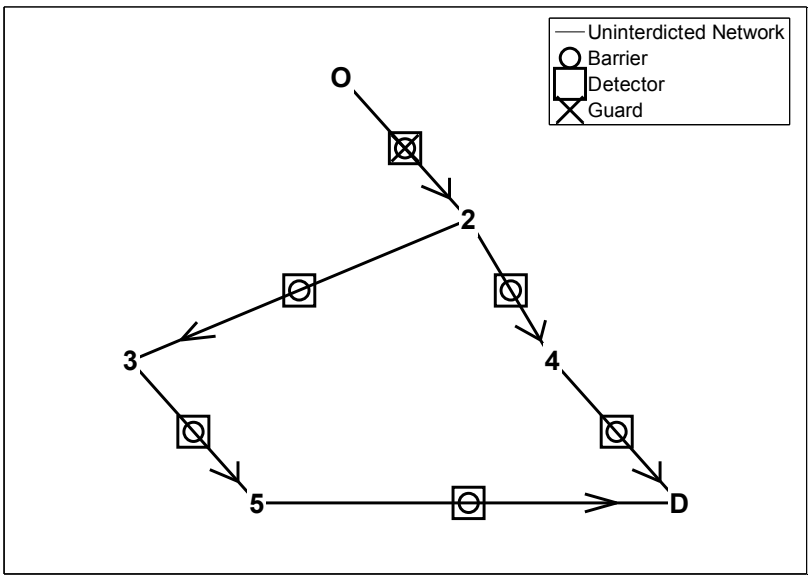


Figure 4-5: Baseline for Upgrades Analysis, Example Problem 1

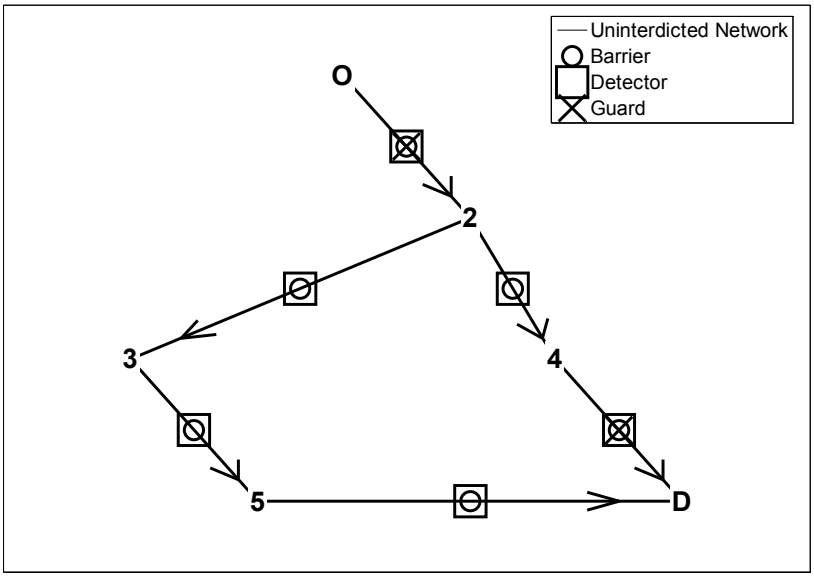


Figure 4-6: Upgraded Network, Example Problem 1

Figure 4-7 shows the P_{FS} vs. Cost graph for the upgraded network of Example Problem 1. Figure 4-8 shows the $Utility_{PPS}$ vs. Cost graph for the upgraded network of Example Problem 1. It is interesting to note that the highest $Utility_{PPS}$ point (as defined in Eq. (3-1)) does not occur when the safeguards budget is fully utilized. This would be an important point to convey to a facility owner, that is, the owner does not necessarily have to spend all of his or her budget to achieve optimal performance in terms of $Utility_{PPS}$.

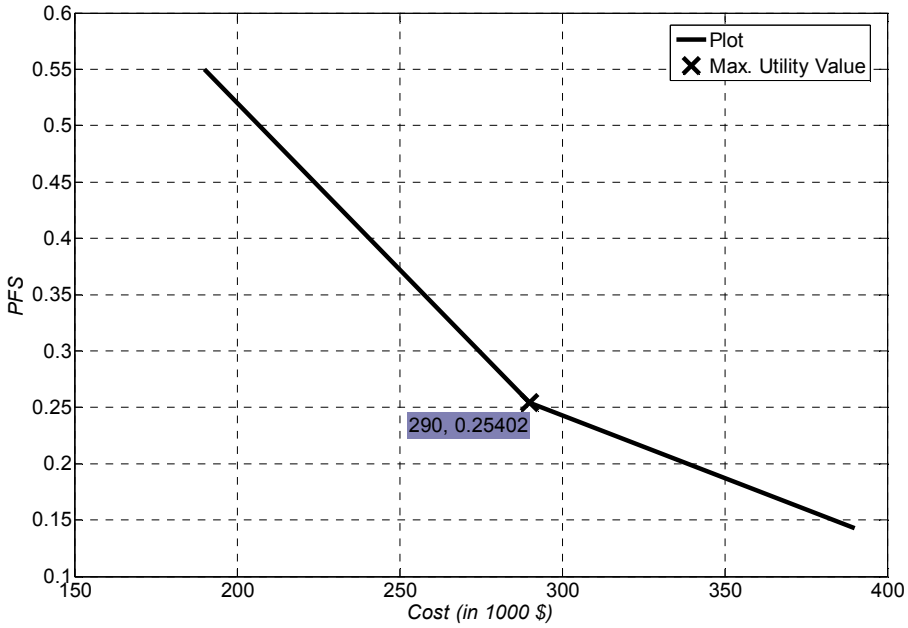


Figure 4-7: Upgraded P_{FS} Vs. Cost, Example Problem 1

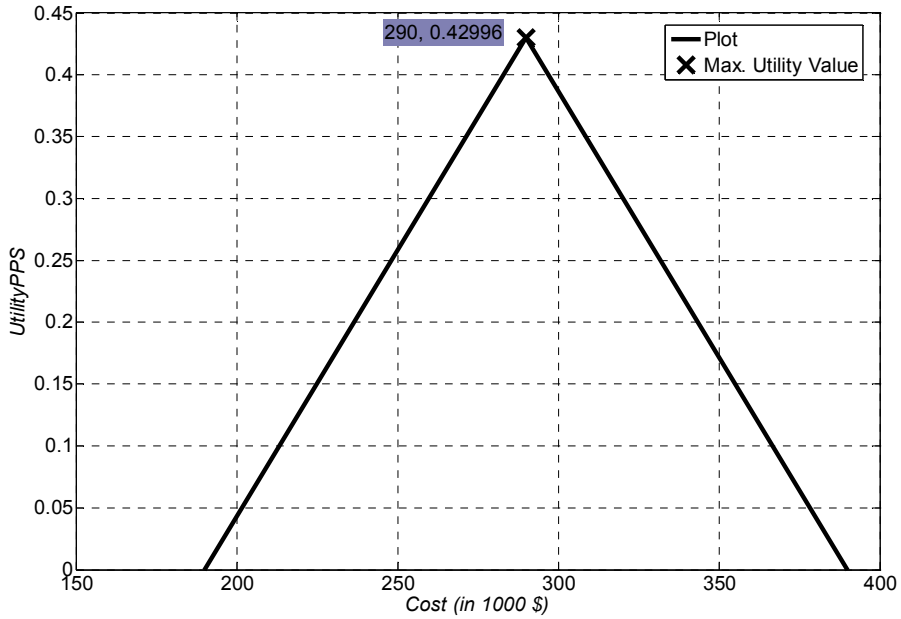


Figure 4-8: Upgraded *Utility*$_{PPS}$ Vs. Cost, Example Problem 1

4.5.3 New System Design

New system design is useful if the facility owner is creating a security system for a facility that has no security system installed. This could either be because the facility is new and thus has not been constructed or the facility could exist but not yet have a security system installed.

Figure 4-9 shows the PPS configuration with the highest value of *Utility*$_{PPS}$ for the network shown in Figure 4-3 when undertaking a new PPS system design. Intuitively, these results make sense. Having a guard response force stationed near the origin of the facility helps to provide ample time for the response force to neutralize the adversary. Additionally, the detectors placed on *O-2*, *2-3*, *3-5* and *5-D* alert the response force that are stationed on *4-D* so that they can respond to incidents on both paths. Finally, a

barrier placed on *5-D* slows down the adversary to allow for ample time for the response force (who are not stationed on that path) to interrupt and neutralize.

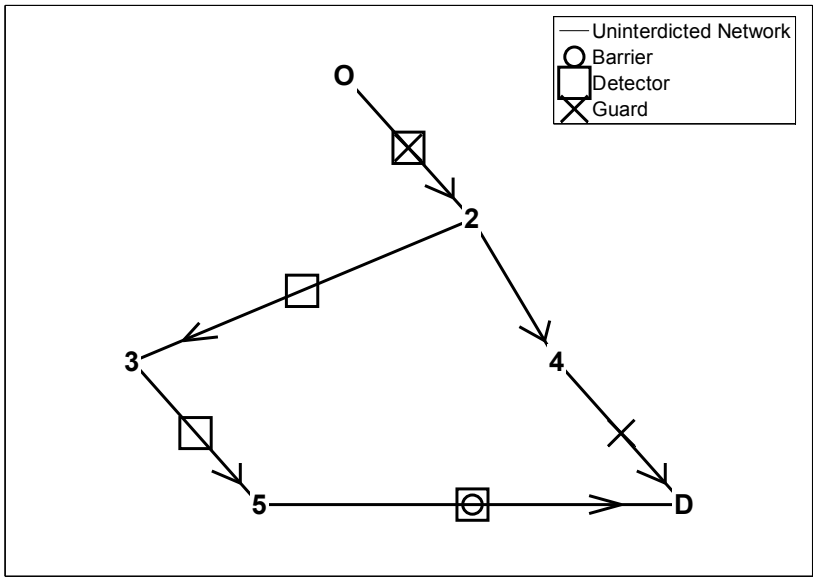


Figure 4-9: Optimal Safeguards Results, Example Problem 1

Figure 4-10 shows the response time for the guard stationed on O-2. Since both critical paths have O-2 as an arc, the guard stationed at O-2 always responds to O-2. Figure 4-11 shows the response time for the guard stationed on 4-D. When the adversary travels on O-2-4-D, the guard remains at 4-D. When the adversary travels along O-2-3-5-D, however, the guard is dispatched to 5-D.

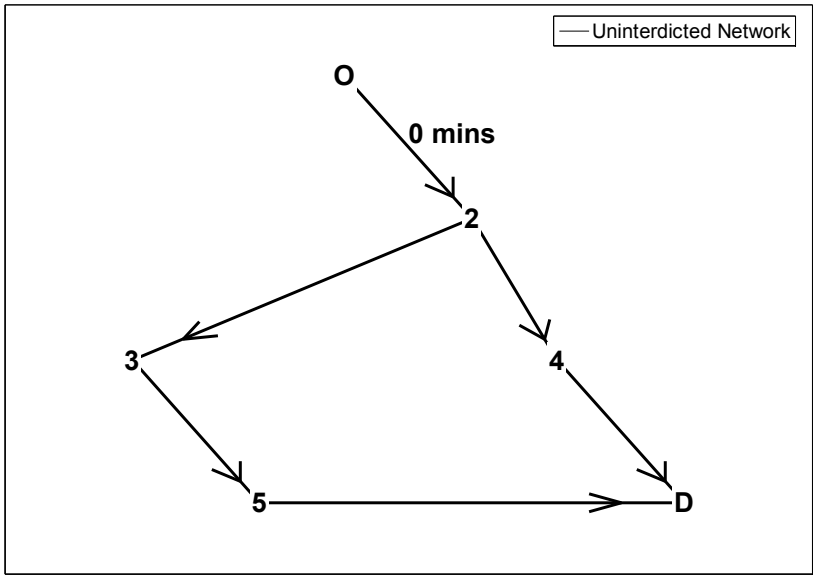


Figure 4-10: Guard Response Times for Guard Stationed on O-2

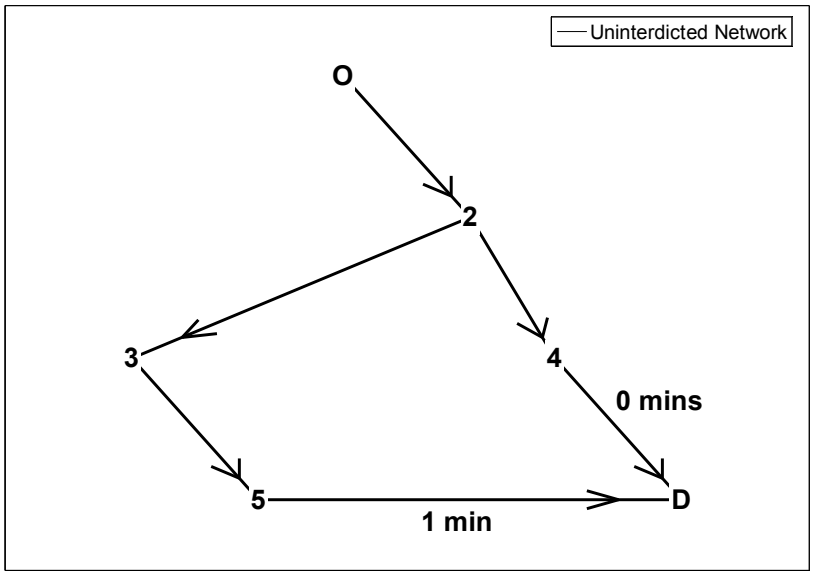


Figure 4-11: Guard Response Times for Guard Stationed on 4-D

Figure 4-12 shows the P_{FS} plotted versus *Cost* for the FORM and Monte Carlo results (they are identical). Running this experiment via MCS with 1,000 samples

yielded varying results, but the results shown in Figure 4-12 are from a 5,000 sample experiment, where the results proved to be stable and identical to the FORM results.

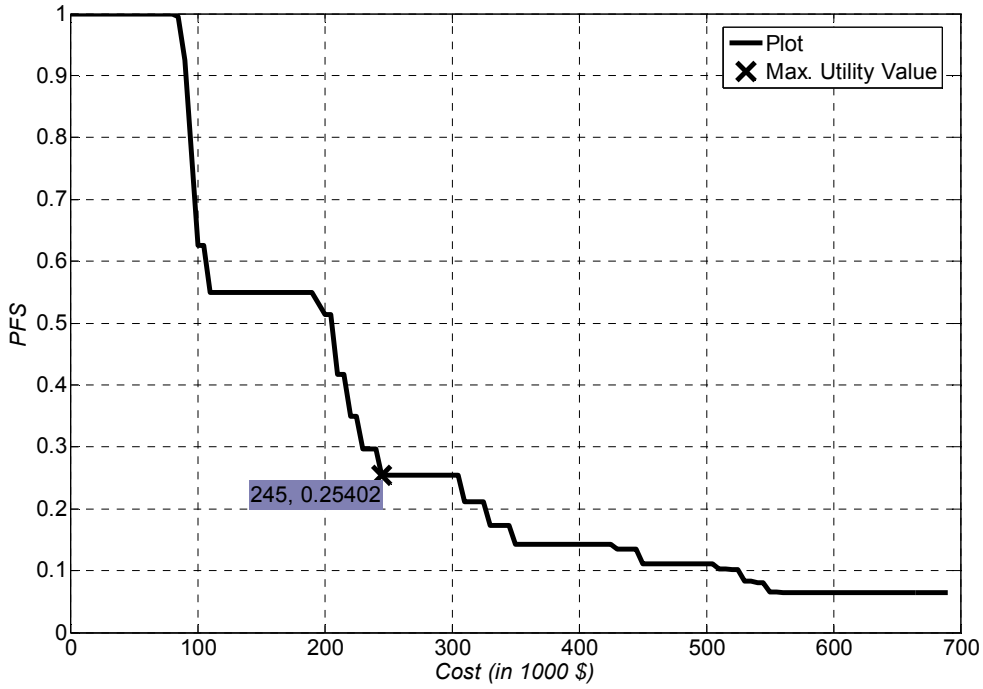


Figure 4-12: P_{FS} Vs. Cost, Example Problem 1

It is obvious that the facility owner can achieve a lower P_{FS} at a higher cost than the highest $Utility_{PPS}$ value, but this point represents the safeguards plan that best balances *Cost* and P_{FS}. In other words, at a cost above \$245K (P_{FS} = 0.25402), a lower P_{FS} can be achieved with additional safeguards, but the return on investment for the facility owner begins to diminish. So, the facility owner can invest in additional safeguards (at a cost above the highest $Utility_{PPS}$ configuration) in order to reduce the overall P_{FS} for the facility.

An additional observation that can be made about the results is how the off-site guards do not provide any protection in defending the facility. Although this is a simple example, the guards do not arrive on site in a timely manner and therefore, P_{FS} is equal to 1 until a cost of $100K, when the first on-site guard is placed in the facility. This is an important conclusion as it shows that off-site guards (given the problem's assumption of off-site guard response time) are not effective in defending facilities against an adversary threat.

While the highest $Utility_{PPS}$ scenario is calculated for an unconstrained problem (no constraints on *Cost* or P_{FS}), the problem can also be reformulated if constraints are enforced. For example, if the facility owner would not allow a P_{FS} value of above 0.1, then the highest $Utility_{PPS}$ value scenario can be recalculated using a truncated version of Figure 4-12. Figure 4-13 shows P_{FS} plotted versus *Cost* for a constrained scenario in which P_{FS} is constrained to be less than or equal to 0.1. A similar approach can be taken if cost constraints exist. Additionally, these constraints can be enforced at the outset of the problem, thereby reducing the solution space of the problem and reducing overall computational effort.

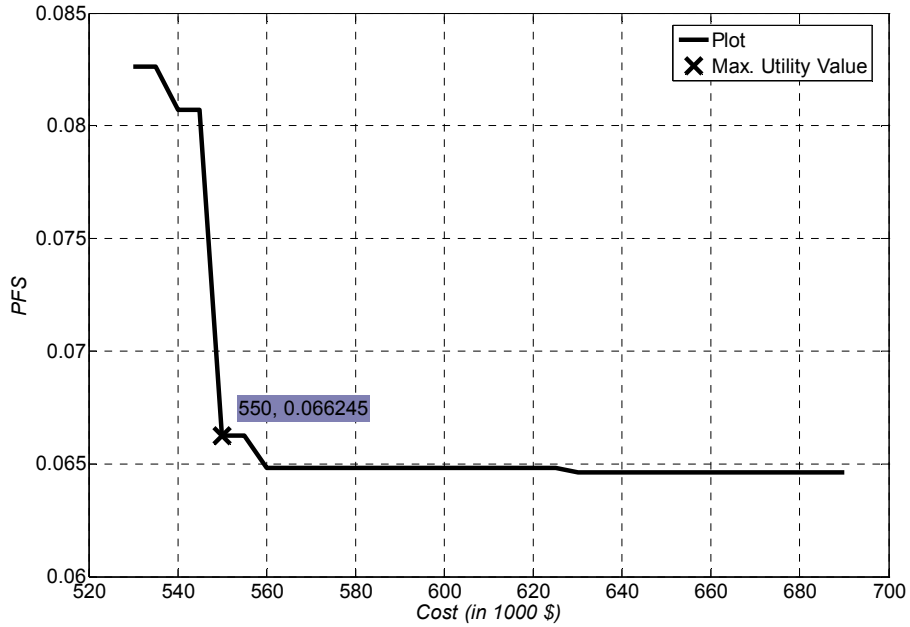


Figure 4-13: P_{FS} Vs. Cost with P_{FS} Constraint, Example Problem 1

It is also interesting to note, that, even if all safeguards are installed, P_{FS} does not reduce to 0. This is due to the fact that neutralization approaches 1 very slowly, i.e., it requires large numbers of guards to achieve a P_N close to 1. A solution to this would be to redesign the security system with the possibility of two guards being stationed on each arc. Figure 4-14 shows the safeguards configuration corresponding to the highest value of $Utility_{PPS}$ for the case in which two guards can be placed on any arc (with safeguard type 4 being the potential 2nd guard per arc). Figure 4-15 shows the *Cost* vs. P_{FS} curve for this problem. The revised scenario results in better performance at a lower cost. This is because the network topology is such that two guards can be stationed on the first arc in the facility and increase the P_N against both paths. From an operator's perspective, this configuration is superior but it is important to note that physical space constraints (as well as cost concerns) prevent the operator from having an unlimited supply of guards on each arc. The remainder of the results presented in Chapters IV and V show analysis allowing

for the possibility of only one guard team stationed on each arc, unless otherwise noted, with the understanding that all analyses can be extended if desired.

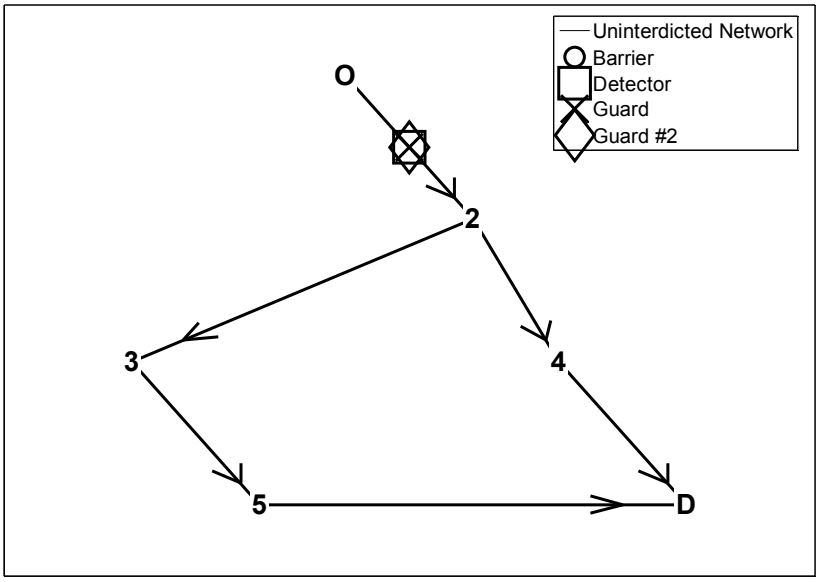


Figure 4-14: Optimal Safeguards Results (Two Guard Per Arc Limit), Example Problem 1

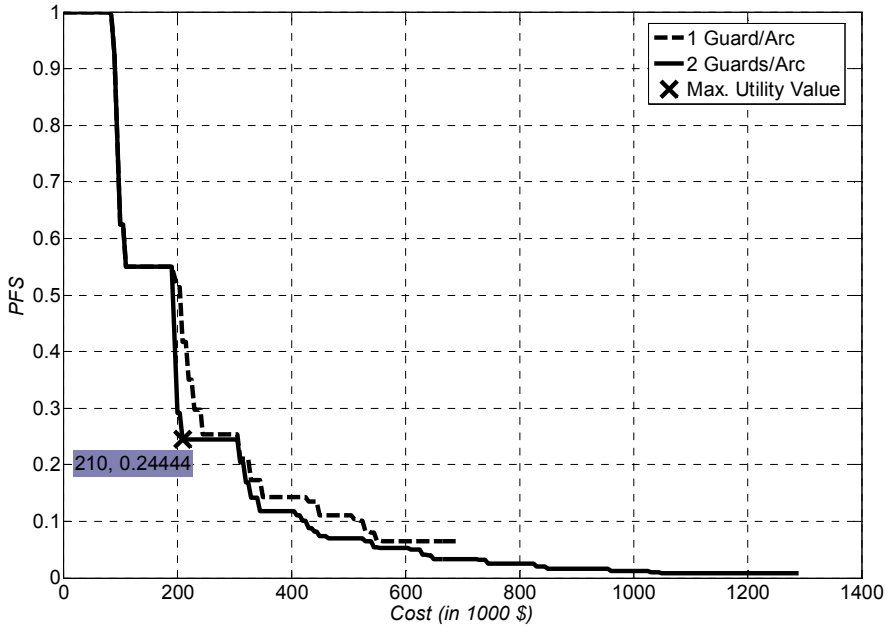


Figure 4-15: P_{FS} Vs. Cost (Two Guard Limit), Example Problem 1

Figure 4-16 shows a plot of $Utility_{PPS}$ vs. *Cost*. $Utility_{PPS}$ (Eq. (3-1)) is a function of both *Cost* and P_{FS}. Close inspection of Figure 4-16 shows that the maximum $Utility_{PPS}$ point is only slightly superior to another point on the graph of *Cost* = \$230K, $Utility_{PPS}$ = .5912. This is an interesting point as it illustrates the importance of balancing *Cost* and P_{FS} rather than relying on one metric to make a final decision.

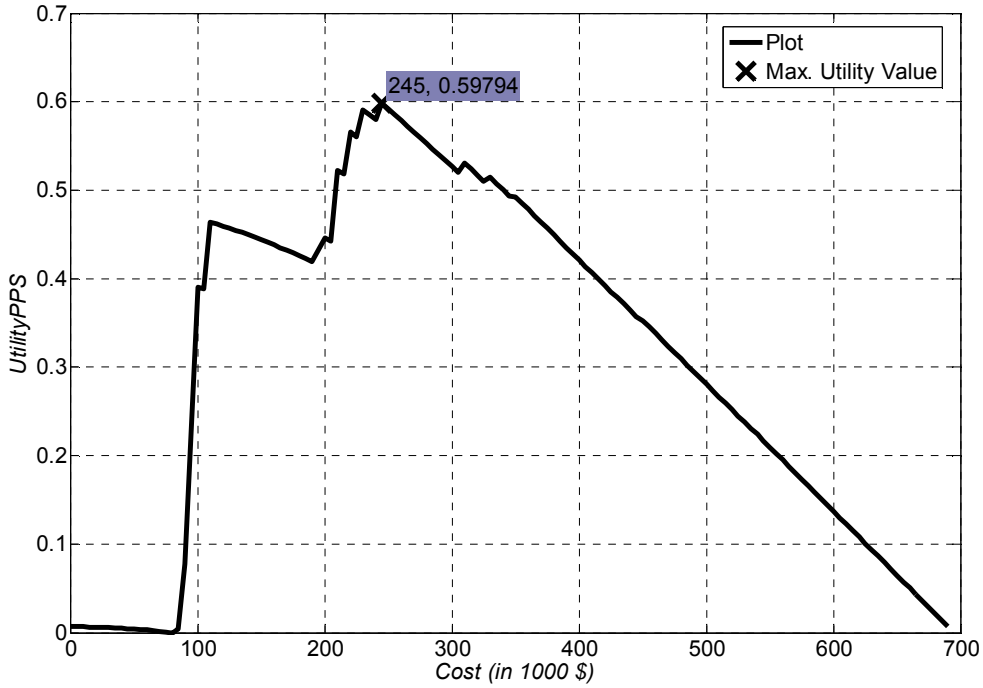


Figure 4-16: Utility Vs. Cost, Example Problem 1

Figure 4-17 shows a Tornado diagram[22] for the PPS configuration with the highest value of $Utility_{PPS}$ using FORM analysis. A Tornado diagram is a one-way sensitivity analysis which shows the effect of changing a particular variable (in this case, the mean value of each individual variable is multiplied by both 0.5 and 2) and recalculating P_{FS} with the final safeguards configuration as shown in Figure 4-9. The

bounds of the horizontal bar for each variable show the effect changing that input has on the final value of the objective function. In the case of this problem, it is clear that variation of the safeguards' detection probability has the greatest impact on the final P_{FS} (as it has the greatest range on the Tornado diagram). This tells the facility owner that if he or she does not have a large amount of confidence in his or her detection probability values, then the final performance of this PPS may not be as predicted. Additionally, safeguards delay time and θ (location in arc of detection) have a small influence on the final P_{FS}. It is important to note, however, that the tornado diagram is scenario specific and changes as the network topology and input variables do. Although expected, it is interesting to note that P_{FS} can vary either in a positive or negative manner, depending on the fluctuation of the input variables.

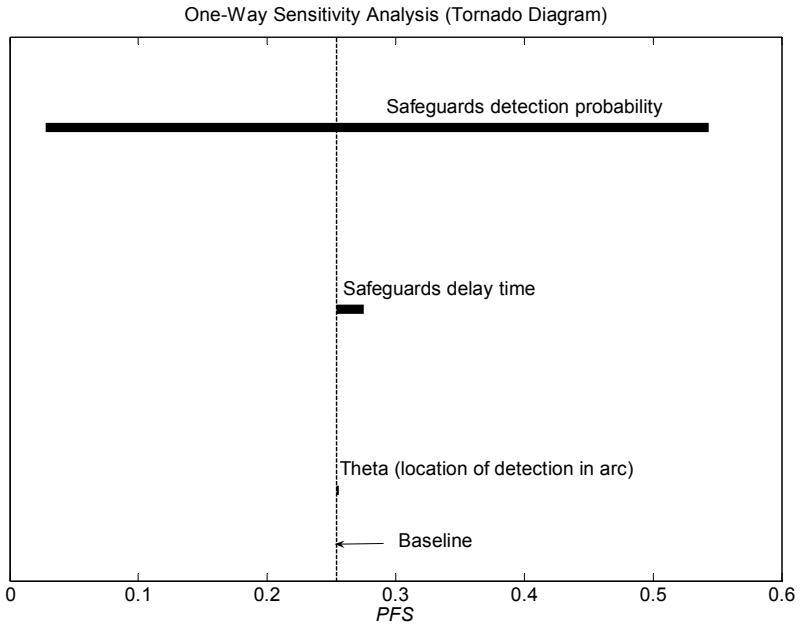


Figure 4-17: One-Way Sensitivity Analysis (Tornado Diagram), Example Problem 1

4.5.4 Computational Effort

The main criticism of Monte Carlo simulation is its large computational effort. This point is illustrated in Table 4-3, where the total CPU time for safeguards optimization and critical path selection are shown. The CPU used to run these experiments is a Dell Inspiron 5100, 2.8 GHz with 768 MB of RAM.

Table 4-3: CPU Time Summary, Example Problem 1

	FORM (s)	***Monte Carlo (s)***	***% Reduction in CPU Time (with FORM)***
Existing Facility Analysis	.065	.701	90.7
Upgrades Analysis	.430	11.8	96.4
New System Design	25.0	3804.8	99.3

In this case, achieving the same results as FORM through Monte Carlo simulation requires significantly greater computational effort. It is interesting to note that the computational savings due to Monte Carlo simulation increase as the problem complexity does. Recall that the results for Monte Carlo are shown for 1,000 samples for the existing facility analysis and upgrades analysis and 5,000 samples for the new system design. The increased number of samples for the new system design is necessary in order to stabilize these results when compared with FORM results. Many techniques exist[13,44] to decrease the required computational effort of Monte Carlo simulation such as antithetic variates, stratified sampling, Latin Hypercube sampling, and importance sampling, but they were not fully explored as they are considered outside the scope of this study. The times calculated are all computing times as reported by the computer algorithm used to solve

these problems and they can be viewed comparatively since they represent the time it takes to perform all of the given actions on a particular CPU.

The main criticism of FORM, conversely, is its inability to generate accurate results for highly non-linear limit states. In all examples presented in Section 4.5, the results from Monte Carlo and FORM were identical, which implies that the limit state is not highly non-linear. Most importantly, the goal of this study is to develop a decision-making methodology, and the suggested configuration chosen by both methodologies is the same. This could be due, however, to the simplicity of the presented problem. Section 4.6 explores a larger network in order to further examine the potential differences between Monte Carlo and FORM.

4.6 Example Problem 2

The second example problem is shown in Figure 4-18. In this problem, the Type 1 safeguard (barrier) now costs \$10K, otherwise all data in this problem are the same as in Example Problem 1, unless otherwise noted. A key difference in this example is the presence of multiple potential origins and destinations for the adversary. Multiple origins and destinations can be easily incorporated into the model by adding what is referred to as a super-origin or a super-destination, a node whose adjoining arcs are all artificial and have $T_t = 0$ and $P_D = 0$. These arcs connect the super-origin or super-destination to all potential origins or destinations, respectively. This concept is illustrated in Figure 4-18. The origin can be either node 2, 7, or 12, while the destination can be either node 6, 11, or 16. Figure 4-18 represents this uncertainty with a super-origin placed at O connecting to 2, 7, and 12 and a super-destination placed at D connecting to 6, 11, and 16. This

concept should be incorporated in the network when there are multiple possible locations for the adversary's origin and/or destination.

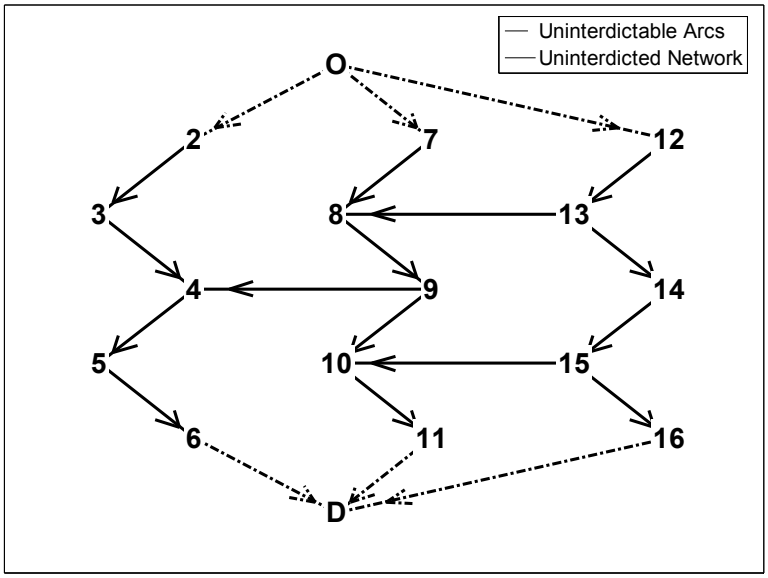


Figure 4-18: Network Graph, Example Problem 2

Additionally, the arcs that connect the super-origin and super-destination with the remainder of the network are also unable to have safeguards installed on them. This type of arc is referred to as an uninterdictable arc. It does not represent a physical part of the facility as it merely exists in order to facilitate analysis of a potential multiple origin or multiple destination facility. In order to account for this in the mathematical methodology, the interdictor merely needs to add constraints to the formulation in Eq. (4-6) which do not allow flow across all uninterdictable arcs (i.e. constraints that force flow, x, across an arc to be zero).

As in the previous examples, MCS analyses began by using 1,000 samples whenever P_{FS} was required, unless otherwise noted.

4.6.1 Existing Facility Analysis

In this problem, a random safeguards plan is generated. The network corresponding to this random safeguards plan is shown Figure 4-19. The result of the analysis are a P_{FS} = .2501 and a Cost = $940K for both FORM and MC analyses.

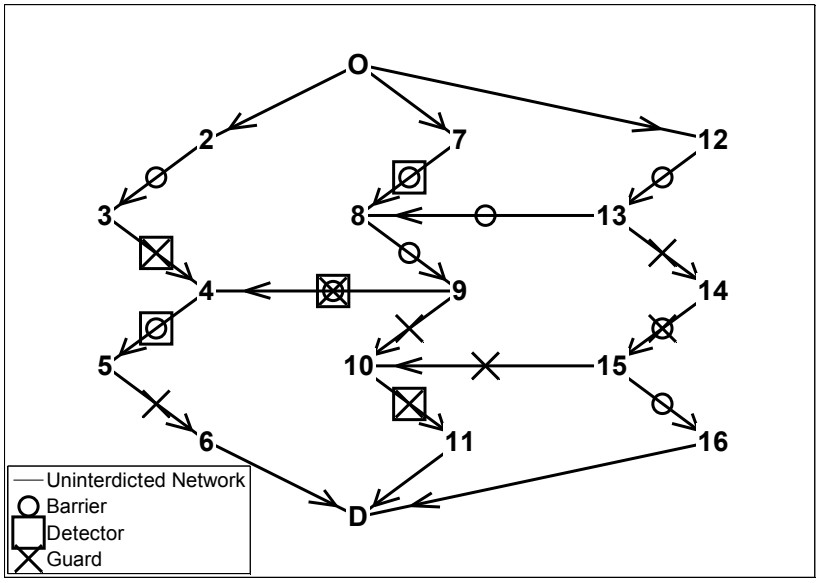


Figure 4-19: Randomly Generated Network, Example Problem 2

4.6.2 Upgrades Analysis

In the upgrades analysis, the baseline network is as shown in Figure 4-20. This safeguards configuration was randomly generated. The upgrades budget is $250K. All other data are as before. Figure 4-21 shows the results of the upgrades analysis for both FORM and Monte Carlo (their results are identical).

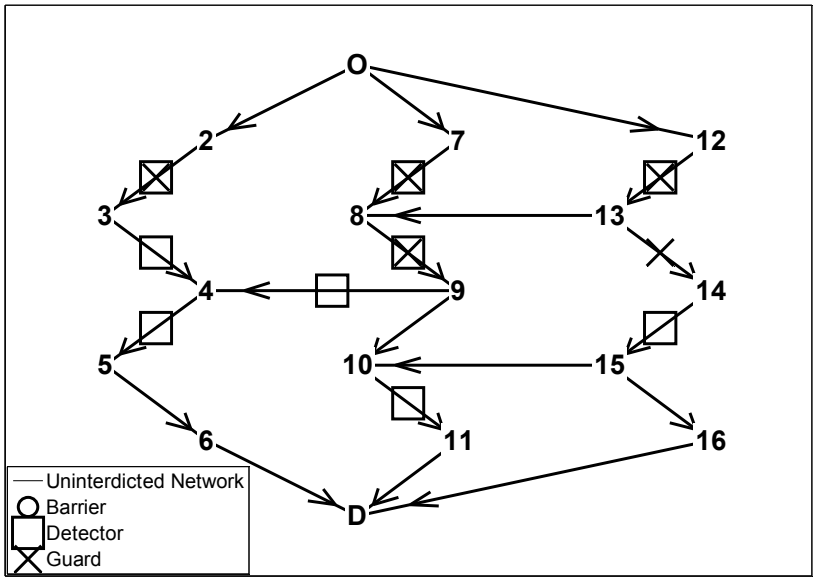


Figure 4-20: Baseline for Upgrades Analysis, Example Problem 2

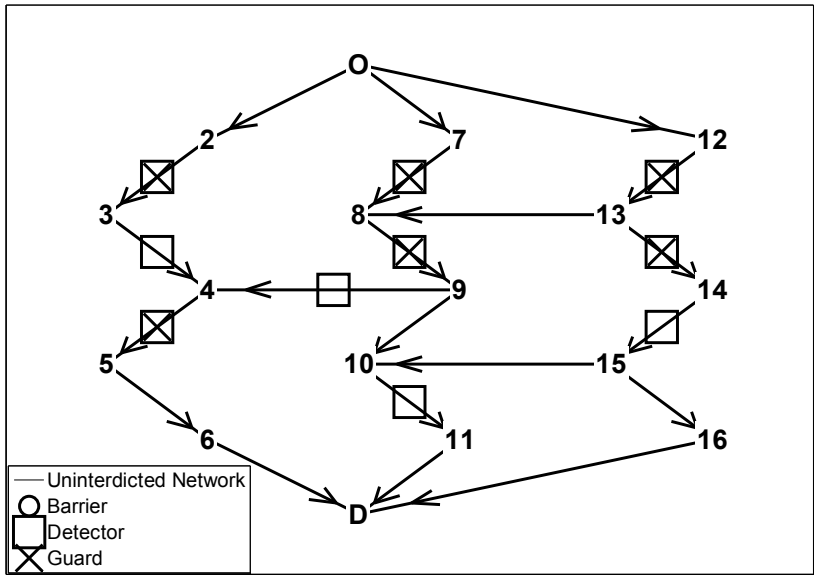


Figure 4-21: Upgraded Network, Example Problem 2

Figure 4-22 shows the upgraded P_{FS} vs. *Cost* for Example problem 2. It is again interesting to note that the point with the highest value of $Utility_{PPS}$ does not occur when the safeguards budget is fully utilized.

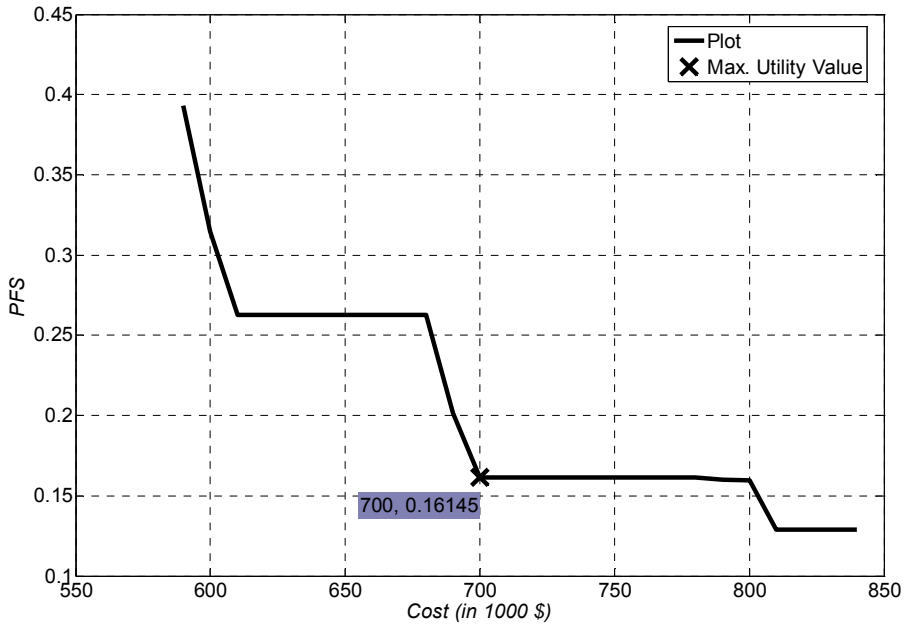


Figure 4-22: Upgraded P_{FS} Vs. Cost, Example Problem 2

4.6.3 New System Design

Figure 4-23 shows the PPS configuration for the highest value of $Utility_{PPS}$ for the network shown in Figure 4-18 when undertaking a new PPS system design using FORM. All variable statistics are as before in Table 4-1. The legend corresponds to the labels given to the safeguards in Table 4-2. As can be expected, this configuration requires significantly more safeguards when compared with Example 1. Additionally, the safeguards are spread out more evenly in this problem as there are now more attack paths for the adversary.

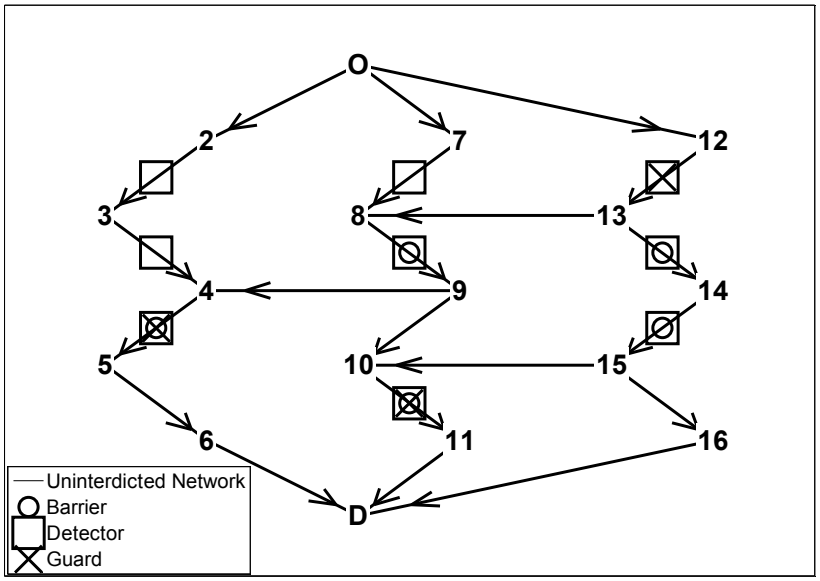


Figure 4-23: Optimal Safeguards Results, Example Problem 2

Cost vs. P_{FS} using FORM for this problem is shown in Figure 4-24. In this example problem, Monte Carlo becomes prohibitively expensive. As a result, Monte Carlo is further considered to be an impractical tool for designing a new PPS. It was still used, however, to solve the upgrades analyses and existing facility analyses to verify the accuracy of FORM.

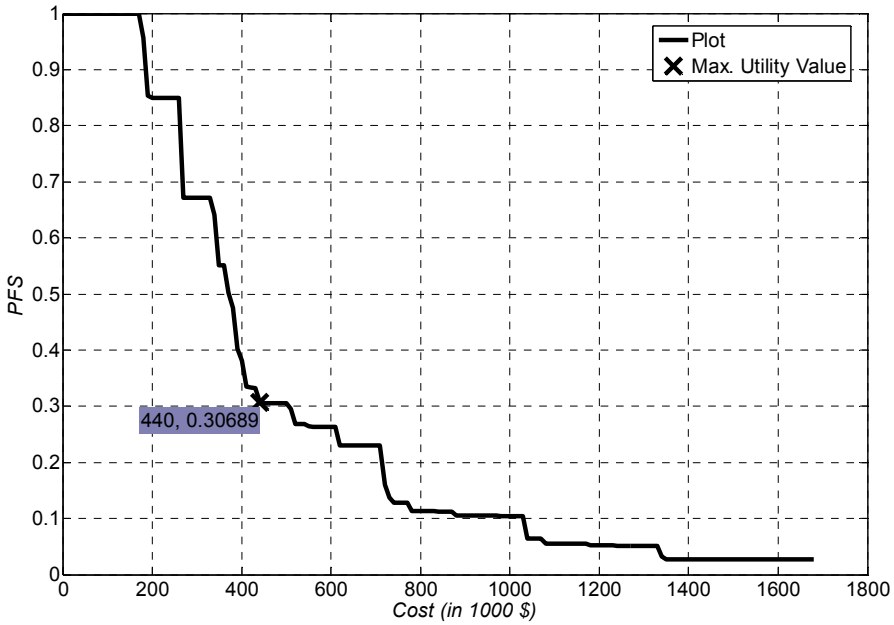


Figure 4-24: P_{FS} Vs. Cost, Example Problem 2

Figure 4-25 shows a plot of $Utility_{PPS}$ vs. *Cost* using the FORM analysis.

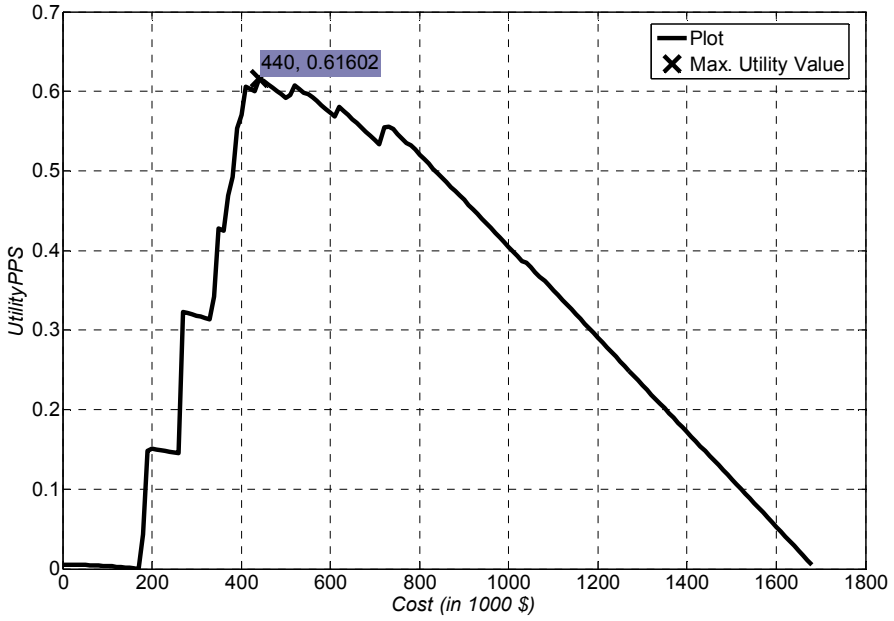


Figure 4-25: $Utility_{PPS}$ Vs. Cost, Example Problem 2

The remainder of the figures that are shown for Example 1 are not reproduced for Example 2 as it thought that they do not provide any further insight into the problem. Their presentation for the discussion of Example 1 is intended to show the capabilities of the developed methodology.

4.6.4 Computational Effort

The computational times comparing FORM and Monte Carlo for safeguards optimization and critical path selection are illustrated in Table 4-4. The CPU used to run these experiments is a Dell Inspiron 5100, 2.8 GHz with 768 MB of RAM. It is clear that achieving the same results through Monte Carlo simulation requires significantly greater computational effort, as was the case with the examples in Section 4.5. The times calculated are all computing times as reported by the computer algorithm used to solve these problems and they can be viewed comparatively since they represent the time it takes to perform all of the given actions on a particular CPU.

Table 4-4: CPU Time Summary, Example Problem 2

	FORM (s)	***Monte Carlo (s)***	***% Reduction in CPU Time (with FORM)***
Existing Facility Analysis	.768	6.31	87.8
Upgrades Analysis	61.8	3540.6	98.3
New System Design	449.2	39674**	98.9**

** Result shown is average of two runs of times for 1K samples and 5K samples. Actual savings would be greater as number of samples would be larger.

4.7 Simple Practical Example

In this section, the results of the proposed methodology are compared to Estimate of Adversary Sequence Interruption (EASI) results, a method developed by Bennett[10] and discussed in detail by Garcia[39] and briefly in Section 2.1.2.1, which many security professionals use to evaluate a PPS interruption probability and which the proposed methodology bases it calculations of P_I on. The experiments focus on calculating P_I for a given scenario, based on the adversary path shown below:

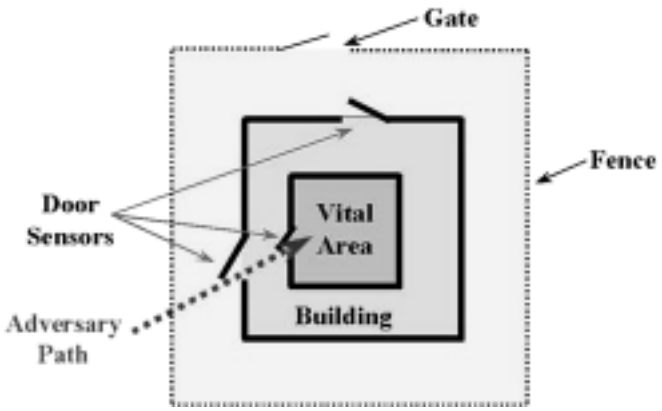


Figure 4-26: Adversary Path, Simple Practical Example[39]

This adversary path is translated into the following network:

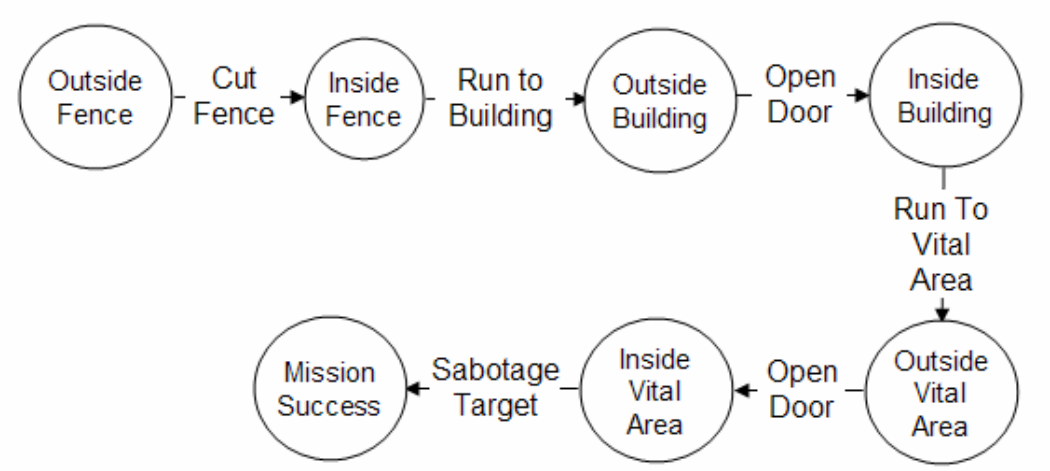


Figure 4-27: Network, Simple Practical Example[39]

Following are the data for the network shown in Figure 4-27. All variables are normally distributed with a coefficient of variation of 0.3. The data given are taken directly from Garcia[39]. Bold values represent changes to the baseline scenario (Scenario 1) in subsequent scenarios.

Table 4-5: Mean P_D Values

Task	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Cut Fence	0	**0.9**	0.9	**0.9**	0	0
Run to Building	0	0	0	0	0	0
Open Door	0.9	0.9	0.9	0.9	0.9	0.9
Run To Vital Area	0	0	0	0	0	0
Open Door	0.9	0.9	0.9	0.9	0.9	0.9
Sabotage Target	0	0	0	0	**0.9**	0

Table 4-6: Mean Delay Time Values (s)

Task	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Cut Fence	10	10	10	10	10	**30**
Run to Building	12	12	12	12	12	12
Open Door	90	90	90	90	90	90
Run To Vital Area	10	10	10	10	10	10
Open Door	90	90	90	90	90	90
Sabotage Target	120	120	120	**240**	120	120

Table 4-7: Mean Response Time Values (s)

Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
300	300	**200**	300	300	300

There are a few limitations to this example. The scenarios do not include P_N. Additionally, in Garcia's example problems, guard response time is identical for all arcs. The methodology presented in this study is more conservative in that guards are assigned to a particular arc and guard response time increases as the response location is further from the guard's station. This simplification results in a more conservative value for P_I when compared with the example problems. This simplification is realistic, however, as guard response time will not be equal for each arc. However, using the developed methodology (with the same assumptions as Garcia), identical P_I values were calculated for all six examples as compared to Garcia's values. These P_I values, for both the developed methodology and Garcia's methodology, are shown below.

Table 4-8: Scenario P_I Results

	Scenario P_I
Scenario 1	0.48
Scenario 2	0.58
Scenario 3	0.9
Scenario 4	0.84
Scenario 5	0.48
Scenario 6	0.48

4.8 Summary

The focus of this chapter was to develop a solution methodology to solve a single adversary attack problem. The methodology presented expands current research capabilities to include multiobjective optimization by maximizing the objective of $Utility_{PPS}$, with the understanding that it can be adapted to fit other objective functions. This chapter develops a method which overcomes the computational hurdles of previous network interdiction methods through the following improvements: (1) decoupling of critical path enumeration and safeguards optimization; (2) enumeration of only a unique critical path set and avoidance of enumeration of paths with repeated arcs; (3) use of efficient analytical approximations to calculate the stochastic constraints, compared to expensive Monte Carlo runs; and (4) use of an efficient decoupled stochastic optimization technique based on the concepts of reliability-based design optimization (RBDO). These improvements make it possible to solve realistic stochastic network interdiction problems in an efficient manner. These computational improvements allow for the objective considered to incorporate timely detection, neutralization probability, and interruption probability, extensions to previous methods. These improvements make solving real

world problems feasible. Chapter V discusses the extension of the presented methodology to account for a multiple team adversary attack.

CHAPTER V

MULTIPLE ADVERSARY TEAM METHODOLOGY

The previous chapters discussed formulations in which a single adversary team is attacking a facility. The third objective of this study is to extend the single adversary team methodology developed in Chapters II-IV to handle multiple team attacks, an issue that arises as adversary teams become more organized and adversary attack scenarios become more complex. This type of problem, in which multiple commodities (adversaries) share the same arcs within a network, is generally referred to as a multicommodity flow problem. If the teams do not interact in any way, the problem can be solved as multiple independent single team problems. If, however, there is some interaction between the adversary teams such as simultaneous travel on the same arcs, the adversaries' actions are no longer independent of one another and they must be modeled together. The multicommodity problem studied in this methodology has the added complication that the objectives of the individual teams are coupled with one another, that is, each team's actions can influence the effectiveness of other teams. It is impossible to separate each team's actions, thus increasing the computational effort required to solve this problem.

Multicommodity flows have been studied in detail and Assad[7] and Kennington[63] provide comprehensive surveys on the solution of these problems. It is not difficult to imagine situations in which an adversary may wish to transport multiple goods through a

facility resulting in a multicommodity problem, but as Lim and Smith point out[72], the best practical use of this problem formulation may involve the facility owner's examination of the worst-case scenario attack. A methodology is developed in this chapter to defend against the worst reasonable attack. As discussed in Section 1.2, a reasonable attack is one which is developed using the Design Basis Threat[1,2] analysis. The assumptions of Chapter IV are maintained throughout this chapter with the exception that the adversary threat is now treated as being a multiple team threat.

Use of multicommodity flow techniques to solve the facility protection optimization problem has to address three issues: 1) analysis of multiple adversary team attack strategies to include simultaneous and sequential adversary attacks, 2) development of a coupled implicit objective function, and 3) calculation of overall system-level reliability. The following section discusses the development of a methodology which can incorporate these concerns, by analyzing the influence of multiple teams on the methodology developed in Chapter IV.

5.1 Multiple Adversary Team Methodology

The standard multicommodity flow formulation[3] is for a problem in which the objective is to minimize overall transport cost (of all commodities), and is stated as follows:

$$\min \sum_{i \in A} \sum_{r \in R} c_i^{\,r} x_i^{\,r} \qquad \textbf{(5-1)}$$

$$s.t. \sum_{out} x_i^{\,r} - \sum_{in} x_i^{\,r} = \begin{cases} 1 \text{ if } x \text{ is a supply node} \\ -1 \text{ if } x \text{ is a demand node} \\ 0 \text{ for all intermediate nodes} \end{cases}$$

where c_i^r is the transport cost of the r^{th} commodity across the i^{th} arc, x_i^r is a variable indicating flow of the r^{th} commodity across the i^{th} arc, and R is the set of commodities.

In the proposed methodology, the multiple adversary teams are assumed to be identical. That is, they each have the same capabilities and any two teams perform the same task in equal time with an equal detection probability. If the problem were to have non-identical adversaries, its complexity would increase significantly. Consider a two-team, two-path scenario for illustration. With homogenous teams, team A traveling on path 1 and team B traveling on path 2 yield the same P_{FS} as team A traveling on path 2 and team B traveling on path 1. Thus, only the first scenarios must be evaluated. Heterogeneity amongst teams, however, means that all scenario combinations must be analyzed.

If the objective function in the multicommodity flow problem is linear, as in Eq. (5-1), the problem can be solved through price-directive decomposition, resource-directive decomposition, or partitioning methods (see Chapter 17 in Ahuja[3]). The formulation shown in Eq. (5-1) can be adapted to optimize any objective function in which the individual adversary objectives are not coupled. For example, if the goal is minimize the total cost of transporting multiple commodities across a network and there are no flow restrictions present, then the formulation shown in Eq. (5-1) merely seeks to find the best path for multiple commodities simultaneously. Some approaches have been developed for nonlinear uncoupled multicommodity problems. Lim and Smith[72] and Castro and Nabona[17] provide examples of this work. Recall, however, from Section 3.2 that the objective of the problem in this study is to maximize $Utility_{PPS}$ subject to cost and performance constraints. This problem is coupled for multiple teams, that is, interaction

effects between teams prevent the problem from being decomposed into k separable problems. Multicommodity problems have not been solved using coupled (non-separable) objectives such as this. As a result, a more complicated problem formulation must be developed.

Previous work[3,6,17,72] on multicommodity flows has not modeled the ability of the multiple commodities to assist (or hinder) one another, thereby lowering (or raising) the commodities' overall likelihood of failure. An example of this type of situation would be two individuals who are trying to infiltrate a facility for sabotage. It is conceivable that the travel time of the two individuals working together could be greatly reduced due to their cooperation with one another, but that their detection probability may increase. As such, a complete network model must be able to deal with both helpful and hurtful consequences of simultaneous flow.

Given that the scenario of multiple individuals working together on the same task seems likely, especially in the area of facility protection in which multiple adversary teams may work with one another to attack a facility, multiple team interaction must be addressed. The interaction of multiple adversarial teams is decomposed into three key components: (1) shared time in which teams work together on the same arc in a network (Section 5.1.1), (2) development of the scenarios in which multiple teams may interact (Section 5.1.2), and (3) overall objective function calculation (Section 5.1.3). The next three sections discuss these concepts in detail.

5.1.1 Shared Time

If the analyst is attempting to solve the problem of multiple team cooperation with regards to a real-world problem, the paths in which multiple teams may split up into are numerous. For a practical facility, these path combinations are a difficult combinatorial problem to solve. In order to analyze only a select group of paths which are of importance to the facility's protection, the analyst must develop a critical path set (CPS). This CPS selection is straightforward with regards to a single team, as shown in Section 4.1. While the same procedure is followed in a facility that requires multiple team cooperation, the choice of paths becomes significantly more difficult, as the interaction effects between the multiple teams must be taken into account. When multiple teams are introduced and there are potential interaction effects of the teams, however, both adversary task (i.e. travel) time and location are important. If a simple multicommodity analysis is performed without regard for task (i.e. travel) time and location in the facility, the interaction effects of multiple teams may be over- or underestimated with regards to one another. This concept is illustrated for two adversary teams in Figure 5-1 (Teams 1 and 2 in Figure 5-1 have origins of O1 and O2, respectively, and share a destination of D; the times shown in the figure represent mean task times):

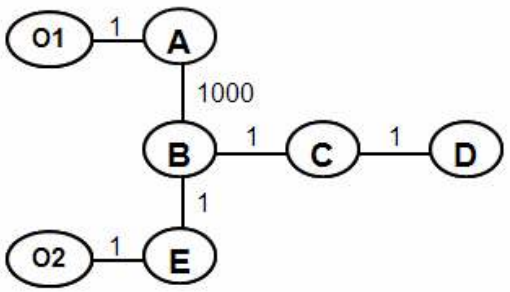


Figure 5-1: Multicommodity Interaction Illustration

Without a representation of the teams that includes both task (i.e. travel) times and location in the facility, one would think that the teams would interact on arcs *B*-*C* and *C*-*D*. Looking at the task times in Figure 5-1, however, one realizes that team 2 arrives at its final destination, *D*, well before team 1 completes its first task, assuming that team 2 will not wait for team 1. For this reason, there is no interaction between the two despite the fact that the two teams are traveling on the same path. This simple illustration demonstrates the importance of considering both task time and location within the facility when dealing with multiple teams.

Another way of looking at this interaction is by comparing the time of early finish (T_{EF}), the time when the first team finishes a given task, the time of early start (T_{ES}), the time when the first team starts a given task, the time of late finish (T_{LF}), the time when the 2nd team finishes a given task, and the time of late start (T_{LS}), the time when the 2nd team starts a given task. In order to take credit for shared time, T_{EF} needs to be greater than T_{LS}, i.e. the later team starts before the earlier team finishes. Shared time represents time across an arc in which two or more adversaries are traveling on the arc at once, as shown in Figure 5-2.

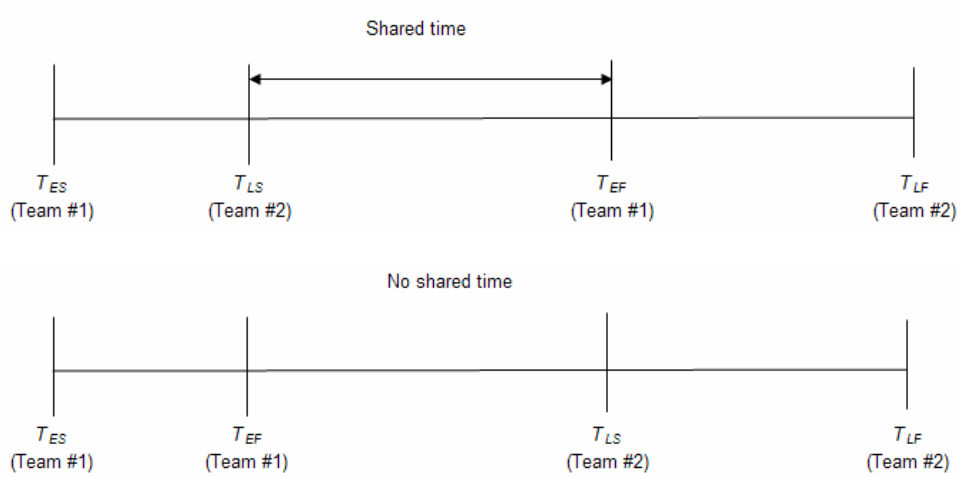


Figure 5-2: Graphical Depiction of Shared Time Concept

The concept of shared time is extremely important as real-world multicommodity problems involve this situation quite often. It should be noted that this discussion assumes two adversary teams, but it is possible to expand the discussion to more, as discussed later in this section.

Using Figure 5-2, T_{EF} is defined as:

$$T_{EF} = \min_{r} \sum_{i=1}^{ST_{END}} T_i^r x_i^r \tag{5-2}$$

Therefore,

$$E(T_{EF}) = \min_{r} \sum_{i=1}^{ST_{END}} E(T_i^r) x_i^r \tag{5-3}$$

where $E(T_i^r)$ is the expected travel time for commodity *r* across arc *i*, ST_{END} is the node at the end of the shared time arc, x_i^r is the flow of the r^{th} team across arc *i* (this is either 0 or 1). Only the times on the chosen path are to be included.

Using Figure 5-2, T_{LS} is defined as:

$$T_{LS} = \max_{r} \sum_{i=1}^{ST_{START}} T_i^r x_i^r \quad \textbf{(5-4)}$$

Therefore,

$$E(T_{LS}) = \max_{r} \sum_{i=1}^{ST_{START}} E(T_i^r) x_i^r \quad \textbf{(5-5)}$$

where ST_{START} is the node at the start of the shared time arc and all else is as before.

Using Figure 5-2, shared time, *ST*, is defined as:

$$ST = T_{EF} - T_{LS} \quad \textbf{(5-6)}$$

Therefore, the statistics of shared time can be calculated as:

$$E(ST) \approx E(T_{EF}) - E(T_{LS}) \quad \textbf{(5-7)}$$

$$Var(ST) \approx Var(T_{EF}) + Var(T_{LS}) - 2[Cov(T_{EF}, T_{LS})] \quad \textbf{(5-8)}$$

The above idea can be generalized for more than two teams. In that case, $E(ST)_{ikm}$ represents the average shared time between teams *k* and *m* across arc *i*. The shared time between each pair of teams on each arc is calculated. For example, if a scenario has three adversary teams and they are interacting on one arc, the interaction effects of Team 1 with Team 2 and Team 1 with Team 3 are calculated in order to determine the total effect of shared time on Team 1. The effect of this shared time is then combined for this arc in order to account for the total influence of all teams. This shared time combination is discussed in Section 5.1.3.

It is not difficult to extend this methodology to non-normal variables if desired[12,73,78,90,92]. With all normal variables, however, $P(ST>0)$ can be expressed as:

$$P(ST > 0) = \Phi\left(\frac{E(ST)}{\sqrt{Var(ST)}}\right) \quad \textbf{(5-9)}$$

While shared time is conceptually simple, it becomes difficult to incorporate this concept into the overall critical path selection and safeguards' optimization procedures. The following section addresses these concerns.

5.1.2 Scenario Development

The next question of interest is how shared time and multiple team interaction affect travel time and detection probability. Teams can interact either simultaneously or sequentially. Simultaneous teams operate within a facility at the same time and these teams can have either the same or different missions (defined simply in terms of each team's origin and destination). A simultaneous attack is a scenario in which many teams are working together to accomplish multiple simultaneous goals. Examples may be one team acting as a diversion while another team detonates a bomb in a facility or two teams with two independent targets such as two separate facilities or different targets within a facility. Additionally, two teams can be assigned the same origins and destinations in an attempt to ensure redundancy among the adversaries in case one team is neutralized. Typically these scenarios provide multiple chances for adversary success. That is, any of the teams accomplishing their goal can be considered a success by the adversary. This is conservative from a physical protection standpoint. In terms of the overall objective of the security system, then, success in stopping **all** teams is required in order for the security system to be considered successful. That is, if **any** of the adversary teams succeed, the PPS is considered a failure. If E_r represents the event in which an adversary is successfully interrupted and neutralized by the security system and $\overline{E_r}$ represents a so-

called failure event, in which an adversary defeats the security system (the complement of E_r), where $\overline{E_r} = 1 - E_r$, then P_{FS} can be generalized as follows:

$$P_{FS} = P(\cup_r \overline{E_r}) \tag{5-10}$$

Eq. (5-10) may be rewritten as

$$P_{FS} = P(\cup_r \overline{E_r}) = P(\overline{E_1 E_2 E_3 ... E_r}) = 1 - P(E_1 E_2 E_3 ... E_r) = 1 - P(\cap_r E_r) \tag{5-11}$$

The multiplication rule of probability for dependent events must then be used. For r events, the multiplication rule can be used to write:

$$P(E_1 E_2, ..., E_r) = P(E_1 | E_2, ..., E_r) P(E_2 | E_3, ..., E_r), ..., P(E_{r-1} | E_r) P(E_r) \tag{5-12}$$

For a single team,

$$P_{FS} = 1 - P(E_1) \tag{5-13}$$

as discussed in Section 2.1.2.

For two simultaneous teams, P_{FS} can be expressed as

$$P_{FS} = 1 - P(E_1 E_2) = 1 - [P(E_1 \mid E_2) P(E_2)] \tag{5-14}$$

For more than two teams, analytical computation of the joint probability of more than two events is difficult and can be solved numerically using Monte Carlo simulation or numerical integration methods[82]. Alternatively, approximate bounds have been developed such as first-order bounds by Ang and Amin[4] and Cornell[27] as follows:

$$\max_{i \in r} \left[P(\overline{E_i}) \right] \leq P_{FS} \leq \min \left[\sum_{i=1}^{r} P(\overline{E_i}), 1 \right] \tag{5-15}$$

In the above equation, the lower bound corresponds to the system failure case in which individual events are perfectly dependent and the upper bound represents the system failure probability if all the events are mutually exclusive. Haldar and

Mahadevan[52] point out that the first-order bounds can be quite wide. Narrower, second-order bounds were derived by Ditlevsen[30] as

$$P(\overline{E_1}) + \sum_{i=2}^{r} \max\left\{\left[P(\overline{E_i}) - \sum_{j=1}^{i-1} P(\overline{E_i}\,\overline{E_j}) \right], 0\right\} \le P_{FS} \le \min\left\{\left[\sum_{i=1}^{r} P(\overline{E_i}) - \sum_{i=2}^{r} \max_{j<i} P(\overline{E_i}\,\overline{E_j}) \right], 1\right\} \quad \textbf{(5-16)}$$

The events need to be ranked in order of decreasing likelihood in order to result in the narrowest bounds on the failure probability. This would make $P(\overline{E_1})$ the highest probability event. Since we are concerned with the worst-case scenario for the adversaries, this analysis focuses on the upper bound and designs the PPS based on this limit.

In addition to simultaneous teams, helper team scenarios (also known as sequential scenarios) allow for scenarios in which one team is the primary and other teams are referred to as secondary (or helper) teams. Helper teams attempt their tasks (e.g. disabling a closed-circuit television monitor or propping an access door open) prior to the start of the simultaneous teams' mission. These tasks influence the underlying facility network in some way (i.e. by lowering detection probability or task time across an arc). The success of the primary team is ultimately all that matters, while the secondary teams exist solely to support the mission of the leader. Even if all the helper teams fail, the primary team can still accomplish its mission; the drawback to the adversary is that his or her mission becomes significantly more difficult. This approach to helper teams is conservative. In reality, failure by a helper team may negatively influence the mission of the simultaneous teams either directly (by not lowering detection probability or task time) or indirectly (by altering the response force to an impending attack).

Additionally, scenarios may arise in which there are a combination of helper and simultaneous teams. For instance, two primary teams may need to breach a wall at the same location and a helper team may be employed to create an opening in the wall prior to the primary teams' arrival (in order to make the primary teams' task easier).

This situation can be generalized to r primary teams, where P_{FS} is defined as the failure due to the optimal adversary scenario as:

$$P_{FS} = \max_{r \in R} P_{FS}^{\ r} \quad \textbf{(5-17)}$$

It is worth noting in the above equation that there is no consideration for the helper teams in the overall scenario P_{FS}. As previously stated, while secondary teams assist the primary teams' overall goal, their success is not essential to overall adversary mission success.

The factors that influence how task time and detection probability change in the presence of multiple teams are the simultaneous primary team time factor (*STTF*), simultaneous primary team detection factor (*STDF*), helper team time factor (*HTTF*), and helper team detection factor (*HTDF*). These factors can be defined based on a per-task basis, but for simplicity's sake, they are assumed for the remainder of this study to be constant for a particular facility. Each of these factors can range from [-1,1]. These factors can be thought of as a relative value of efficiency of additional teams and they are defined based on study into the effects of multiple teams or the user's best estimate. In the case where *STTF*=0, there is no added benefit or detriment of having multiple teams. If the time to complete the given task is *E*(*ST*), then that is the total task time, regardless of the number of teams. A *STTF* of greater than 0 means that the presence of multiple teams hinders one another and raises the overall task time to greater than the original

time. A negative *STTF* means that the presence of additional teams is speeding up the completion of the given task. An example of a task that may have a negative *STTF* would be breaching a wall. Once an opening has been created in the wall, it does not need to be recreated for multiple teams. On the other hand, a safeguard such as a security checkpoint could have a positive *STDF* since increasing the number of adversaries attempting to gain access through the checkpoint increases the likelihood they are caught.

HTTF is defined in the same manner as *STTF* as a measure of helper team efficiency in altering task time. *HTDF* is defined in the same manner as *HTTF* as a measure of helper team efficiency in altering detection probability.

The following section discusses how to incorporate the effects of both sequential and simultaneous teams into the overall $Utility_{PPS}$ calculation.

5.1.3 Mathematical Model of $Utility_{PPS}$

Given the presence of multiple teams, calculation of $Utility_{PPS}$ can no longer be represented in a closed form, as in Eq. (3-1). Expanding the model developed in Chapter IV using the concepts developed in the last two sections, the objective function becomes implicit and follows the general outline below (with details to follow):

1. Calculate effect of helper teams on primary simultaneous teams.
2. Calculate shared time amongst primary simultaneous teams.
3. Calculate effect of primary simultaneous teams on one another.
4. Calculate T_G.
5. Calculate P_N.
6. Calculate P_{TD}.

7. Calculate system level P_{FS} for each scenario of teams amongst paths (e.g. worst path set for 2 simultaneous primary teams, etc.)
8. Determine overall P_{FS} among different scenarios of teams (e.g. 1 helper team, 1 primary team; 2 simultaneous primary teams)
9. Calculate $Utility_{PPS}$

Helper teams involve two key concepts which must be addressed before incorporating them into the methodology. Since they perform work prior to the leader's arrival, there is a chance they may succeed or fail. If they succeed, their work impacts the adversary in some form. If they fail, their work can have a detrimental effect (although it is conservatively assumed in this methodology that capturing a helper team does not further alert the response force to an impending attack on the facility, as the methodology is designing for a worst case scenario. Another approach would be to assume that helper team detection results in primary team mission failure). So, first the success of the helper teams must be analyzed. This is done by analyzing the helper teams as a group of simultaneous teams, calculating P_{FS} for each team. Probability of helper success is defined through the complement $P_{HS} = 1 - P_{FS}$. This value is then used, along with *HTTF* and *HTDF* in determining the helper team's effects on P_D (see Eq. (3-6) for P_D derivation) and T as follows:

$$P_{D_i} = [1-(1-P_{D_O})\prod_{l\in S}(1-P_{D_{li}}y_{li})]\left[1+\sum_{h\in H}(HTDF_h)(P_{HS\,h})\right] \quad \textbf{(5-18)}$$

$$T_i = T_i\left[1+\sum_{h\in H}(HTTF_h)\left(P_{HS_h}\right)\right] \quad \textbf{(5-19)}$$

$$T_l = T_l\left[1+\sum_{h\in H}(HTTF_h)\left(P_{HS_h}\right)\right] \quad \textbf{(5-20)}$$

where h is the helper team index, H is the set of helper teams, T_i is the i^{th} task time, T_l is the delay time due to $\underline{l^{th}}$ safeguard, and all else is as before in Chapter IV. The summation present in Eqs. (5-19) and (5-20) aggregate the effect of multiple helper teams.

Once the updated detection probabilities and travel times have been calculated, the next step is to calculate the shared time amongst the simultaneous teams. Expected shared time is calculated based on Eq. (5-7) for each combination of arcs and teams. Then, the effect that shared time has on the baseline P_D and T is calculated as follows:

$$P_{D_{ik}} = P_{D_i}\left\{1+\frac{\left[\sum_m ST_{ikm}\right]STDF_i}{T_i+\sum_l T_l y_{il}}\right\} \quad \textbf{(5-21)}$$

$$T_{ik} = T_i\left\{1+\frac{\left[\sum_m ST_{ikm}\right]STTF_i}{T_i+\sum_l T_l y_{il}}\right\} \quad \textbf{(5-22)}$$

The above equations correspond to a weighted sum of the non-shared time on an arc and the shared time, multiplied by the multiple team adjustment factors (*STDF* and *STTF*). The summation present in the numerator Eqs. (5-21) and (5-22) aggregate the effect of multiple simultaneous teams.

If shared time is equal to the total travel time (including the sum of baseline and delay due to safeguards, as shown in the denominator) for an arc, then Eqs. (5-21) and (5-22) equations reduce to:

$$P_{D_{ik}} = P_{D_i}\left\{1+STDF_i\right\} \quad \textbf{(5-23)}$$

$$T_{ik} = T_i\{1 + STTF_i\} \quad \textbf{(5-24)}$$

On the other hand, if no shared time exists on an arc, Eqs. (5-21) and (5-22) reduce to $P_{D_{ik}} = P_{D_i}$ and $T_{ik} = T_i$. P_{Dlik} and T_{lik} for the safeguards are calculated in a similar manner to the above equations, with P_{Dli} and T_{li} replacing $P_{D\,i}$ and T_i in Eqs. (5-21) and (5-22) to the left of the outside brackets.

Once the updated detection probabilities and travel times are calculated (both baseline and safeguards), the next step is to calculate guard response time and probability of neutralization. This process follows the same procedure as shown in Section 3.2.1.3, with the only difference being that the presence of multiple adversary teams (at multiple locations or the same location) results in the guard response force potentially being further dispersed, as the same number of guards is now attempting to neutralize a larger number of adversaries. This dispersion means that there are on average less response force personnel to engage in neutralization for each adversary. For instance, if there are two potential adversary paths in a facility, in a scenario in which the adversaries will travel down separate paths, the guard forces will split up to neutralize both adversaries. As a result, the probability of neutralization for the guard response force decreases (as P_N is based on the number of adversaries compared with the number of response force personnel, as in Eqs. (3-9) and (3-10)). In other words, achieving the same level of neutralization for a multiple adversary attack requires additional response force personnel, and therefore, a higher safeguards budget.

The next step is to calculate the probability of timely detection. The procedure is the same as outlined in Eq. (3-7) with the only addition being that the probability of timely detection must be calculated for each team on each arc.

Once all of the above calculations have been performed, they can be combined to compute P_{FS} for each path set (e.g. primary teams 1 and 2 travel on path A, primary team 1 travels on path A and primary team 2 travels on path B, etc.), as shown in Eq. (3-5). This calculation is performed for each set of paths identified during the critical path selection process. This calculation is also performed for each combination of helper and simultaneous teams up to the maximum number of teams specified in the analysis. It is assumed that the adversary teams utilize the maximum number of teams specified in the Design Basis Threat[1,2] in order to maximize their effectiveness. This information for real facilities is classified, but in the event that an analyst performs this analysis on a facility, he or she would have access to this data. Therefore, sub-optimal scenarios (i.e. one in which the adversary only utilizes two teams when the Design Basis Threat allows for three teams) are ignored. The worst case value of P_{FS} is retained for each scenario. This results in a table similar to the one shown below for an example with three paths sets for each scenario and three teams (the example below is shown entirely for illustrative purposes and does not correspond to any problem in this study):

Table 5-1: Example Multiple Team P_{FS} Results

Adversary Configuration	Path Set 1	Path Set 2	Path Set 3	Worst-Case Path
1 primary team, 2 helper teams	0.2	0.15	0.3	0.3
2 simultaneous primary teams, 1 helper team	0.1	0.05	0.15	0.15
3 simultaneous primary teams, 0 helper teams	0.26	0.32	0.15	0.32

The path sets shown in the above table are not the same between adversary configurations, i.e. Path Set 1 for 1 primary team, 2 helper teams ≠ Path Set 1 for 2 simultaneous primary teams, 1 helper teams. Figure 5-3 shows an example network with Path A (O-2-3-5-D) and Path B (O-2-4-D). Path set 1 for the adversary configuration of 1 primary team, 2 helper teams in the above table may correspond to primary team 1 traveling on path A, with helper team 1 helping on O-2 and helper team 2 helping on 2-3. Path set 1 for the adversary configuration of 3 simultaneous primary teams may correspond to all three primary teams traveling on Path A. The important point is to enumerate all possibilities for each adversary configuration and determine the configuration and path set which results in the most vulnerable attack. This corresponds to Steps 7 and 8 in the objective function outline discussed previously.

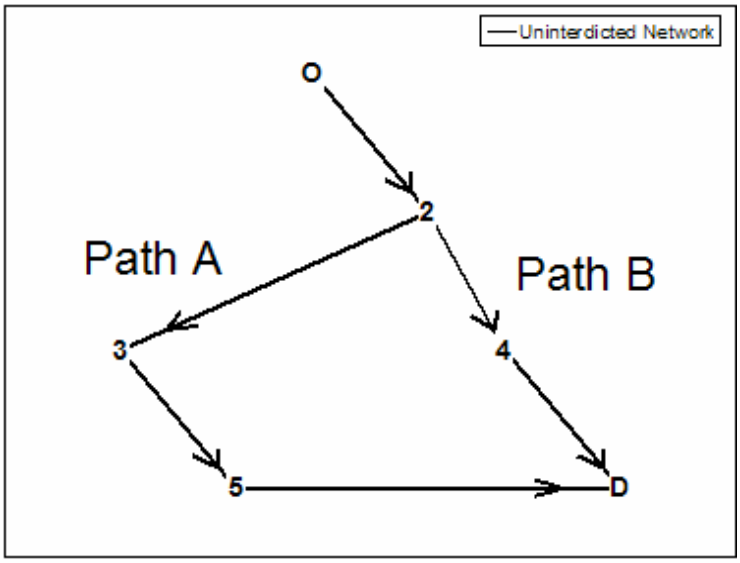


Figure 5-3: Example Network

Once the worst-case path is identified for each scenario, the overall P_{FS} is then chosen as the worst case P_{FS} for the set of scenarios. The reason for choosing this approach is that the analyst realizes that he or she does not know which configuration and path the adversary will choose to use, but he or she assumes that the adversary will choose the scenario which gives them the greatest likelihood of success. Using the above example, the overall P_{FS} for three teams would be 0.32.

The multiple team analysis results in the following overall problem statement (based on the single team analysis in Eq. (3-3)):

$$\begin{aligned} &\max_{n\in N} \min_{r\in R} E[Utility_{PPS_n}] \\ &\text{s.t.} \min_{n\in N}(Cost_n) \le Cost \le \max_{n\in N}(Cost_n) \\ &\text{s.t.} \min_{n\in N} E[P_{FS_n}] \le E[P_{FS}] \le \max_{n\in N} E[P_{FS_n}] \end{aligned} \quad \textbf{(5-25)}$$

where all variables are as before.

This problem statement decomposes into the following optimization at each cost step, *n* (similar to Eq. (4-6)):

$$\max_{r,k} \min \mathrm{E}(P_{E_k}) = \sum_{i=1}^{n} \left[E(P_{D_{ik}}) P_{TD_k|D_{ik}} E(P_{N_k|TD_k}) \right] \quad \textbf{(5-26)}$$

$$\text{s.t.} \sum_{l \in S} C_l \sum_{i \in A} y_{li} \leq budget$$

where all variables are as before. The optimization is performed over *R*, the set of adversary scenarios, and *CPS*, the critical path set, in order to ensure the resulting safeguards plan protects against the worst case path (and all others).

The solution of the above problem formulation follows the same solution procedure as outlined in Chapter IV for a single team analysis. Examination of the calculation of P_{FS} for multiple team analysis shows that the single team adversary analysis is a special case of the multiple team analysis in which steps 1-3 and 8 are eliminated from the process outlined at the beginning of this section.

For the remainder of this study, all results shown are based entirely on FORM results and do not include Monte Carlo analysis, as Chapter IV demonstrated both methodologies yield the same optimal decisions. Additionally, the results for the remainder of this chapter focus on new system design. As it is the most comprehensive analysis of the three, both existing facility and upgrades analyses are assumed to be feasible.

Following are two example problems. They represent the same two facilities from Chapter IV, each being analyzed as a new facility PPS design combating a multiple team adversary attack.

5.2 Example Problem 1

In this example problem, Example Problem 1 from Section 4.5 is revisited. Figure 5-4 shows the network from this problem, with all primary teams having the same origin (*O*) and destination (*D*). A new system design is undertaken for 1, 2, 3, 4, and 5 teams in order to show the difference in performance among these different scenarios. For each of these scenarios, all combination of helper and simultaneous teams up to the maximum number of teams were utilized (i.e. for a two adversary team threat, scenarios are 1 primary team, 1 helper team and 2 primary simultaneous teams, 0 helper teams). Observe that the results from Section 4.5 are utilized here as the results for a single team. Figure 5-5 shows the *Cost* vs. P_{FS} curve for the multiple team experiment. Figures 5-5 through 5-9 show the safeguards plans corresponding to the maximum $Utility_{PPS}$ for the different adversary threats.

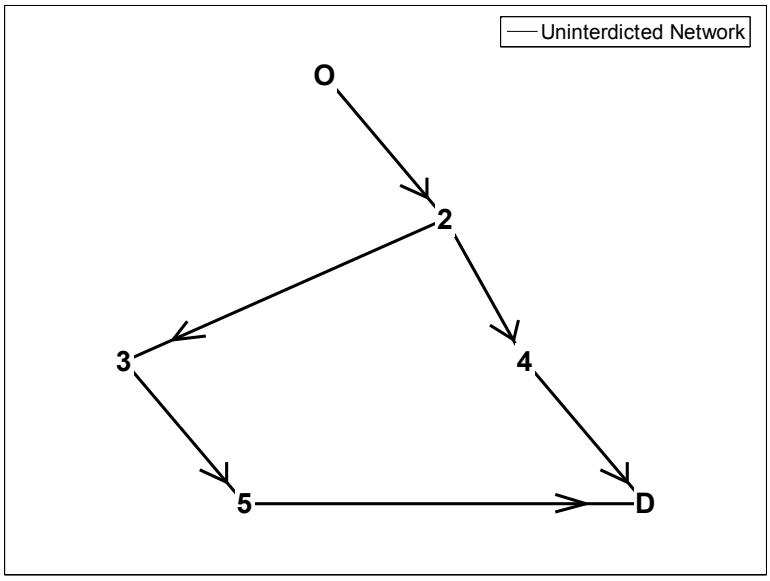


Figure 5-4: Multiple Adversary Example Problem 1

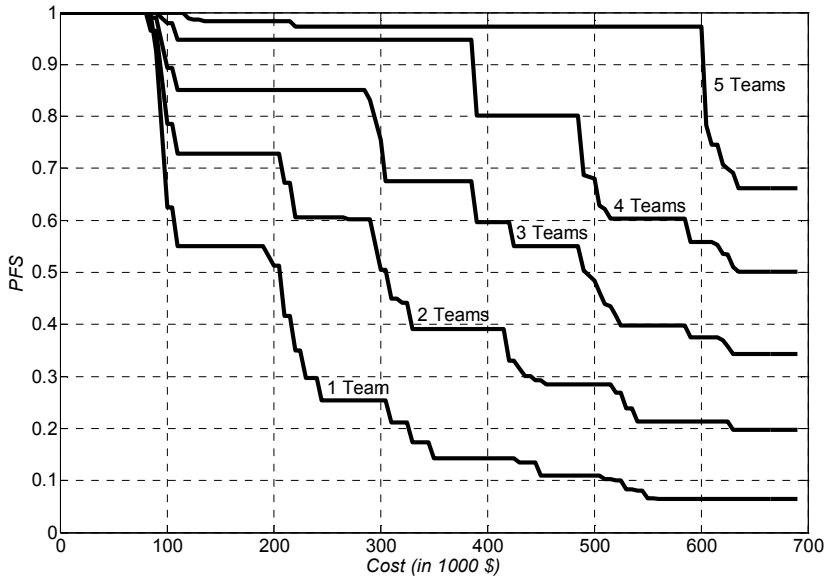


Figure 5-5: P_F Vs. Cost, Multiple Adversary Example Problem 1

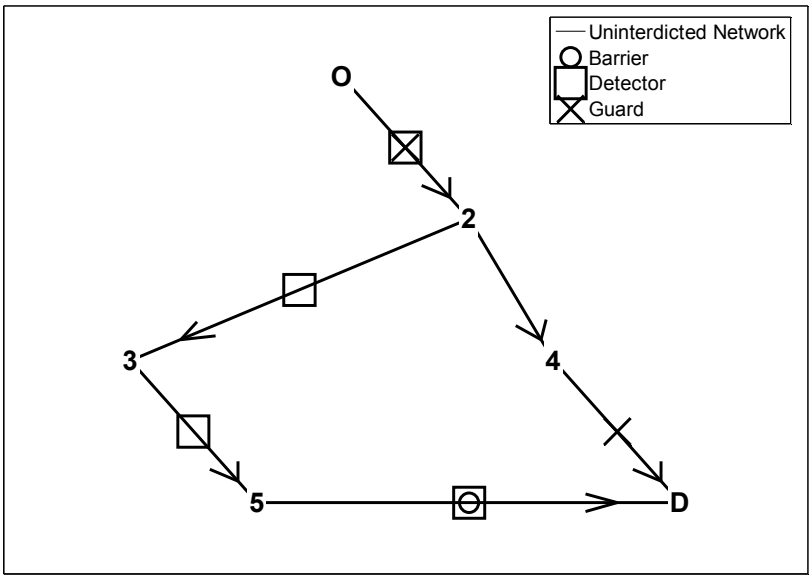


Figure 5-6: Maximum $Utility_{PPS}$ Configuration, Multiple Adversary Example Problem 1, Single Adversary Team

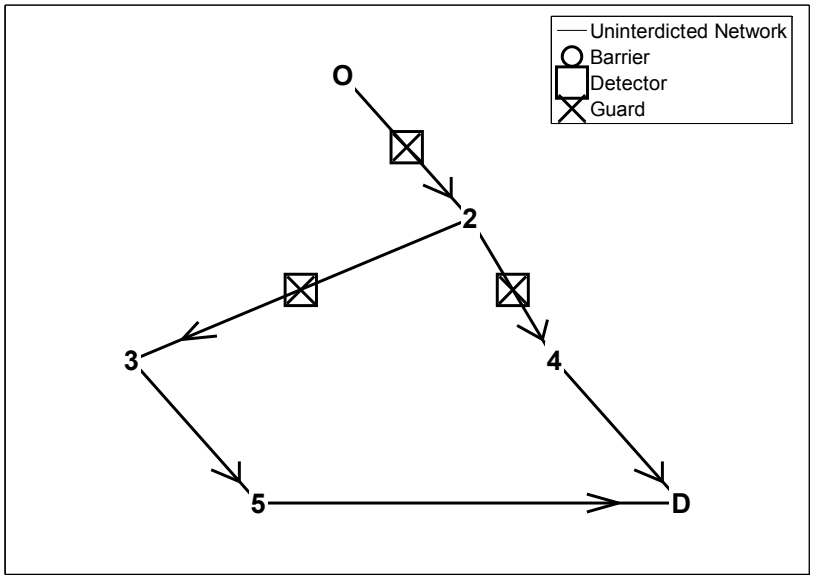


Figure 5-7: Maximum $Utility_{PPS}$ Configuration, Multiple Adversary Example Problem 1, Two Adversary Teams

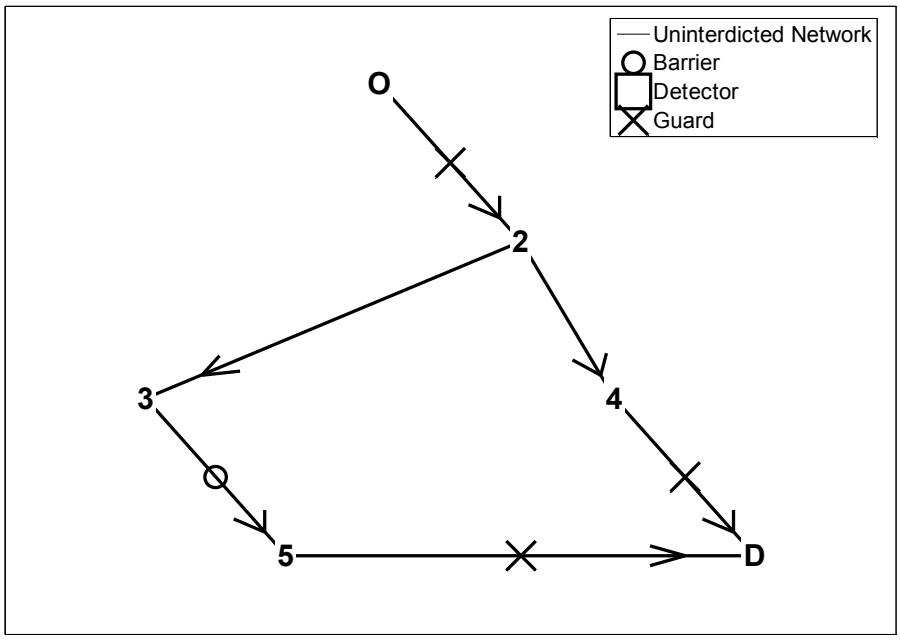


Figure 5-8: Maximum $Utility_{PPS}$ Configuration, Multiple Adversary Example Problem 1, Three Adversary Teams

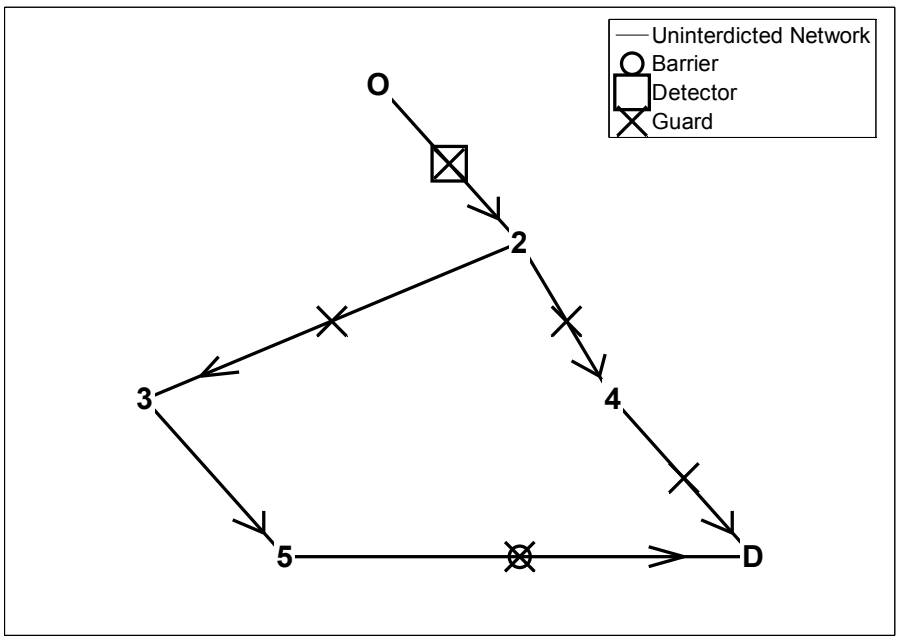


Figure 5-9: Maximum $Utility_{PPS}$ Configuration, Multiple Adversary Example Problem 1, Four Adversary Teams

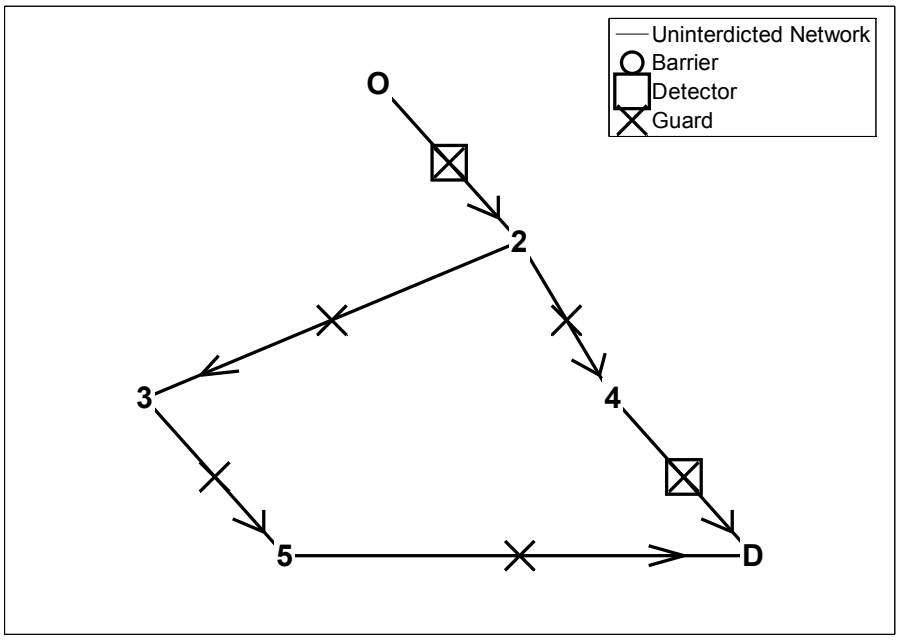


Figure 5-10: Maximum $Utility_{PPS}$ Configuration, Multiple Adversary Example Problem 1, Five Adversary Teams

As is expected, the resultant performance of the security system worsens as the number of adversary teams increases. This makes sense as more adversary teams are likely to be more successful in defeating a given security system. In turn, this requires a greater response force to combat an increased adversary force. It is interesting to note that, as the number of adversaries increases in Figure 5-5, the highest $Utility_{PPS}$ point of each scenario shifts further from the origin (up and to the right), meaning that a greater cost is required to achieve the same level of performance. In order to combat the low performance of the PPS against a greater number of adversaries, the facility owner can experiment with allowing multiple guards on each arc, as discussed in Section 4.5.1. An experiment showing the results of allowing two guards per arc for a two team analysis is shown below. As expected, allowing two guards per arc results in a higher $Utility_{PPS}$ value (as indicated by the two guards per arc curve being lower and to the left of the one guard per arc curve).

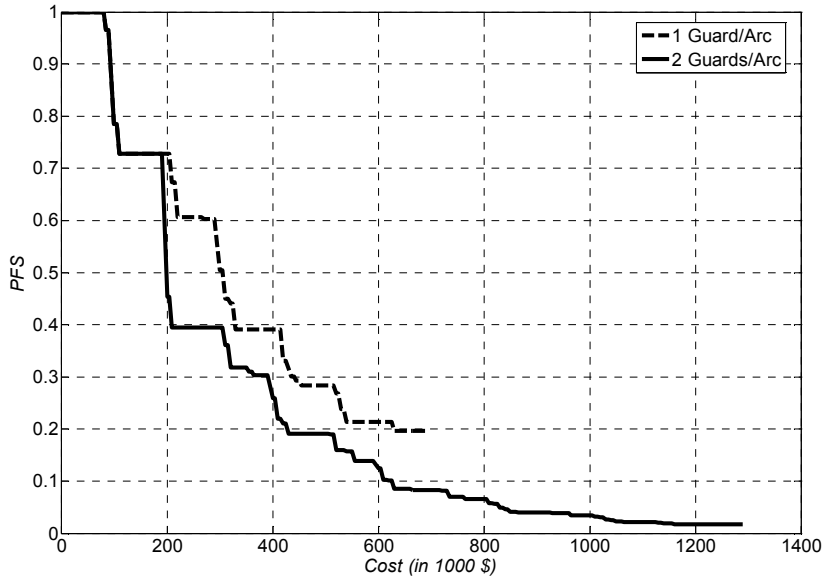


Figure 5-11: P_{FS} Vs. Cost (Two Guard Limit), Two Adversary Example Problem 1

CPU times and ratios of safeguards optimization and critical path selection are summarized in Table 5-2 below. The CPU used to run these experiments is a Dell Inspiron 5100, 2.8 GHz with 768 MB of RAM. As can be expected, the computational effort increases significantly as the number of adversaries increases.

Table 5-2: CPU Time Summary, Multiple Adversary Example Problem 1

	Time (s)	***Ratio (Relative to 1 Team)***
1 Team	25.0	1
2 Teams	160.2	6.41
3 Teams	603.1	24.12
4 Teams	1729.2	69.17
5 Teams	3729.1	149.16

The information presented in Table 5-2 is shown graphically in Figure 5-12 in order to demonstrate the exponential rise in computational time as the number of adversary teams increases. This figure is not meant to act as a deterrent to the analyst. The intent is to reinforce the understanding that the complicated analysis and in-depth information generated by undertaking a multiple team adversary analysis comes with the price of increased computational effort. However, use of FORM made this analysis possible, whereas Monte Carlo simulation would have been impossible.

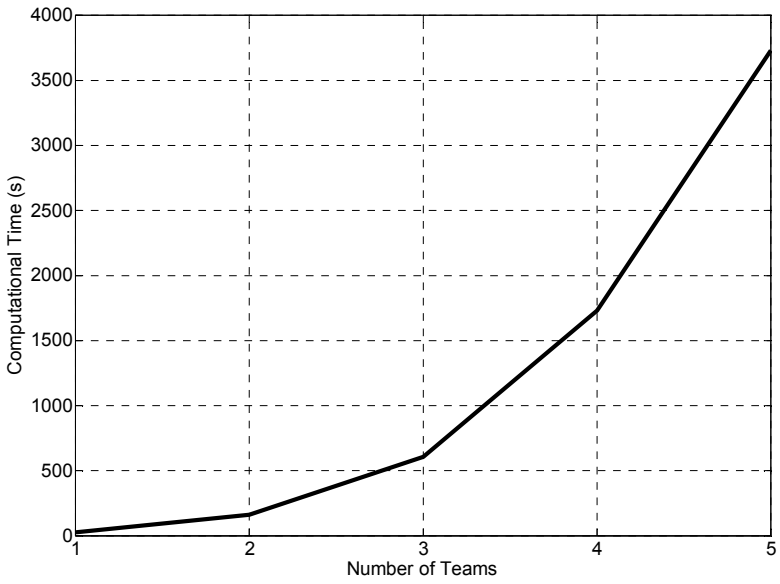


Figure 5-12: CPU Time Comparison, Multiple Adversary Example Problem 1

5.3 Example Problem 2

In this example, Problem 2 from Section 4.6 is revisited. Figure 5-13 shows the network from this problem. A new system design was attempted for 1, 2, 3, 4, and 5 teams in order to show the difference in performance among these different scenarios. Observe that the results from Section 4.6 are utilized here as the results for a single team. The analyses for 4 and 5 teams, however, did not succeed due to computational limitations as a result of the large number of potential attack scenarios present when analyzing more than one team. As the number of teams increase, this value increases exponentially, as demonstrated in Figure 5-12. Thus, it is inferred that this methodology is only effective for new facility design for real-world problems for up to three teams. This conclusion is again evaluated in Chapter VI, when the hypothetical facility is analyzed. However, the proposed methodology still represents an improvement over current stochastic network interdiction techniques[26,59,81,94] (refer to Section 3.3) which can

only address attacks by a single adversary and for a limited objective function. Additionally, the use of FORM, as discussed in the previous section, makes this analysis possible, whereas Monte Carlo based methods prevent these problems from being analyzed.

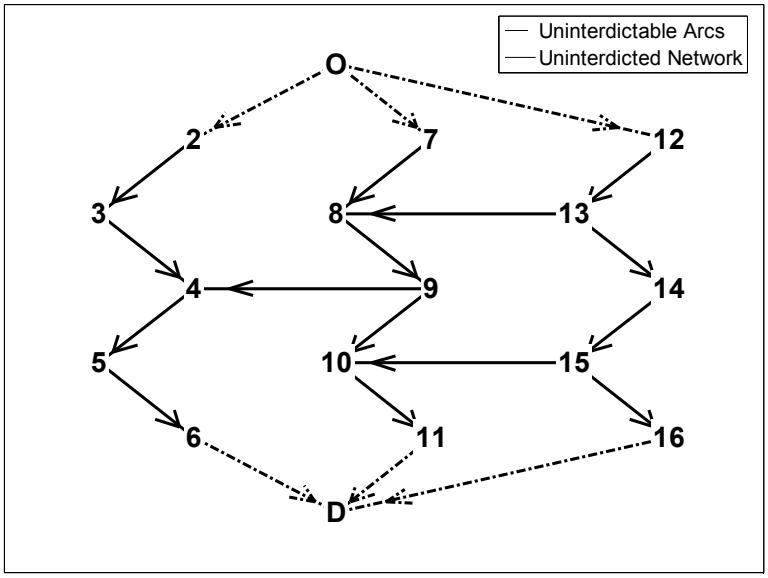


Figure 5-13: Multiple Adversary Example Problem 2

Figure 5-14 shows the *Cost* vs. P_{FS} curve for this experiment.

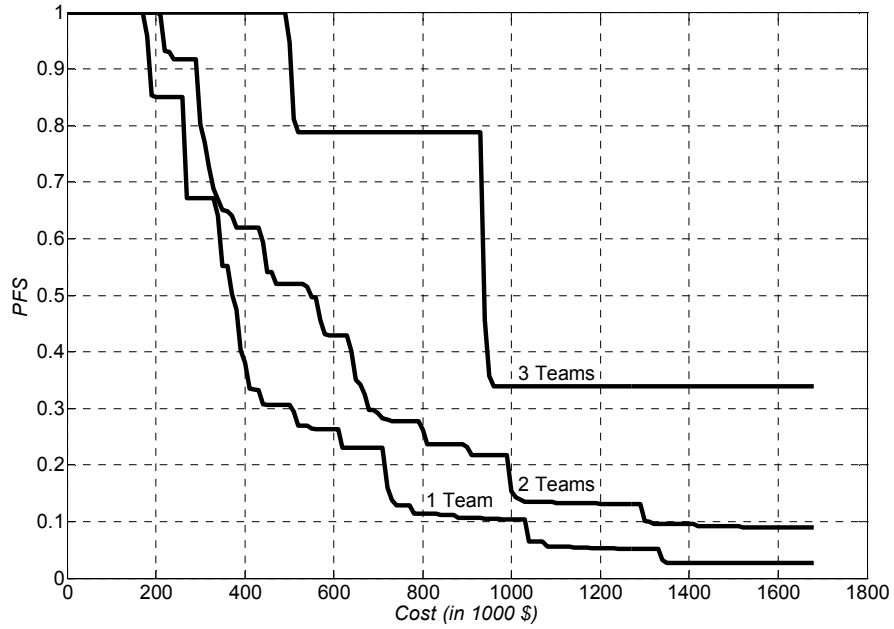


Figure 5-14: P_F Vs. Cost, Multiple Adversary Example Problem 2

As is expected, the resultant performance of the security system worsens as the number of adversary teams increases. The performance worsens at a quicker rate with multiple teams when compared with Example Problem 1. This may be due to the fact that there are more attack paths in this problem and, as a result, it is more difficult to defend against all of these potential attacks. Again, in order to combat the low performance of the PPS against a greater number of adversaries, the facility owner can experiment with allowing multiple guards on each arc, as discussed in Section 4.5.1 and shown again for the previous example problem.

CPU times and ratios of safeguards optimization and critical path selection are summarized in Table 5-3 below. The CPU used to run these experiments is a Dell Inspiron 5100, 2.8 GHz with 768 MB of RAM. As can be expected, the computational

effort increases significantly as the number of adversaries increases. The three team analysis in this case takes 29.5 hours to complete.

Table 5-3: CPU Time Summary, Multiple Adversary Example Problem 2

	Time (s)	***Ratio (Relative to 1 Team)***
1 Team	449.15	1
2 Teams	11674	25.99
3 Teams	106,150	236.33

The information presented in the above table is shown graphically in Figure 5-15 in order to demonstrate the exponential rise in computational time as the number of adversary teams increases.

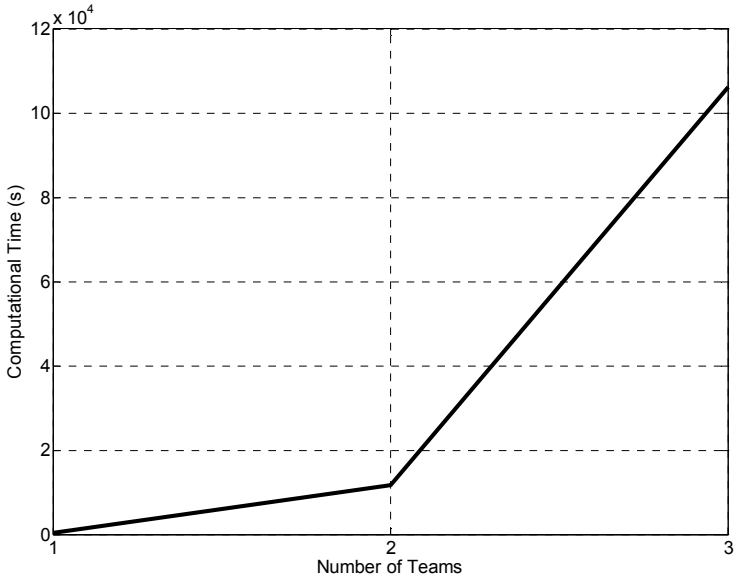


Figure 5-15: CPU Time Comparison, Multiple Adversary Example Problem 2

5.4 Summary

This chapter utilized the computational methods proposed made in Chapter IV to develop a methodology to handle multiple adversary teams in facility protection system optimization. In doing so, the problem's complexity increases significantly. The presence of multiple teams complicates 1) modeling of adversary attack strategies, including simultaneous and sequential adversary attacks and 2) development of a complex coupled implicit objective function that includes calculation of overall system-level reliability. These complications result in a significant increase in the computational effort required to design a facility protection system when compared with the single team methodology presented in the previous chapter. This chapter included discussion of all of these concepts. It was then proven useful through demonstration on two example problems. Chapter VI demonstrates the proposed methodology on a problem of real world complexity.

CHAPTER VI

DEMONSTRATION OF METHODOLOGY

6.1 Hypothetical Facility Overview

The final goal of this study is to assess the effectiveness of the proposed facility protection methodology by applying it to a problem modeled after a real world scenario. Garcia has provided a hypothetical facility[41], designed to closely emulate operations at a real critical facility, for use in testing the methodology developed in Chapters II-V. The fictional facility, Hartley International Transportation Hub (Hartley Hub), is located in a medium-sized city in the southwestern United States (designed to emulate Albuquerque, NM), designed for the year 2010. The hub contains an airport, a rail cargo center, a trucking center, an air cargo terminal, a secure cargo area, and a police substation. Each of these individual areas has its own security force.

A map of the area surrounding Hartley Hub is provided below:

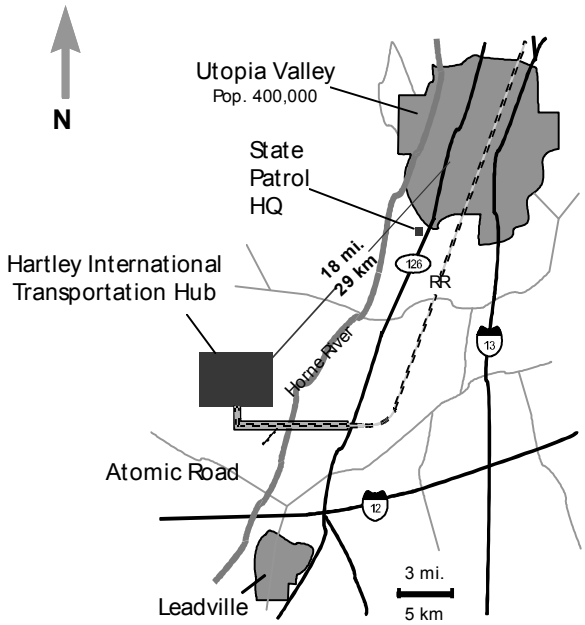


Figure 6-1: Map of Hartley Hub[41]

A map detailing the response force locations at Hartley Hub is provided below:

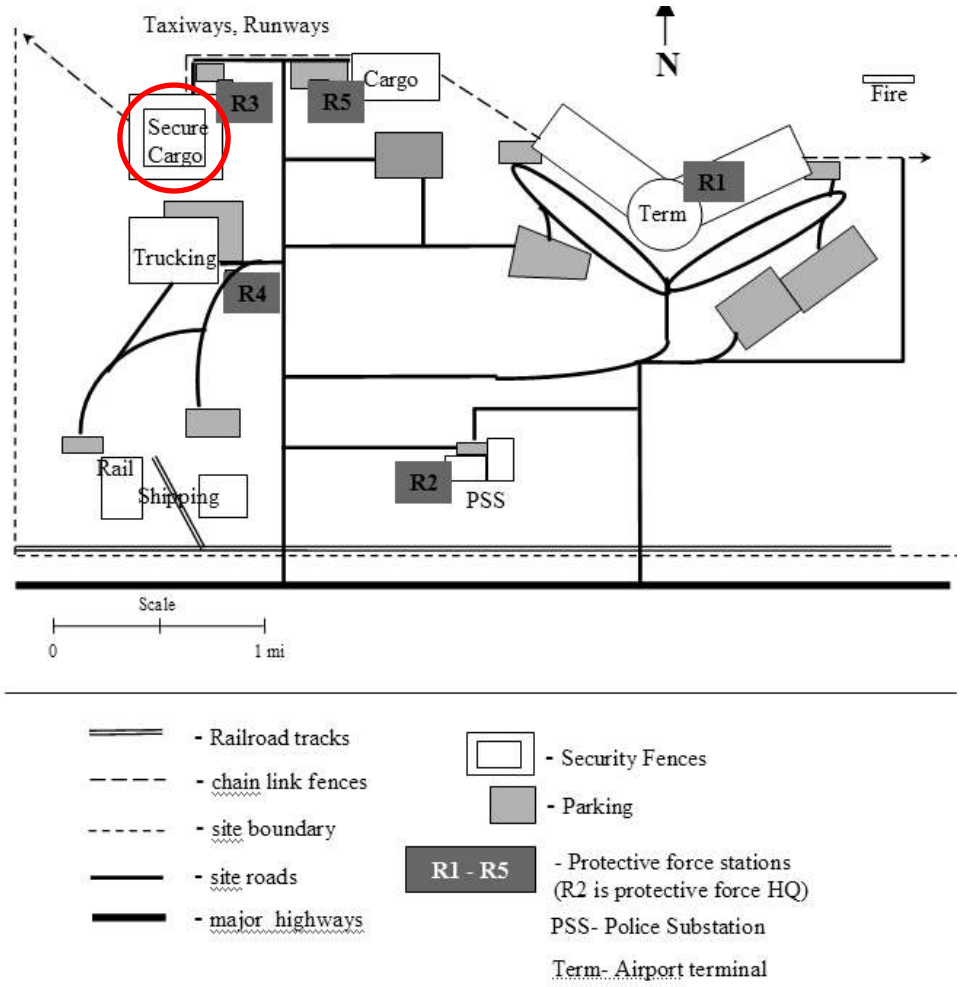


Figure 6-2: Hartley Hub Response Force Locations[41]

In Hartley Hub, the area which the facility protection optimization focuses on is the Secure Cargo Area (SCA), which is shown in detail in Figure 6-3. This figure includes dimensions which are used to calculate travel times for the facility network. Travel times (by foot) are calculated based on assuming a sprinter can run at an average sprint speed of 80% of the current world record for the 100 m dash[58] (9.77 s, or 10.235 m/s). This corresponds to a sprint speed of 8.19 m/s in open areas. Travel times (by car) are calculated based on assuming vehicle drivers travel at 20 miles/hr within the

perimeter of the facility, or an average driving speed of 8.94 m/s in open areas. Travel times are then calculated accordingly.

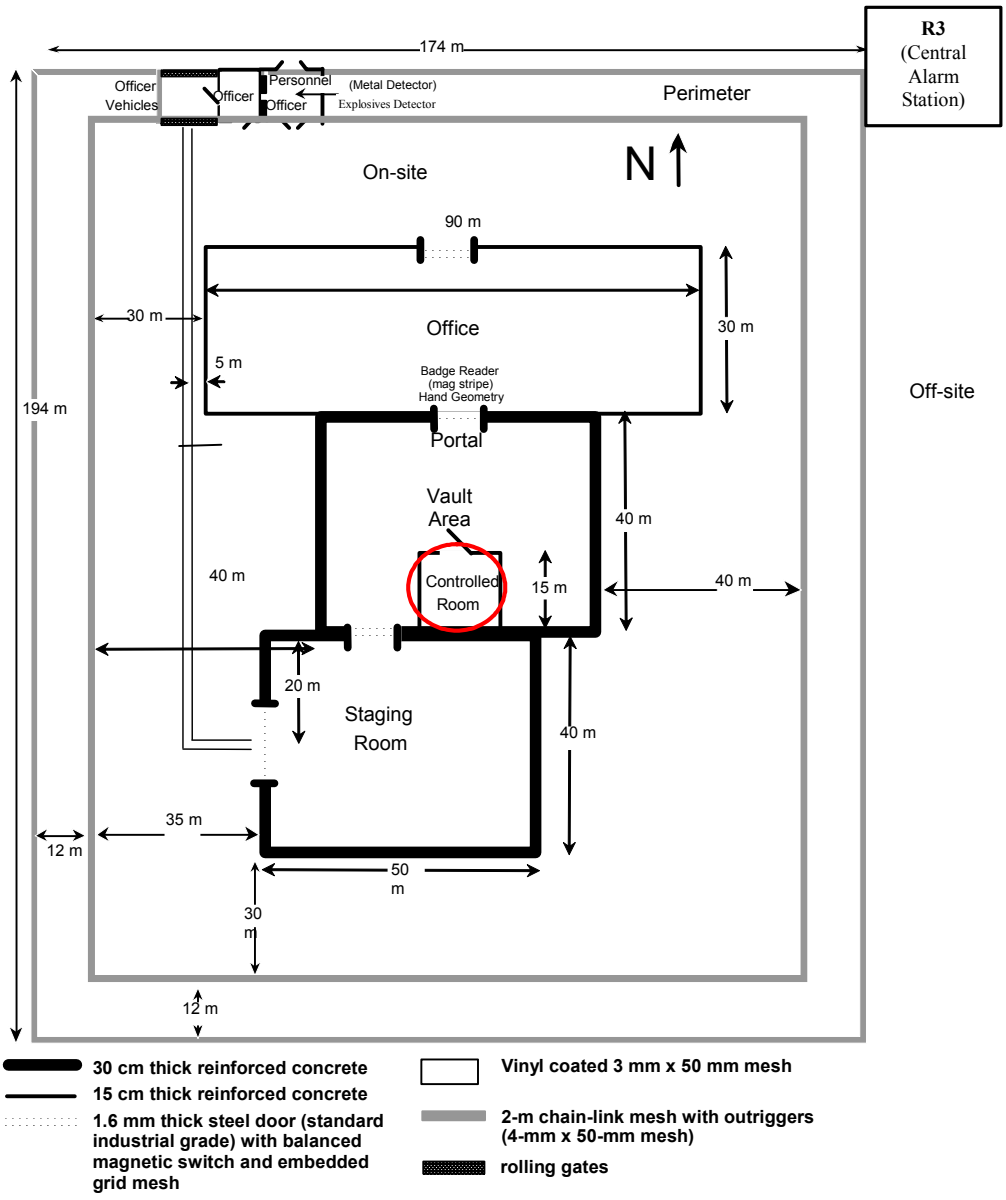


Figure 6-3: Secure Cargo Area (SCA)

6.2 Hypothetical Facility Operations

The SCA is open Monday through Friday from 5 AM until midnight. The facility is intended to be a temporary storage location for items such as prototype microelectronics, precious metals or gems, drugs seized as evidence or money being taken out of circulation, considered high value assets. Biological agents, such as anthrax, Ebola virus samples, and some radioactive substances are stored in a special controlled room within the SCA. Figure 6-4 and Figure 6-5 show the physical protection systems elements in place for the exterior and interior of the SCA, respectively.

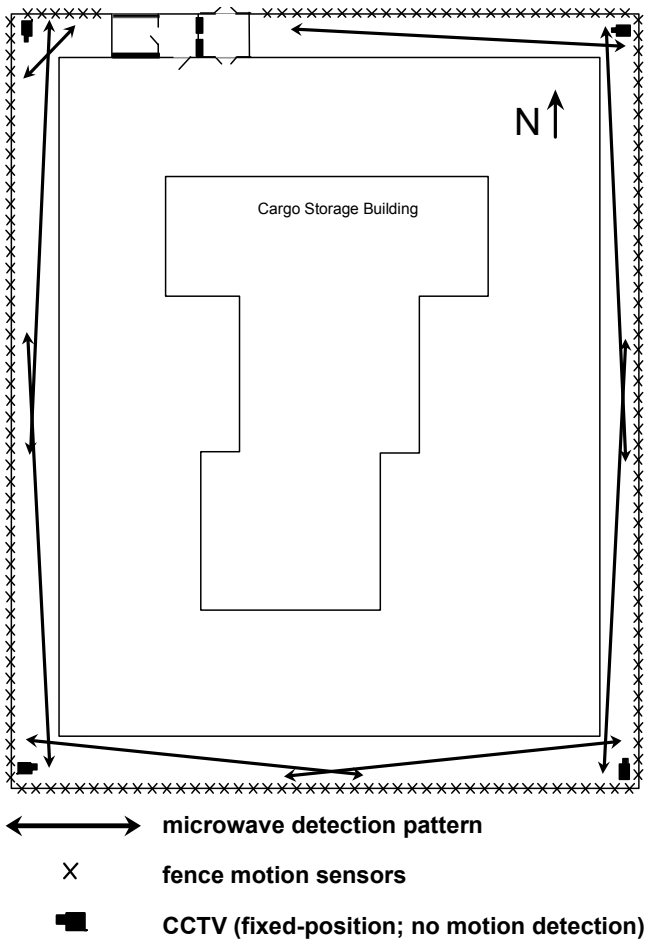


Figure 6-4: Secure Cargo Area —Exterior Protection Plan[41]

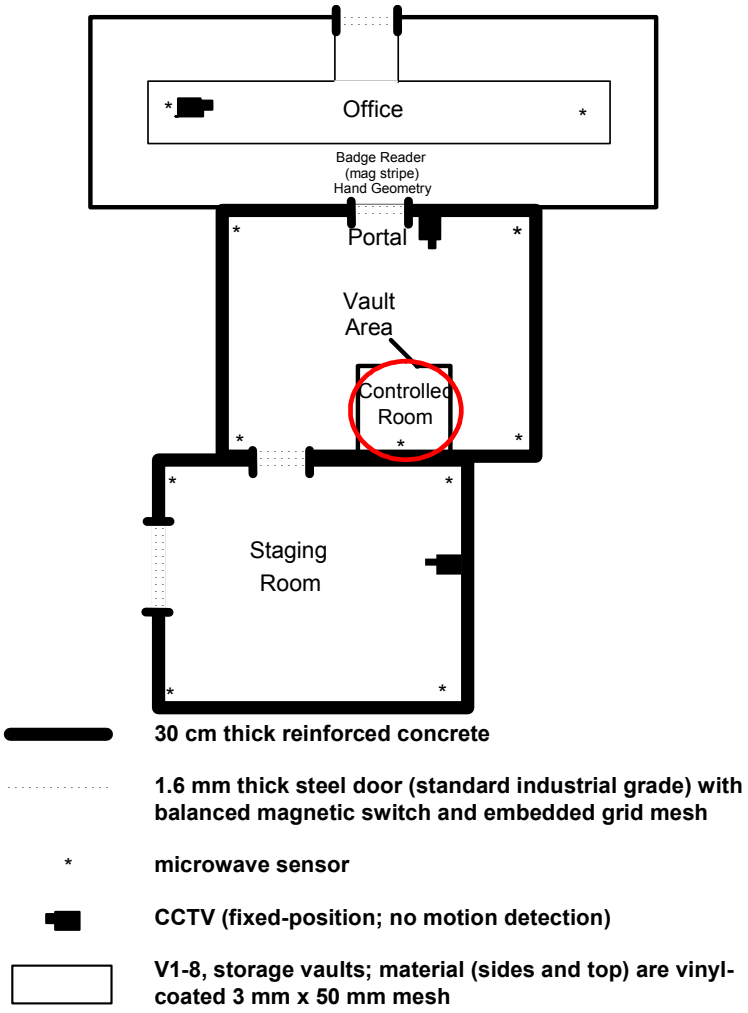


Figure 6-5: Secure Cargo Area —Exterior Protection Plan[41]

The security force at the SCA consists of three security officers during normal operating hours, while two officers are on duty when the facility is closed. During normal operations, one officer operates the central alarm system, one operates the personnel entry, and one watches the vehicle entry portal. When the facility is closed, one officer operates the alarm system and one is on random patrol. Entering personnel must pass through a personnel portal, which consists of a badge exchange, metal detector

and an explosives detector. All entering and exiting vehicles are searched via a vehicle portal.

The Cargo Storage Building at the SCA includes a Vault Area, which contains a Controlled Room, which holds biological, chemical, and radiological samples. The Vault Area may be accessed either through a door at the right off the Staging Area or through a door between the Office Area and Vault Area. The second door is opened through a magnetic stripe badge reader and a hand geometry sensor. No material (such as radiological material from the vault) may pass through this door; all material must pass through the Staging Area. Within the Vault Area is a Controlled Room, which holds environmental chambers for the storage of biological and radiological goods.

The Staging Area of the SCA is accessed through a vehicle-sized roll-up door. During operating hours, the roll-up door remains open and unlocked and the balanced magnetic switch that acts as a detector is set to "access", preventing it from detecting any intruders. Outside of the SCA is a dual-gate vehicle barrier which requires independent gate opening.

There are approximately 60 employees at the SCA. The workforce includes a Facility Manager, a Security Manager, an Operations Managers, engineers and technicians, material handlers, clerks, secretaries, custodians, and security officers. Also, on occasion, subcontractors are allowed to enter the facility to perform maintenance and emergency repair work, such as HVAC, computers, machinery, etc.

The primary response force for the SCA is the local police, whose headquarters are located at R2 (refer to Figure 6-2). The local police are also responsible for the protection of the airport, the remainder of Hartley Hub, and other areas of the city.

Accordingly, it takes them 20 minutes to respond to an attack on the SCA. When alerted by the dispatcher that the SCA is under attack, all eight police officers who patrol the city are instructed to return to police headquarters. The first four to arrive at headquarters become the response force for the incident, while the remaining officers are released back to normal patrol. The officers collect their gear, discuss their plan, and proceed to the SCA as a group. Once they arrive at the SCA, they contact the local SCA security officer in accordance with established procedures. The response procedure for SCA Response Force is as follows:

1. An alarm in the SCA is detected.
2. A local officer (in the control room) assesses the alarm to determine whether or not it is an actual intrusion. The action takes 0.5 minutes.
3. The local officer reports the incident to the police force at R2. This action takes 0.3 minutes.
4. The local officer communicates to other R3 officers to take action. This action takes 1.5 minutes.
5. R2 alerts response personnel to report to R2.
6. Response force at R2 collects equipment.
7. Response team travels to R3. Actions 5-7 take 11.7 minutes.
8. R2 officers deploy and proceed with interruption of adversary. This action takes 1.5 minutes.

If only local response is necessary, only actions 1-4 and 8 are required. If off-site response is required, all actions are necessary. In order to simplify calculations, local response actions are combined into one individual action with local action being

described as a truncated normal with mean of the travel time to reach the destination plus 2.3 minutes (the result of actions 1-4). Offsite police force actions are combined into one action described as a truncated normal with mean of 15.5 minutes and a standard deviation of 1.55 minutes.

The staffing associated with the security personnel is summarized below:

Table 6-1: Response Force Staffing

		No. of Officers	
Response Force	***Description***	***Workdays***	***Nights and Weekends***
R1	Airport Terminal Alarm Station	3	3
R2	Police Substation (Response Force Headquarters)	10*	10
R3	Secure Cargo Area	3	2
R4	Trucking Center	1	1
R5	Cargo Area	1	1
		18	17

*Includes site response team

It is assumed that the only security personnel that respond to an incident in the SCA are the R3 response force (on-site response) and the R2 response force (off-site police, if necessary).

Table 6-2 shows the performance data assumed for the safeguards in place in the existing facility described in the hypothetical facility[41]. These values are assumed in order to avoid potential classification issues. Cost data are assumed to be the net present value of life cycle costs, including installation, operations, and maintenance.

Table 6-2: Safeguards Data

	Element	***P_D***	***Delay Time (s)***	***Cost ($100K)***
Detectors	*CCTV*	.1	0	1
	CCTV with Motion Capture	.5	10	5
	Microwave Sensor	.5	30	5
	Reinforced Door with Balanced Magnetic Switch	.25	10	1
	Personnel Portal	.5	60	5
	Vehicle Portal	.5	60	5
Response Force	*On-Site Guard #1*	.85	60	20
	On-Site Guard #2	.85	60	20
Barriers	*Reinforced Concrete Walls*	0	150	1
	Chain Link Fence with Razor Wire	0	60	1
	Fence Motion Sensor	.25	30	5
	Anti-Vehicle Barrier	0	90	1

Figure 6-3 was transformed into a task-based network topology (shown in Figure 6-6 with arc numbers shown in parentheses) in order to utilize the methodology developed in Chapters II-V. While an attack involving theft of materials in the controlled room (refer to Figure 6-3) is possible, scenarios which involve sabotage are more conservative as the potential for response force personnel to interrupt the adversary decreases significantly since there are less tasks for the adversary to perform to achieve mission success. As a result, analysis of the hypothetical facility focuses on sabotage scenarios. The performance level achieved against a sabotage scenario provides a lower limit for performance achieved in protecting against a theft scenario.

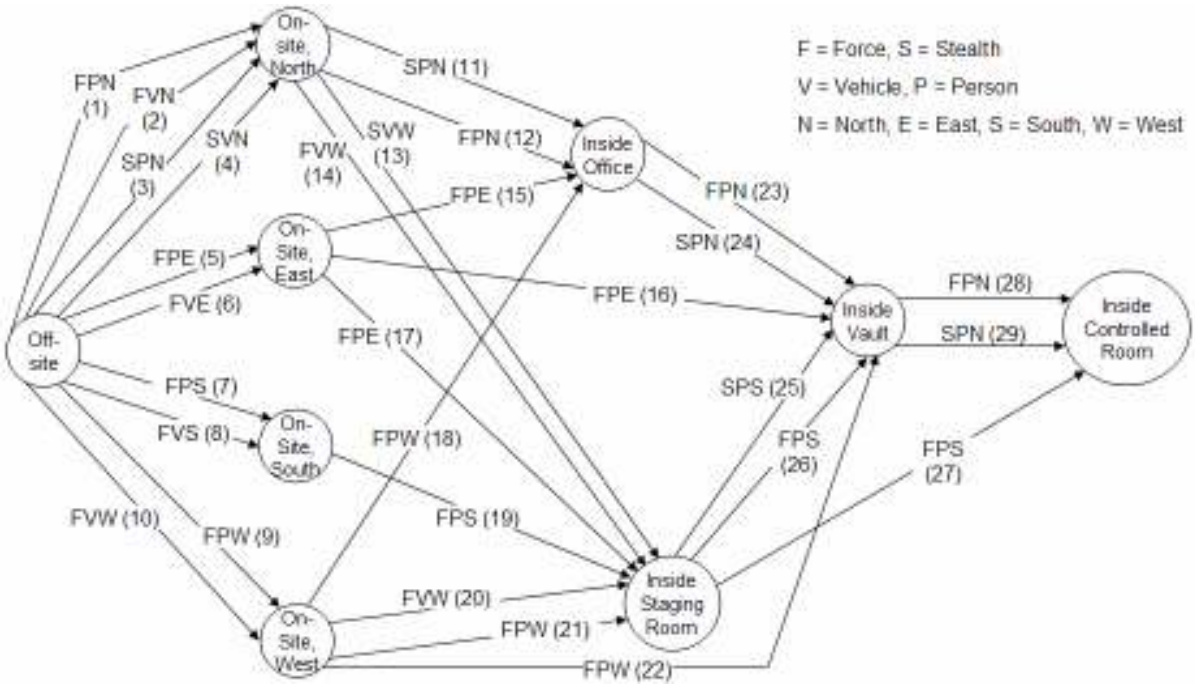


Figure 6-6: Hypothetical Facility Network for Sabotage

The following sections discuss solution of this problem in three steps: 1) evaluation of the effectiveness of the existing security system, 2) upgrades analysis, and 3) optimal design of a new security system. Specific Design Basis Threat[1,2] data for a particular facility is classified, so the adversarial threat was assumed to be one team for the existing facility analysis and new facility design (for illustration purposes) and two teams for the upgrades analysis (to demonstrate that a multiple team analysis could be undertaken on a realistic problem). (Note that the methodology presented in Chapter V showed success with all example problems up to a threat of three adversary teams). FORM-based optimization is used; Monte Carlo based optimization is deemed too computationally intensive (given the results of earlier example problems) and therefore, not practical and not utilized as an approach to solve these problems.

6.3 Existing Facility Analysis

An analysis of the provided hypothetical facility provides the following safeguards plan (where a 1 indicates installation of the specified safeguard type on the individual arc and a 0 indicates the safeguard was not installed), where arc numbers correspond to those in Figure 6-6:

Table 6-3: Hypothetical Facility Baseline Safeguards

	Arc #																												
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*	*21*	*22*	*23*	*24*	*25*	*26*	*27*	*28*	*29*
CCTV	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1
CCTV with Motion Capture	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Microwave Sensor	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Reinforced Door with Balanced Magnetic Switch	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	1	1	1	1	0	0	0
Personnel Portal	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Vehicle Portal	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
On-Site Guard #1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
On-Site Guard #2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reinforced Concrete Walls	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Chain Link Fence with Razor Wire	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fence Motion Sensor	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Anti-Vehicle Barrier	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The existing facility analysis was performed for both day and night operations in order to compare the results. The cost for the baseline safeguards is $18,300,000. The day P_{FS} = .3743, while the night P_{FS} = .503, due to a reduced guard presence.

6.4 Upgrades Analysis

An upgrades analysis was undertaken for the facility using the existing safeguards plan, with a budget of $5 million for 1 and 2 teams. This analysis resulted in the *Cost* vs. P_{FS} plots shown in Figure 6-7. It is interesting to note that the highest $Utility_{PPS}$ safeguards system does not spend the entirety of the budget provided in either upgrades analysis. While this was also shown in the example problems presented in Chapters IV and V, it is much more significant when proven on a large scale problem. The highest $Utility_{PPS}$ value for the single team scenario occurs at a cost of $20.3 million, whereas the upgrades budget allows for a total expense of $23.3 million, meaning that this methodology would result in a potential savings of $3 million in calculating the highest $Utility_{PPS}$ combination of cost and performance for this threat for this facility. In a practical facility, however, the facility owner may require a safeguards plan that ensures optimal performance given a particular budget. In this case, the analyst can explore the possibility of P_{FS} decrease due to additional safeguards expenditures using Figure 6-7.

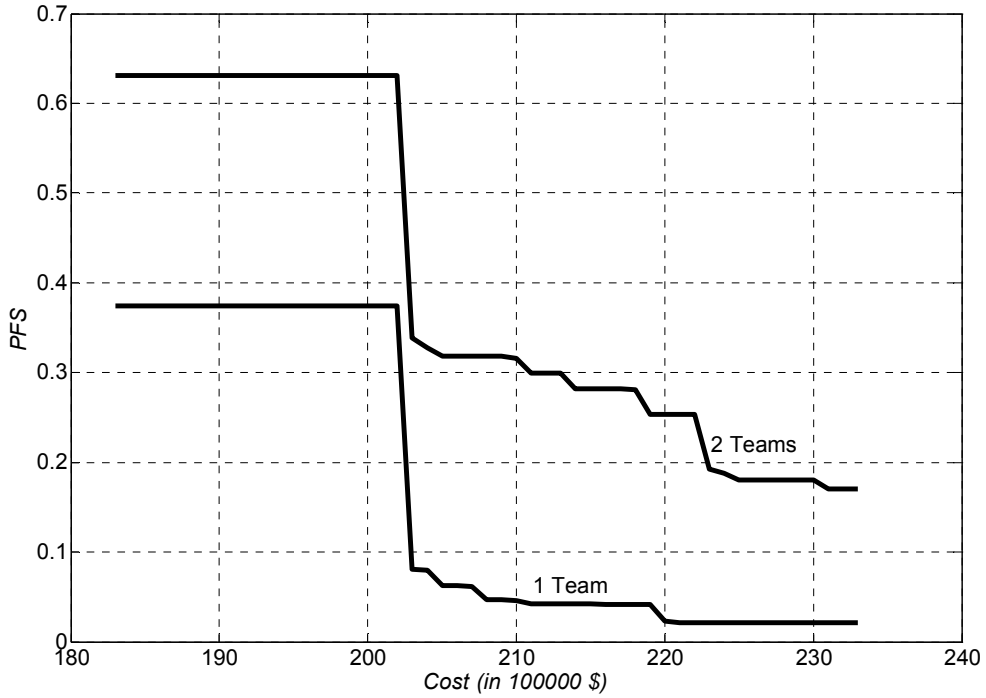


Figure 6-7: Upgraded P_{FS} Vs. Cost, Hypothetical Example

6.5 New Safeguards Design

The results of this analysis are shown below. The highest $Utility_{PPS}$ point is found to be at *Cost* = $6.4 million, with a P_{FS} = .060591. This represents a significant savings over the baseline facility. Two and three team analyses were attempted using the limit-constrained multiobjective optimization method (shown in Section 4.2.1). Even with the reduced computational requirement of this method, the analysis was unable to complete due to computational limitations due to: 1) the number of scenarios identified using critical path enumeration is prohibitively large to solve this problem with a multiple adversary team threat, and 2) the optimization algorithm used to perform the safeguards optimization could not efficiently perform the large scale multiobjective optimization required for the problem formulation in Eq. (3-3).

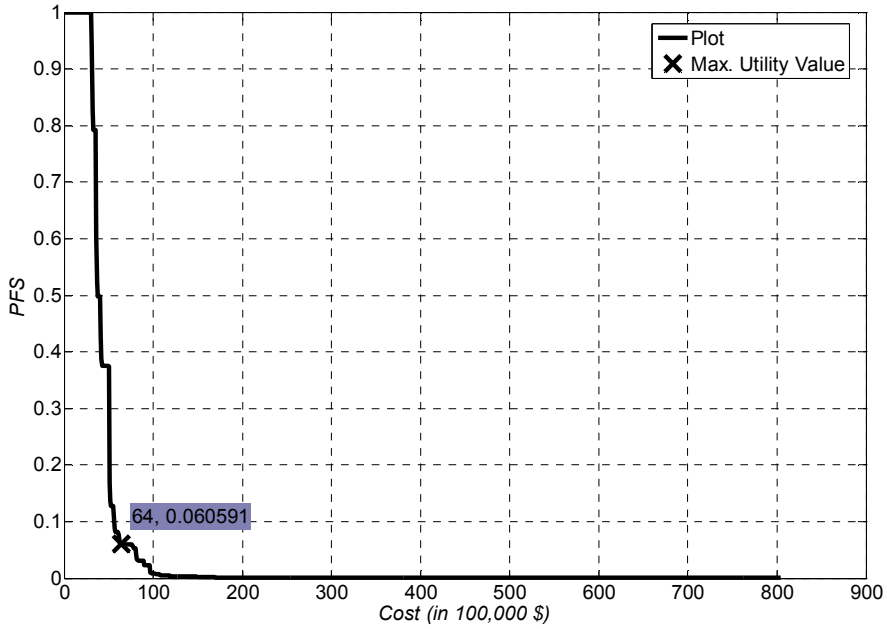


Figure 6-8: New P_{FS} Vs. Cost, Hypothetical Example

6.6 Computational Effort

The computational times comparing the different problem formulations are shown in Table 4-4. The CPU used to run these experiments is a Dell Precision 650, 2.4 GHz with 1.00 GB of RAM. As can be expected, a large-scale problem requires significantly computational effort (even with FORM) in order to achieve results. With Monte Carlo simulation, obtaining these results would not have been possible due to the required computational effort.

Table 6-4: CPU Time Summary, Hypothetical Example

	CPU Time (s)
Existing Facility Analysis	44.77
Upgrades Analysis (1 Team)	4881
Upgrades Analysis (2 Teams)	74,074
New System Design	537,480

6.7 Summary

The focus of this chapter was to assess the methodology developed throughout this study on a problem of real world complexity. It was found to be successful in doing so using FORM, illustrating that the methodology can be used for new facility design, upgrades analysis, and existing facility analysis. Monte Carlo Simulation was found to not be a practical method for use in this methodology.

CHAPTER VII

CONCLUSION AND FUTURE WORK

7.1 Summary

Current techniques for design and analysis of critical facility protection systems lack the ability to handle large scale problems, comprehensive objective functions, and multiple team threats. This study makes the following contributions to these areas:

- The methodology combines analysis techniques of adversary path analysis with the design capabilities of graph-theory based network interdiction, allowing for the development of a methodology that supports new facility design, upgrades analysis, and existing facility analysis.
- The methodology develops an in-depth evaluation of system effectiveness, incorporating conflict between adversarial and facility guard forces and timely detection, concepts that were previously omitted from facility protection methodologies.
- The methodology utilizes an efficient first-order reliability method based approach (instead of traditional Monte Carlo based methods) to incorporating stochastic behavior.
- The methodology presents a novel approach to decoupling adversary critical path selection and safeguards optimization.

- All of the methodological contributions of this study allow for realistic problems to be analyzed, although it may only be possible to analyze two or three teams at a larger facility. Further, this capability is demonstrated on a realistic problem, which has not been possible for previous methodologies.
- The developed single adversary team methodology is extended to a multiple team methodology. This is a significant development of this study as a methodology to combat multiple adversary team attacks against a facility does not currently exist.

It is clear from the examples that existing facility analysis and upgrades analysis are straightforward and not computationally prohibitive. However, the computational burden is significant for a new system design. This is a result of the number of combinations that are generated for new safeguards analysis due to the large number of enumerated paths, scenarios (with multiple teams), and potential budget values. With an existing facility analysis or upgrades analysis, the number of potential budget values drops significantly (while the other quantities remain the same), significantly reducing the computational effort required to perform the analysis. FORM-based methods utilized in this study make new system design possible, whereas previous Monte Carlo based methods make the solution of large scale problems computationally impossible.

While the developed methodology has made several contributions to physical protection system design and analysis, several limitations remain. Cost is treated throughout this study as a deterministic, linear variable (that is, the cost of two safeguards is twice as much as one safeguard, and these costs are known with 100% certainty). Cost may not be certain, however, and while assuming cost is a deterministic variable is an appropriate first step in analysis, the methodology should be extended to include

uncertainty in cost. For example, discounts exist for purchasing multiple items in bulk and the effects of these pricing strategies should be explored.

Another limitation of the proposed methodology is that the network and its components are considered static. That is, their performance does not change over time. Of particular interest are response force and adversary actions. The response force in this study was treated as having a particular patrol area. While this is a good first estimate, in reality response force personnel are dynamic individuals whose patrol areas are likely to change as information becomes available, e.g. one guard receives a distress call from another guard. This dynamic response should be incorporated through agent based modeling. Adversary actions are assumed to be known based on adversary paths, but realistic scenarios involve adversaries potentially changing paths during their attack. Agent based modeling can also be utilized to model adversary actions.

A final limitation of this methodology is that it addresses only sabotage scenarios that require access to the sabotage target. This methodology does not consider theft scenarios or scenarios in which sabotage may occur at a distance (i.e. detonating a bomb at the perimeter of a facility that damages a building inside the facility). These scenarios should be explored in future work.

Additionally, the following section discusses areas that are not limitations of the current methodology, but could be explored for future work.

7.2 Future Work

While this study has made significant progress in the area of facility protection optimization uncertainty, there are several extensions that should be explored in future work:

- Input data should be analyzed before implementing this methodology. One possible approach for this would be through elicitation of expert opinion. Experts can help to quantify some of the unknown parameters that are used in the model, such as the simultaneous team time factor. Additionally, if possible, it is suggested that the methodology's assumptions (i.e. *STTF* or *HTTF*) be tested experimentally (i.e. comparing a sample task completion time for multiple teams with a single team). While this work may be classified, its findings could prove to be very important in proving this methodology to be useful.
- Further development of critical path enumeration techniques should be explored. A smaller number of critical paths for large scale scenarios could significantly reduce the computational burden of this methodology on larger problems.
- Potential insider attacks need to be included in this methodology. While a significant amount of work has been done to design physical protection systems to prevent outsider attacks, the disabling of a particular PPS component by an insider may render the system useless. Additionally, insiders can perform crimes over extended timelines, such as stealing extremely small portions of a nuclear material over a period of time of many years. Defending against this type of attack is significantly different. A methodology needs to be developed to balance cost and performance of defending against both insider and outsider attacks.

- Finally, other problem types may be explored using the presented methodology. The first potential application is with respect to the US Border Control Initiative. In order to do this, the problem could be set up similar to the facility protection problem, which the exception being that the acceptable throughput for a border control system would not have to be zero (as it is in the methodology presented in this study). Security systems for other facilities such as airports, ports, etc. may also be explored as potential avenues for further use of this methodology. All three of these suggested applications are natural candidates for graph theory based representations given their distributed geographic layouts.

REFERENCES

1. 10 CFR 73. (2005). "Design Basis Threat." *Code of Federal Regulations*, Nuclear Regulatory Commission.

2. 10 CFR 73. (2003). "Design Basis Threat." *Code of Federal Regulations*, Nuclear Regulatory Commission.

3. Ahuja, R., Magnati, T., and Orlin, J. (1993). Network Flows: Theory, Algorithms, and Applications. New Jersey: Prentice Hall.

4. Ang, A.H-S. and Amin, M. (1967). "Studies of Probabilistic Safety Analysis of Structures and Structural Systems," *Structural Research Series No. 320*, University of Illinois, Urbana.

5. Ang, A.H.-S. and Tang, W.H. (1984). Probability Concepts in Engineering Planning and Design, Vol. II: Decision, Risk, and Reliability, Wiley, New York.

6. Aronson, J.E. (1989). "A Survey of Dynamic Network Flows." *Annals of Operations Research*, 20, 1–66.

7. Assad, A.A. (1978). "Multicommodity network flows - a survey." *Networks*, 8(1), 37-91.

8. Bellman, R.E. (1957). Dynamic Programming. Princeton: Princeton University Press.

9. Benders, J. F. (1962). "Partitioning Procedures for Solving Mixed-Variables Programming Problems." *Numerische Mathematik*, 4, 238–252.

10. Bennett, H.A. (1972). "The "EASI" Approach To Physical Security Evaluation SAND76-0500." Sandia National Laboratories, Albuquerque, NM.

11. Birge, J.R. and Louveaux, F. (1997). Introduction to Stochastic Programming. 54-61, New York: Springer.

12. Box, G.E.P. and Cox, D.R. (1964). "An Analysis of Transformation." *Journal of the Royal Statistical Society*, Series B, 26, 211-252.

13. Boyle, P., Broadie, M. and Glasserman, P. (1998). "Monte Carlo methods for security pricing." *Journal of Economic Dynamics and Control*, 3, 1267-1321.

14. Breitung, K. (1984). "Asymptotic Approximations for Multinormal Integrals." *Journal of Engineering Mechanics, ASCE*, 110(3), 357-366.

15. Brown, R.H. (1963). “Theory of Combat: The Probability of Winning.” *Operations Research*, 11(3), 418-425.

16. Brusco, M.J. and Stahl, S. (2005). Branch-and-Bound Applications in Combinatorial Data Analysis. New York: Springer.

17. Castro, J. and Nabona, N. (1996). “An implementation of linear and nonlinear multicommodity network flows.” *European Journal of Operational Research*, 92, 37-53.

18. Cemgil, A. (2003). “Graph Layout Generation Package” [Online]. [Accessed January 9, 2007]. Available from World Wide Web: http://www-sigproc.eng.cam.ac.uk/%7Eatc27/.

19. Censor, Y. (1977). "Pareto Optimality in Multiobjective Problems." *Appl. Math. Optimization*, 4, 41-59.

20. Cerbone, D. and Noe, T. (1996). “Multiobjective Optimum Design.” [Online]. [Accessed December 19, 2006]. Available from World Wide Web: http://www.glue.umd.edu/~azarm/optimum_notes/multi/multi.html.

21. Charnes, A. and Cooper, W.W. (1961), Management Models and Industrial Applications of Linear Programming. Vol. I, Wiley, New York.

22. Clemen, R.T. and Reilly, T. (2001). Making Hard Decisions With DecisionTools. Duxbury: Pacific Grove, CA, 180-181.

23. Clemens, P.L. (2002). “Fault Tree Analysis” [Online]. [Accessed November 21, 2005]. Available from World Wide Web: http://www.sverdrup.com/safety/fta.pdf.

24. Chiralaksanakul, A. and S. Mahadevan. (2005). "First-Order Methods for Reliability-Based Optimization." *ASME Journal of Mechanical Design*, 127(5): 851-857.

25. Choi, K.K and Youn B.D. (2001). “Hybrid Analysis Method for Reliability-Based Design Optimization.” *Proceeding of ASME: 27th Design Automation Conference*, Pittsburgh, Pennsylvania.

26. Cormican, K.J., Morton, D.P., and Wood, R.K. (1998). “Stochastic Network Interdiction.” *Operations Research*, 46(2), 184-197.

27. Cornell, C.A. (1967). “Bounds on the Reliability of Structural Systems,” *Journal of the Structural Engineering, ASCE*, 93(ST1), 171-200.

28. Da Cunha, N.O. and Polak, E. (1967). "Constrained Minimization Under Vector-valued Criteria in Finite Dimensional Spaces." *J. Math. Anal. Appl.*, 19, 103-124.

29. Davis, T.A. and Sigmon, K. (2004). *Matlab Primer*. CRC Press.

30. Ditlevsen, O. (1979). "Narrow Reliability Bounds for Structural Systems," *Journal of Structural Mechanics*, 3, 453-472.

31. Ditlevsen, O. and Madsen, H. O. (1996). Structural Reliability Methods. New York, NY: Wiley.

32. Du X., Chen W. (2004). "Sequential Optimization and Reliability Assessment method for efficient probabilistic design." *Journal of Mechanical Design*, 126(2): 225-233.

33. Eppstein, D (1997). "Finding the K Shortest Paths." *SIAM Journal on Computing*, 28(2), 652-673.

34. Ferber, J. (1999). Multi-agent Systems: Introduction to Distributed Artificial Intelligence. Addison-Wesley Longman, Boston, MA.

35. Feutz, R.J. (1965). "Introduction to Fault Tree Analysis." Boeing Company Report Number D6-16182.

36. Fletcher, R. and Powell, M.J.D. (1963). "A Rapidly Convergent Descent Method for Minimization," *Computer Journal*, 6, 163-168.

37. Floyd, R.W. (1962). "Algorithm 97: Shortest path." *Communications of ACM*, 5, 345.

38. Fulkerson, D.R. and Harding, G.C. (1977). "Maximizing the minimum source-sink path subject to a budget constraint." *Mathematical Programming*, 13, 116–118.

39. Garcia, M.L. (2001). *Companion to the Design and Evaluation of Physical Protection Systems*. Butterworth-Heinemann, MA.

40. Garcia, M.L. (2001). *The Design and Evaluation of Physical Protection Systems*. Butterworth-Heinemann, MA.

41. Garcia, M.L. (2005). "Design and Evaluation of Physical Protection Systems Exercise Data Book: Hypothetical Facility." Sandia National Laboratories, Albuquerque, NM.

42. Gass, S.I. (1987). "A process for determining priorities and weights for large scale linear goal programmes." *Journal of the Operational Research Society*, 37, 779-785.

43. Gill, P.E., Murray, W., and Wright, M.H. (1981). Practical Optimization. London: Academic Press.

44. Glasserman, P. (2005). Monte Carlo methods in financial engineering. Springer.

45. Glover F. (1990). "Tabu Search: A Tutorial." *Interfaces*, 20 (4), 74-94.

46. Golden, B.L. (1978). "A problem in network interdiction." *Naval Research Logistics Quarterly*, 2, 711–713.

47. Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley.

48. Goldfarb, D. (1970). "A Family of Variable Metric Updates Derived by Variational Means," *Mathematics of Computing*, 24, 23-26.

49. Gomory, R.E. (1958). "Outline of an algorithm for integer solutions to linear programs." *Bulletin of the American Mathematical Society*, 64, 275-278.

50. Gomory, R.E. (1960). "Solving linear programming problems in integers." In Bellman, R. and Hall, M. Jr. (Eds.), *Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics* (Vol. 10, 211-215). Providence, R.I.: American Mathematical Society.

51. Gomory, R.E. (1963). "An algorithm for integer solutions to linear programs." In R.L. Graves & P. Wolfe (Eds.), *Recent Advances in Mathematical Programming* (269-302). New York: McGraw-Hill.

52. Haldar, Achintya, and Mahadevan, Sankaran. (2000). Probability, Reliability, and Statistical Methods In Engineering Design. 181-274, New York, NY: Wiley.

53. Han, S.P. (1977). "A Globally Convergent Method for Nonlinear Programming," Vol. 22, *Journal of Optimization Theory and Applications*, p. 297.

54. Hicks, M.J., Snell, M.S., Sandoval, J.S., and Potter, C.S. (1999). "Physical protection systems cost and performance analysis: a case study." *Aerospace and Electronic Systems Magazine, IEEE*, 14 (4), 9-13.

55. IAEA. (1972). "Recommendations for the Physical Protection of Nuclear Material." INFCIRC 225, Correction 1.

56. IAEA. (1999). "The Physical Protection of Nuclear Material and Nuclear Facilities." INFCIRC 225, Revision 4.

57. Ijiri, Y. (1965). Management Goals and Accounting for Control. North Holland, Amsterdam.

58. International Olympic Committee (2006). "Athletics—Current World Records."

59. Israeli, E. (1999). "System Interdiction and Defense." Ph.D. thesis, Naval Postgraduate School, Monterey, California.

60. Israeli, E. and Wood, R.K. (2002). “Shortest-path Network Interdiction." *Networks*, 40(2), 97-111.

61. Jensen, K. (1992). Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Springer-Verlag.

62. Kellerer, H., Pferschy, U., and Pisinger, D. (2004). Knapsack Problems. Berlin: Springer.

63. Kennington, J.L. (1978). “A survey of linear cost multicommodity network flows.” *Operations Research*, 26(2), 209-236.

64. Kirkpatrick, S., Gelatt Jr., C. D., Vecchi, M. P. (1983). "Optimization by Simulated Annealing." *Science*, 220 (4598), 671-680.

65. Kleder, M. (2005). “All Pairs Shortest Path Graph Solver” [Online]. [Accessed January 9, 2007]. Available from World Wide Web: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8808&objectType=file.

66. Köylüoğlu, H.U. and Nielsen, S.R.K. (1988). “New Approximations for SORM Integrals.” *Structural Safety,* 5, 119-126.

67. Kuschel, N. and Rackwitz, R. (2000). “A New Approach for Structural Optimzation of Series Systems.” In Melchers and Stewart (Eds.), *Applications of Statistics and Probability* (987-994). Balkema, Rotterdam.

68. Lanchester, F.W. (1916). Aircraft in Warfare, the Dawn of the Fourth Arm. London: Constable.

69. Lawler, E.L. and Wood, D.E. (1966). “Branch-And-Bound Methods: A Survey.” *Operations Research*, 14(4), 699-719.

70. Lawrence, C. T. and Tits, A. L. (1997). “Feasible Sequential Quadratic Programming for Finely Discretized Problems from SIP.” In Reemtsen, R. and Ruckmann, J. J. (Eds.), Semi-Infinite Programming, in the Series Nonconvex Optimization and Its Applications. Kluwer Academic Publishers.

71. Liang J., Mourelatos Z. P., Tu J. (2004). “A Single Loop Method for Reliability-Based Design Optimization.” *Proceedings of ASME Design Engineering Technical Conferences and Computer and Information in Engineering Conferences*, Salt Lake City, Utah, DETC2004/DAC-57255.

72. Lim, C. and Smith, J.C. (2007). “Algorithms for Discrete and Continuous Multicommodity Flow Network Interdiction Problems.” *IIE Transactions*, 39(1), 15-26.

73. Liu, P.L. and DerKiureghian, A. (1986). "Multivariate distribution models with prescribed marginals and covariances." *Probabilistic Engineering Mechanics*, 1(2):105–112.

74. Liu, Z. (1998). "Performance Analysis of Stochastic Timed Petri Nets Using Linear Programming Approach." *IEEE Transactions on Software Engineering*, 24(11), 1014-1030.

75. Man, K.F., Tang, K.S., and Kwong, S. (2001). Genetic Algorithms. London: Springer-Verlag.

76. Math Works. (2006). "Matlab Documentation" [Online]. [Accessed January 5, 2007]. Available from World Wide Web: http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html.

77. McMasters, A.W. and Mustin, T.M. (1970). "Optimal interdiction of a supply network." *Naval Research Logistics Quarterly*, 17, 261–268.

78. Nataf, A. (1962). "Détermination des Distribution don't les Marges Sont Donées." *Comptes Rendus de l' Academic des Sciences*, 225, 42-43.

79. Nuclear Regulatory Commission (NRC). (2005). "Design Basis Threat." *Federal Register*, 70 (214).

80. Osborne, M.J. (2004). An Introduction to Game Theory. New York, NY: Oxford University Press.

81. Pan, F., Charlton, W. S., and Morton, D.P. (2003). "A stochastic program for interdicting smuggled nuclear material." In D.L. Woodruff, editor, Network Interdiction and Stochastic Integer Programming, 1–20, Kluwer Academic Publishers.

82. Pandey, M.D. and Sarkar, A. (2002). "Comparison of a simple approximation for multinormal integration with an importance sampling-based simulation method." *Probabilistic Engineering Mechanics,* 17 (2), 215-218.

83. Pareto, V. (1971). Manual of political economy. Translated by Ann S. Schwier, Edited by Ann S. Schwier and Alfred N. Page. New York: A.M. Kelley.

84. Peterson, J. L. (1977). "Petri Nets." *ACM Computing Surveys*, 9(3), 223–252.

85. Petri, C. A. (1962). "*Kommunikation mit Automaten*." Ph.D. Thesis. University of Bonn.

86. Petri, C.A. (1966). Communicating With Automata. New York: Griffiths Air Force Base Technical Report RADC-TR-65—377.

87. Phillips, C.A. (1992). “The Network Destruction Problem SAND-92-0186C.” Sandia National Laboratories, Albuquerque, NM.

88. Powell, M.J.D. (1978). "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," *Numerical Analysis*, In G.A. Watson (Eds.), Lecture Notes in Mathematics, 630. Springer-Verlag.

89. Powell, M.J.D. (1978). "The Convergence of Variable Metric Methods For Nonlinearly Constrained Optimization Calculations," in Mangasarian, O.L., Meyer, R.R. and Robinson, S.M. (Eds.), Nonlinear Programming 3. Academic Press.

90. Rackwitz, R. and Fiessler, B. (1978). “Structural reliability under combined random load sequences.” *Computers and Structures*, 9, 489–494.

91. Rausand, M. and Høyland, A. (2004). System Reliability Theory: Models, Statistical Methods, and Applications. Hoboken, NJ: John C. Wiley & Sons.

92. Rosenblatt, M. (1952). “Remarks on a multivariate transformation.” *Ann. Math. Stat.*, 23(3), 470–472.

93. Royset J.O., Der Kiureghian A., Polak E. (2001). “Reliability-based Optimal Structural Design by the Decoupled Approach.” *Reliability and Structural Safety*, 73, 213-221.

94. Sanchez, S. and Wood, R.K. (2002). “The BEST Algorithm To Solve Stochastic Integer Programs.” *INFORMS Annual Meeting*, San Jose, November 17-20.

95. Schneier, B. (1999). “Attack Trees: Modeling Security Threats.” *Dr. Dobb’s Journal*, 24 (12), 21-27.

96. Solberg, I. (2000). “Fminconset” [Online]. [Accessed January 9, 2007]. Available from World Wide Web: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=96&objectType=file.

97. Tu, J., Choi, K.K and Young H.P. (1999). “A New Study on Reliability-Based Design Optimization.” *ASME Journal of Mechanical Engineering*, 121:557-564.

98. Tvedt, L. (1990). “Distribution of Quadratic Forms in Normal Space – Application to Structural Reliability.” *Journal of Engineering Mechanics*, 116(6), 1183-1197.

99. United States Joint Forces Command. (2006). “Joint Conflict and Tactical Simulation (JCATS)” [Online]. [Accessed December 19, 2006]. Available from World Wide Web: http://www.jwfc.jfcom.mil/about/fact_jcats.htm.

100. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F. (1981). "Fault Tree Handbook." Washington, D.C.: U.S. Nuclear Regulatory Commission.

101. Warshall, S. (1962). "A theorem on Boolean matrices." *Journal of the ACM*, 9, 11-12.

102. Washburn, A. and Wood, R.K. (1995). "Two-Person Zero-Sum Games for Network Interdiction." *Operations Research*, 43(2), 243-251.

103. Winblad, A., Snell M., Jordan, S.E., Key, B., Bingham, B. (1989). "ASSESS (Analytic System and Software for Evaluating Safeguards and Security) outsider analysis module SAND-89-1602C." Sandia National Laboratories, Albuquerque, NM.

104. Wood, R.K. (1993). "Deterministic Network Interdiction." *Mathematical and Computer Modeling*, 17(2), 1-18.

105. Wooldridge, M. (2002). An Introduction to Multiagent Systems, Wiley, Chichester, England.

106. Wu, Y.T., Shin Y., Sues, R., and Cesare M. (2001). "Safety-Factor based Approach for Probabilistic-based Design Optimization." *42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference and Exhibit*, Seattle, Washington.

107. Zou T. and Mahadevan, S. (2006). "A direct decoupling approach for efficient reliability-based design optimization," *Structural and Multidisciplinary Optimization*, 31(3), 190-200.