9-2021

Enterprise Resource Allocation for Intruder Detection and Interception

Adam B. Haywood

Enterprise Resource Allocation for Intruder Detection and Interception

DISSERTATION

Adam B. Haywood
AFIT-ENS-DS-21-S-043

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-DS-21-S-043

ENTERPRISE RESOURCE ALLOCATION FOR INTRUDER DETECTION AND INTERCEPTION

DISSERTATION

Presented to the Faculty
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Operations Research, Ph.D.

Adam B. Haywood, BS, M.Ed., MS

July 21, 2021

AFIT-ENS-DS-21-S-043

ENTERPRISE RESOURCE ALLOCATION FOR INTRUDER DETECTION AND INTERCEPTION

DISSERTATION

Adam B. Haywood, BS, M.Ed., MS

Committee Membership:

Dr. Brian J. Lunday (ENS)
Chair

Dr. Matthew J. Robbins (ENS)
Member

Dr. Meir N. Pachter (ENC)
Member

Adedeji B. Badiru, PhD
Dean, Graduate School of Engineering and Management

AFIT-ENS-DS-21-S-043

Abstract

This dissertation considers the problem of an intruder attempting to traverse a defender's territory in which the defender locates and employs disparate sets of resources to lower the probability of a successful intrusion. This research is conducted in the form of three related research components: the first component examines the problem in which the defender subdivides their territory into spatial stages and knows the plan of intrusion. The second component studies a similar problem but is unaware of the intrusion plan, introduces more defensive assets capable of lowering the probability of a successful intrusion, and examines alternative solution methods for instances of the problem. The third component further studies the underlying problem by using a game-theoretic framework in which the attacker observes defender location decisions prior to formulating an appropriate intrusion plan.

Security systems must effectively detect and intercept would-be intruders with an efficient use of limited assets. For the organization of security operations, these operations are often decomposed into spatially distinct stages to organize efforts and facilitate localized management of assets. Given two respective sets of detection resources and interdiction resources, each having different types of resources with heterogeneous capabilities, this research addresses the problem of locating and allocating them over a sequence of spatially-defined stages to effectively detect and intercept an intruder. We set forth a mixed-integer nonlinear mathematical programming model – and seven alternative variants – to address the underlying problem using a leading commercial solver for global optimization. Empirical testing evaluates and compares the effect of alternative model variants on the efficacy and efficiency of the solver to identify global optimal solutions over multiple synthetic instances for a set of scenarios

corresponding to specific problem feature settings. Subsequently, a designed experiment examines the impact of selected problem features on the ability of the leading commercial solver to address increasingly-sized instances of the underlying problem, portending its utility for larger applications. The testing results reveal that the number of types of detection and interdiction resources significantly affect the relative optimality gap achieved, and the number of defender stages is a significant predictor for the required computational effort required when solving a scenario instance. Ultimately, the superlative model variant is identified via two phases of empirical testing and performs well with regard to both solution quality (measured by relative optimality gap achieved) and required computational effort over various sizes of scenarios, identifying solutions within 0.005% of the global optimum for 77.2% of the 900 instances tested, and while only terminating due to the imposed time limit of 900 seconds for 56.8% of the same instances. The research concludes with a description of the extensions to which these results will be applied.

Effectively detecting and interdicting intruders within a defender's territory is a common security problem. Often, the defender's territory is decomposed into spatially distinct stages for organizational convenience. Given an intruder attempting to traverse a spatially-decomposed region via multiple possible paths, this research aims to effectively and cost-efficiently identify a defensive strategy that locates sets of detection resources and interdiction resources, each of which has different types of resources that vary by cost and capability. We formulate and validate a mixed-integer nonlinear programming model to solve the underlying problem first using a leading commercial solver (BARON) and then via two genetic algorithms (RWGA and NSGA-II). Computational testing first identifies instance size limitations for identifying a global optimal solution via BARON, motivating the use of metaheuristics. Subsequent testing demonstrates the superior performance of RWGA and NSGA-II

on 10 randomly generated instances for each of 20 various instance sizes. For each 20 of these instance sizes, both RWGA and NSGA-II produce higher-quality and more non-dominated solutions than BARON while using much less computational effort. Subsequent testing of only RWGA and NSGA-II over a designed set of test instances identifies NSGA-II as the recommended technique to solve larger-sized instances of the underlying problem.

A relevant, applied problem in the location analysis literature is the effective location and allocation of resources to detect and interdict intruders traversing a defended region. For selected applications, a defender's resources are designed to detect and/or interdict intruders on specific parts (or *stages*) of the respective paths. Within this context, this research is motivated by the problem of effectively defending a set of population centers against attack by a limited number of intercontinental ballistic missiles (i.e., intruders) via the location of ballistic missile defense resources to detect and interdict them over a range of launch-to-target missile paths and their respective, spatio-temporally defined stages of flight. Assumed is an adversary capability to observe the defensive asset locations and respond with an ICBM targeting strategy that maximizes the expected damage of an attack. The research presents a bilevel programming model for the corresponding Stackelberg game and, via transformations and reformulations, identifies a single-objective mixed-integer nonlinear program that can be addressed with any of several commercially available solvers. Upon proving the convexity of the resulting formulation to assure reported solutions are globally optimal, comparative testing identifies the commercial solver scip as preferred for solving instances of the underlying problem. Empirical testing via a designed experiment examines which scenario features of the underlying problem are most significant for predicting the required computational effort to solve problem instances, yielding insight into the practical nature of this research to address instances of increasing

size.

In aggregate, this dissertation examines a sequence of models of increasing complexity and fidelity to address the underlying problem of locating defensive assets within an enterprise designed to detect and interdict intruders. Selected assumptions vary across the sequence of models, differing in the manner of addressing intruder detection, the number of intruders, and the rational behavior of an adversary. For each such model, the research proposes and empirically examines an appropriate, accompanying solution methodologies, assessing their efficacy and efficiency for realistic, synthetically-generated instances. Although the research culminates with the proposition of a game theoretic model, arguably the most compelling approach to the problem, aspects of each phase of the dissertation research offer new contributions to the corpus of modeling and solution techniques to benefit this application and other asset location problems.

To J - for always being there.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor, Dr. Brian J. Lunday, for his incredible support and guidance in writing each contribution to this dissertation. Our weekly meetings were a constant source of great insight and brainstorming. I would also like to thank my committee members, Dr. Matthew J. Robbins and Dr. Meir N. Pachter, for lending their expertise and feedback to the dissertation. I am lucky to have the opportunity to work with you all.

I owe my appreciation to the DOD SMART Scholarship for granting me the award that allowed me to pursue this degree and for facilitating my future employment within the United States Air Force. Additionally, thank you to HAF/A9 for being my sponsor upon my receipt of the scholarship.

Lastly, I would like to thank my family and friends for supporting me throughout the program. Specifically, thank you to my family in Cleveland and Columbus for always being happy to see me when I come to visit, my cat for being a great companion to bounce ideas off of, and the fantasy football league for giving me a welcome reprieve from academic stress. Finally, thank you J, for being an amazing partner and always believing in me... sometimes even more than I believed in myself.

Adam B. Haywood

Table of Contents

Page

List of Figures

List of Tables

Table Page

ENTERPRISE RESOURCE ALLOCATION FOR INTRUDER DETECTION AND INTERCEPTION

I. Introduction

1.1 Motivation

The advent of the ballistic missile in World War II came with the need for ballistic missile defense (BMD) (Missile Defense Agency, 2013). Resulting from the evolution of missile technology over the ensuing decades, contemporary versions of these weapons can strike a precise location on a different side of the planet, and they can carry nuclear warheads as well (i.e., intercontinental ballistic missiles or ICBMs). Currently, the United States BMD enterprise consists of various, strategically-placed sensors to detect, identify, and track missile threats. The sensors' role is a fundamental component of successful BMD; their destruction would be a critical loss. Working in concert with the sensors are interceptor launchers deployed to destroy incoming ballistic missiles; each launcher and its interceptors have certain associated costs and likelihoods of successful intercept, given positive identification and tracking of an inbound missile.

If the United States (US) is to become and remain well-defended against various missile threats in the future, it is important to study the BMD enterprise as a whole and identify strategies to optimize the enterprise with respect to risk and cost, all while adhering to the priorities of the Department of Defense (DoD). To wit, Joint Publication (JP) 3-01, *Countering Air and Missile Threats*, provides the doctrinal guidance for defense against air and missile threats targeting the United States and

its allies, establishing the US BMD priorities with respect to protecting assets such as high-value Geopolitical Assets/Areas and high-value air assets (HVAA) (United States Joint Chiefs of Staff, 2017). Hereafter, we use the term *high value assets* or HVAs to refer to the assets being protected by the BMD enterprise, distinguishing them from the BMD assets (e.g., sensors, interceptors) within the enterprise.

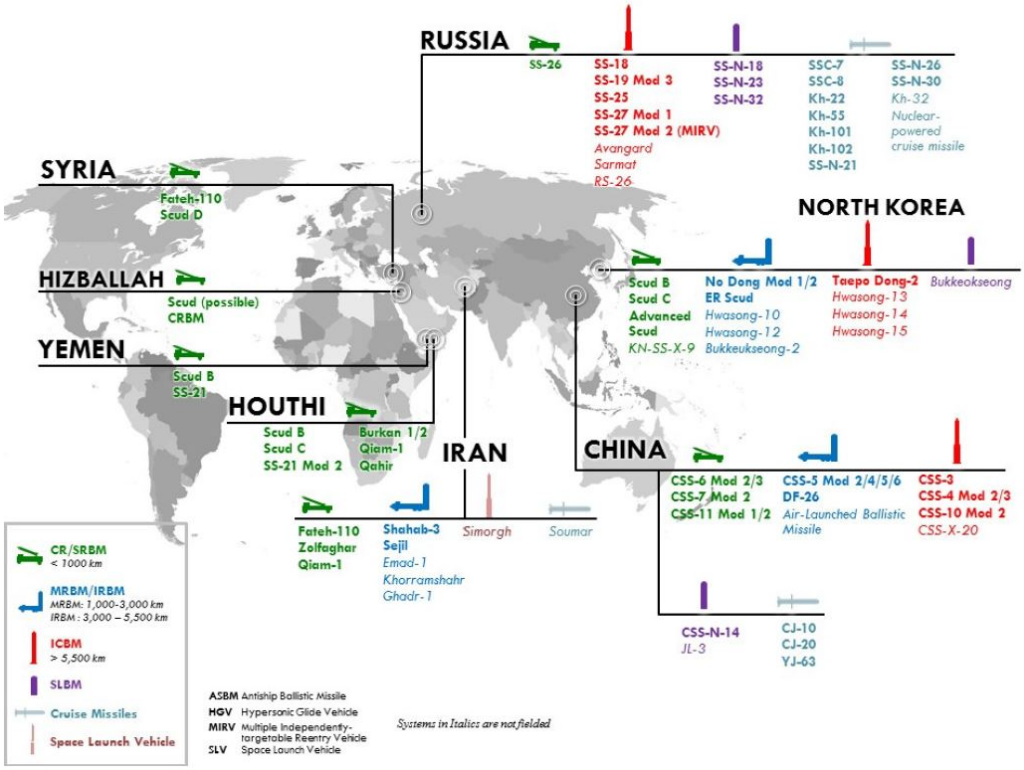

Figure 1. Current and Future Potential Adversary Offensive Missile Capabilities - page 7, 2019 Missile Defense Review (United States Department of Defense, 2019)

The current and future landscape of missile defense must protect against attacks by each of four types of missiles: ballistic missiles, cruise missiles, hypersonic cruise missiles, and hypersonic glide vehicles (Speier et al., 2017). Ballistic missiles are ubiquitous due to arms proliferation, and the defense against these missiles is at the core of the current missile defense enterprise (United States Department of Defense, 2019). As can be seen in Figure 1, cruise missiles are only owned by China, Russia, and Iran; they are much fewer in number than ballistic missiles for each nation (United

States Department of Defense, 2019), yet they pose a threat that must be effectively countered. Lastly, both types of hypersonic weapons are currently a focus of research and development by geopolitical adversaries (e.g., China, Russia), and for which the US has no current defense (Speier et al., 2017).

Although each of the four types of missile threats have flight profiles that can be decomposed for analysis, ballistic missiles are one of the simpler weapons to describe in terms of flight phases. The flight path of a BM is typically characterized via three phases of flight: boost, midcourse, and terminal (National Research Council, 2008). The *boost phase* consists of the time in which the missile is being powered by a rocket from its launch, e.g., for an intercontinental ballistic missile (ICBM) having a range of over 5500 kilometers, into the upper atmosphere of Earth (National Research Council, 2008). Once there, the missile separates from the booster, and the payload adopts a ballistic trajectory towards its target, based upon the Earth's gravitational pull. Within the ballistic portion of the missile's trajectory, the *midcourse phase* describes the time between payload separation and when the missile re-enters the Earth's atmosphere (National Research Council, 2008). The *terminal phase* characterizes the remainder of the BM trajectory (National Research Council, 2008). The exact distinction between the midcourse and terminal phases is not rigid; it depends on the range of the missile and the specific payload (National Research Council, 2008). Unlike ballistic missiles, cruise missiles are guided and powered for the entire flight to the target (United States Department of Defense, 2019). Hypersonic cruise missiles follow a trajectory similar to cruise missiles, albeit at Mach 5.0 or faster, thereby reducing the time during which a defender can detect and intercept them (Speier et al., 2017). Finally, hypersonic glide vehicles (HGVs) are boosted into the upper atmosphere (i.e., at 50 kilometers or higher but lower than a BM's peak trajectory) and return to the target at hypersonic speed with maneuverability during

the terminal phase of flight (Speier et al., 2017).

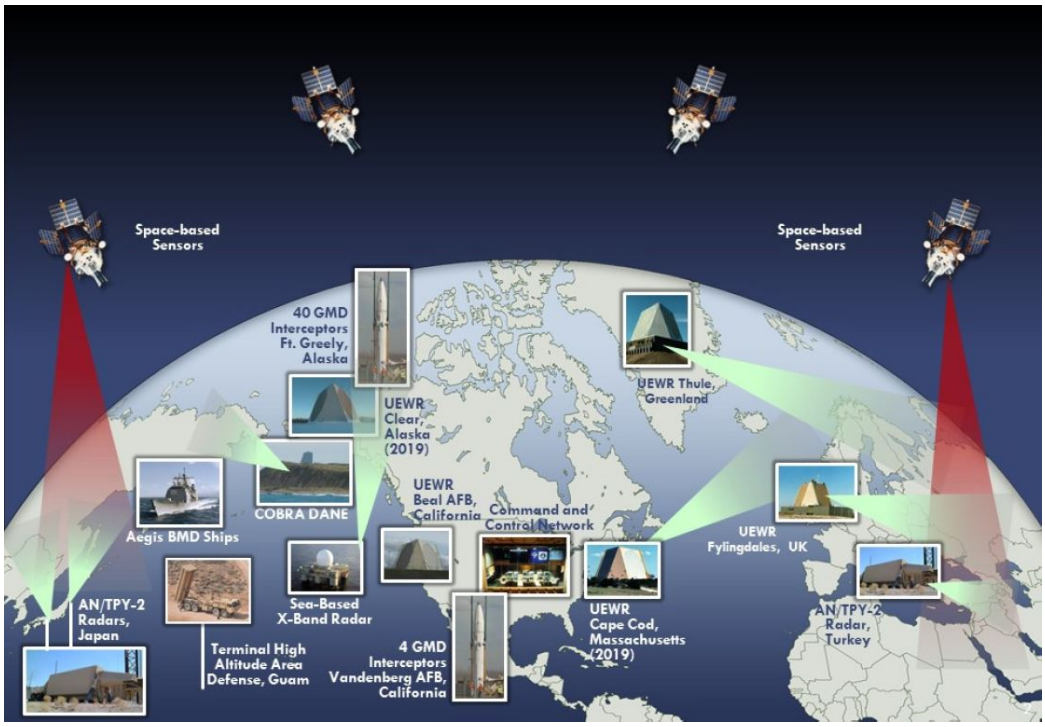

Figure 2. Current Homeland Ballistic Missile Defense Architecture - page 42, 2019 Missile Defense Review (United States Department of Defense, 2019)

Figure 2 provides a visual representation of the assets currently resourced and operational for US Homeland Ballistic Missile Defense. There are four distinct types of sensors that comprise the sensing aspect of the BMD System: ground-based and affixed, sea-based and mobile, space-based, and mobile ground-based. Ground-based and affixed in location are the Army/Navy Transportable Radar Surveillance (AN/TPY-2) in Turkey and Japan; Cobra Dane radar at Shemya, Alaska; and Upgraded Early Warning Radar (UEWR) systems in California, the United Kingdom, and Greenland. The AN/TPY-2 is the largest air-transportable X-band radar in the world, and it can discriminate objects in the midcourse phase of flight (Missile Defense Agency, 2018*b*). The Cobra Dane radar also provides midcourse coverage for the BMD system and is capable of detecting objects out to 2000 miles (Missile Defense Agency, 2016*a*). The UEWR provides midcourse coverage as well, but it is able to

detect objects as far as 3000 miles away (Missile Defense Agency, 2016*b*). In addition to the three current UEWR systems, two existing Early Warning Radar systems at Clear Air Force Station, Alaska, and Cape Cod, Massachusetts, are expected to be upgraded and operational in the near future (United States Department of Defense, 2019). Sea-based and mobile sensors include the Sea-based X-band (SBX) radar as well as the Aegis radar system. The SBX radar is an X-band radar mounted on a self-propelled, semi-submersible platform capable of patrolling the Pacific Ocean when deployed (Missile Defense Agency, 2018*a*). The Aegis weapon system, which is employed on 22 US Navy cruisers and 62 destroyers, uses a AN/SPY radar (US Navy, 2019). Some of these ships patrol the Pacific Ocean and are capable of detecting (and intercepting) an intruder missile in the midcourse phase (US Navy, 2019). Space-based sensors include two satellites in orbit that provide accurate tracks of midcourse re-entry vehicles to BMD system interceptors (Missile Defense Agency, 2017*b*). This system was successfully demonstrated in 2013 when a test missile was launched by the US from Hawaii towards a large empty area of the Pacific Ocean. A space-based sensor relayed information to an Aegis ship, which launched an SM-3 missile and successfully intercepted the "intruder" (United States Department of Defense, 2019). Finally, selected, mobile ground-based assets have sensors to support terminal intercept of inbound BMs. For example, Terminal High Altitude Area Defense (THAAD) has a built-in AN/TPY-2 radar that provides updated tracking data to its interceptors (Missile Defense Agency, 2018*b*).

The cost and feasibility of successfully detecting and intercepting a missile is not uniform across the sequential stages of its flight (National Research Council, 2008). Because intercepting missiles in the boost phase has been deemed to be too impractical to date, intercepting them in the midcourse phase has been the next logical point of focus. Interception in the midcourse phase has its own set of challenges,

such as the *midcourse discrimination problem*. When a ballistic missile enters the midcourse phase, the payload is no longer being propelled by its booster and employs various decoys to make it difficult to intercept (National Research Council, 2008). To destroy the missile, the interceptor must be able to correctly distinguish the missile from the decoys. The sensors employed by current missile defense systems are tasked with identifying the actual missile threat among the decoy threats. Finally, interception during the terminal phase is accompanied by greater certainty with respect to detecting, identifying, and tracking a threat, but it has relative disadvantages. Given the geographic expanse of the Continental United States, terminal defense of all potential HVAs requires an extensive investment in many systems. More expensive systems designed for boost or midcourse detection (and intercept) may be less costly, in aggregate, to achieve the same outcomes.

The US BMD system consists of several assets capable of intercepting a missile in different phases of its flight, and which are designed to provide a layered defense of assets (United States Department of Defense, 2019; Thompson, 2020). The US possesses the Ground-Based Midcourse Defense (GMD) system, which consists of Ground-Based Interceptors (GBI) (United States Department of Defense, 2019). These interceptors are staged at Fort Greely, Alaska and Vandenberg Air Force Base in California (United States Department of Defense, 2019). The Terminal High Altitude Area Defense (THAAD) is a mobile system capable of intercepting missiles in their terminal phase of flight (United States Department of Defense, 2019). The US currently possesses seven THAAD batteries placed around the world, including Guam and the Republic of Korea (United States Department of Defense, 2019). Each THAAD battery is comprised of a truck-mounted launcher which can be loaded with as many as eight interceptors (Missile Defense Agency, 2018*b*). The Phased Array Tracking Intercept of Target (PATRIOT) missile defense system is another BMD as-

set capable of intercepting missiles in their terminal phase of flight (United States Department of Defense, 2019). There are currently eight battalions with thirty-three PATRIOT batteries stationed in the US and seven battalions with twenty-seven PATRIOT batteries stationed overseas (United States Department of Defense, 2019). Each of these batteries consists of six launchers (with some possible exceptions), and each launcher is capable of firing either 16 PAC-3 missiles or 4 PAC-2 missiles simultaneously (Gourley, 2011).

A specific missile threat to the US and its allies that motivates this study arises from the Democratic People's Republic of Korea (DPRK). In recent years, there has been an escalation in the DPRK's missile capability and testing that poses increasing danger. In 1998, the DPRK launched their first ballistic missile – the Taepodong-1 (Arms Control Association, 2019). Shortly thereafter, the DPRK agreed to stop testing and launching ballistic missiles. For the next ten years, there were only a few isolated incidents that could be classified as a missile test. However, in the most recent ten years, there has been a significant increase in testing, and in the last year alone, there have been at least eight reported tests (Arms Control Association, 2019). These various events comprise testing of both short-range ballistic missiles and intermediate-range ballistic missiles, most of which terminated their flight in the Sea of Japan without causing any physical harm (Arms Control Association, 2019).

In addition to the increase in *quantity* of its ballistic missile tests, the DPRK has achieved increasing *success* of its testing. Between 1998 and 2014, there were approximately six successful DPRK missile launches (Arms Control Association, 2019). Since then, there have been approximately 25 successful DPRK tests related to ballistic missiles. Most of those tests simply launch a missile, but some tests relate to the engines that propel the missiles into the upper atmosphere.

The US BMD Enterprise includes assets deliberately arrayed to counter the threat

of ballistic missile attack against CONUS by the DPRK. As discussed previously, the assets that the US has to intercept a missile fired from the DPRK include the GBIs in Fort Greely, Alaska, which are staged directly under the expected path of a DPRK ballistic missile (United States Department of Defense, 2019). In addition, the Aegis cruisers deployed by the US Navy are capable of defending the areas near Japan that the DPRK missiles historically have flown over during their tests.

Recently, there has been a lack of consensus regarding not only the investments that should be made to the future US missile defense enterprise, but also regarding the question of where to locate the current assets and allocate the existing resources to be most effective. For example, the US Navy's Chief of Naval Operations (CNO) recently advocated for the US Department of Defense to reconsider the use of Aegis ships being used to statically patrol a region of the Pacific Ocean to protect against the missile threat from the DPRK, and the CNO proposed DoD simply leverage ground-based systems to protect those assets instead (Larter, David B., 2018). If the US missile defense system is to be most effective, an enterprise-wide examination is necessary, leveraging appropriate modeling to recommend future courses of action for resource investment.

1.2 Problem Statement

Given the motivating problem, this research seeks to address the following problem statement:

> *Efficiently allocate and locate limited detection-and-tracking and interception resources within a Ballistic Missile Defense enterprise to effectively*[1] *defend a set of stationary, ground-based assets against an intercontinental ballistic missile*

[1]This adverb implies the US priority goals and tradeoffs via multiple, (potentially) competing objectives.

attack, while seeking solution robustness to account for increasingly sophisticated adversary strategies.

However, we note the necessarily classified nature of selected capabilities in the U.S. missile defense enterprise. For that reason, this research instead addresses an analogue to the aforementioned problem statement, while still accounting for important characteristics of both ballistic missiles and the BMD enterprise.

Although the aforementioned problem statement indicates defense against a ballistic missile threat, we seek to develop models suitable for countering BM, CM, HGV, or HCM threats, subject to appropriate parameterization. As such, the models within this research should be generalizable to a threat having characteristics common to each of these types of missiles. Each type of missile has a launch point, an HVA target, and a spatiotemporal flight path, the latter of which is commonly decomposed into phases for the application of defensive assets (e.g., boost, midcourse, and terminal stages of flight for a BM).

With respect to countering an intruding missile, the defensive efforts in each stage of flight are comprised of enterprise resources applied to respectively detect, track, and subsequently intercept the missile. Of course, the entire defensive enterprise performance is of interest, not any stage-specific performance.

Moreover, successful intercept of missiles is not the *only* metric for success. If the enterprise can perform in an efficient manner with respect to other outcomes (e.g., absolute enterprise cost, modifications to an existing enterprise, return on investments), the enterprise is more likely to garner the support from military and political leadership needed to acquire, deploy, and operate it. Additionally, such outcomes have minimal acceptable standards of performance; whereas cost should be minimized, there does exist limited capital (i.e., budget) for acquiring new assets for the enterprise.

Thus, from a practical perspective, this research addresses the previously mentioned problem statement *indirectly* by orienting instead on the more generalized problem statement:

> *Efficiently allocate and locate limited detection, tracking, and interception resources within a defensive enterprise to effectively defend a set of stationary, assets against an attack by multiple intruding assets, for which the intrusion paths can be reasonably decomposed into geospatial (and possibly spatiotemporal) stages, while seeking solution robustness to account for increasingly sophisticated adversary strategies.*

1.3 Intended Contributions

This dissertation will make three contributions to the literature, which collectively will address the problem statement in Section 1.2, albeit for the more generalized (i.e., unclassified) framework of intercepting intruders using assets within a detection-and-interception enterprise. To wit, this research will:

1. Develop an enterprise model to locate and allocate limited resources for the effective detection and intercept an agent for which the intrusion plan is known.

2. Develop an enterprise model to locate and allocate resources for the effective and efficient detection and intercept of agents for which the decision maker has limited knowledge about intrusion plans.

3. Within the context of a Stackelberg game, develop an enterprise model to allocate resources for the effective detection and intercept of agents for which the possible intrusion paths are known, but the agents observing the location decisions and subsequently traverse routes corresponding to a (collective) best response.

1.4 Organization of the Dissertation

Chapters II, III, and IV respectively address the three enumerated contributions in Section 1.3 wherein each chapter motivates and describes the problem of interest, reviews the pertinent literature (or literature to be surveyed) that informs modeling and/or solution methodology development, and presents the expected modeling techniques, solution methods, and/or analyses.

II. Enterprise Resource Location-Allocation to Detect and Interdict Intruders

2.1 Introduction

Many contemporary problems require an enterprise model that aims to identify the appropriate use of disparate resources to detect and intercept intruders in a system. One such problem is ballistic missile defense (BMD), wherein a defender must array sets of radars and interceptors to defend cities against an adversary's launch of ballistic missiles. Other problems such as border protection, the interdiction of refugee movement, cybersecurity, and the prevention of infection spread by natural biological immune systems are likewise characterized by similar objectives, resource-outcome relationships, and constraints. Each of these motivating applications entails a defender seeking to protect fixed assets and an intruder attempting to reach, and possibly attack, those assets by traversing a spatial region. As it relates to a BMD application, intruder missiles seek to destroy high value assets (HVAs) in the defender's territory. In the refugee and border protection application, refugees seek safe havens and resources such as shelter and water within the territory of the defender (Mahecic, 2020). In the cybersecurity application, a hacker may attempt to steal sensitive user data from a server within a computer network (Schlesinger and Solomon, 2020). In the immune system application, an infection may attempt to spread to attack vital organs via the bloodstream (O'Connell and Cafasso, 2018).

Each of these applications also has defenders with a set of HVAs to protect, as well as limited resources to aid in that protection. The resources of the defender typically contribute towards either *detection* or *interception* of the intruder(s) although, for selected applications, a subset of resources may serve both purposes. In the border protection application, sensors along the border and the region within it alert a

defender to border crossings and enable interceptors (e.g., border agents) to meet, detain, and process the refugees. In the cybersecurity application, firewalls inhibit access to a network and alert users of attempts to steal data, as well as enable efficient employment of (virtual and physical) countermeasures to interdict the intrusion attempt and prevent further penetration of the network. In the immune system, white blood cells patrol the body and, if an infection is detected by receptors on the surface of the white blood cell, more are sent to interdict the infection and multiply rapidly to fight it. (For this application, white blood cells both detect and interdict intruding infection agents.) In the BMD application, a defender's radar assets detect and track intruder missiles, and defender interceptors engage and destroy the missiles.

These applications also exhibit a defender allocating detection and interception resources to different spatial stages of a would-be intruder's attack. In the BMD application, inbound ballistic missiles have three stages of flight (i.e., boost, midcourse, and terminal (National Research Council, 2008)), and the defender attempts detect and intercept the intruder in each stage. In the border application, similar to the BMD application, there are multiple layers of detection and interception in place. This framework allows border officials more opportunities to detect refugee movement. In the cybersecurity application, a firewall uses multiple filters to attempt to detect malicious packets of information and then discards them if they are deemed malicious. The infection application can be partitioned into stages (e.g., introduction, bloodstream, organs), although the body's immune system does not necessarily consider them separately.

In each of these applications, it is also apparent that an enterprise approach is necessary. Examining the costs of, and resource allocation to, the various BMD assets should be conducted on an enterprise level rather than an asset-by-asset basis. Examining only the sensors in the BMD application allows for tracking of an intruder

missile, but if the interceptors are not located optimally, the missile can still damage an HVA, unimpeded. In the border security application, if sensors are placed optimally and the border crossings are detected but the refugees are not intercepted, the enterprise has failed to achieve its intended outcomes. If users are only alerted to a network breach after data is stolen by a hacker and there are no firewalls in place, the cybersecurity enterprise is likewise unsuccessful. If the human body fails to rapidly multiply the white blood cells surrounding an infection, the infection will continue to spread and attack more areas of the body. A holistic approach allows for a more cost-effective allocation of resources within the enterprise while addressing the system-wide outcomes, vis-á-vis a myopic approach that yields suboptimal costs and performance.

In each of these different applications, optimizing one area of a defense enterprise is not sufficient. There is a natural trade-space to be examined between the cost and performance of an enterprise. When considering the relative priorities imposed on the different objectives, a more detailed tradeoff analysis is appropriate. Given the motivating problems above, this research seeks to address the following problem statement:

> *Given two respective sets of detection resources and interdiction resources, each having different types of resources with heterogeneous capabilities, locate and allocate them over a sequence of spatially-defined stages and respective candidate locations within each stage to effectively detect and intercept an intruder for which the intrusion plan is known.*

2.1.1 Literature Review

There are several threads of research pertinent to the aforementioned problem statement, and a review of the published, technical literature relating to the different

threads is necessary. The major areas of research related to this study are resource location and allocation models, enterprise resource models, and network interdiction models. The literature on resource location and allocation is rich and extensive, encompassing various types of problems over a large span of time. This research thread can be traced back to Hakimi's (1964) early study of location problems, and later followed by Matlin (1970), who studied the Missile Allocation Problem (MAP). MAP adopts the offensive framework of allocating missiles to targets in a manner that maximizes the expected damage inflicted. Beyond the more abstract problem of allocating resources is the examination of resource location, as well as resource location-and-allocation. Considering binary coverage assumptions (i.e., a demand either is covered or not by a located facility), there are two major classes of models in this thread. Within the first major class, set covering location problems (e.g., Church and ReVelle (1976)) and maximal covering location problems (e.g., Church and ReVelle (1974); Berman and Krass (2002)) identify the optimal location of facilities having fixed covering distances to serve demands and seek to minimize the number (or cost) of facilities used as well as maximize the demands covered, alternatively as objectives or constraints. Within the second major class, p-median and p-center techniques determine location-and-allocation decisions (Hakimi, 1964, 1965), wherein every demand is covered by (i.e., assigned to) a facility, but a specified, limiting covering range does not exist for facilities. Additional examinations consider partial coverage (Karasakal and Karasakal, 2004) and probabilistic coverage (Daskin, 1983). Beyond the scope of this review are several extensive surveys of the related literature. An interested reader is referred to works by Drezner and Hamacher (2001), Daskin (2011), Laporte et al. (2015), and Church and Murray (2018).

The literature that specifically applies resource location and allocation methodologies to locating detection and interdiction resources (hereafter referred to equiva-

lently as a *sensor and interdictor location problem*) is also quite extensive, although the applications are more nuanced. Related to BMD, there is work within a game theoretic context that studies optimally placing missile batteries (e.g., Han et al. (2016), Boardman et al. (2017)). These works use the framework of a two-person, three-stage, extensive form, zero-sum game for which there is assumed to be complete and perfect information between players to model the BMD engagement. In a border protection application, Musman et al. (1997) studied the issue of detecting elusive targets along a border with using limited sensor assets, and Lessin et al. (2019) examined the problem of relocating sensors to account for incapacitated or degraded sensors. In the cybersecurity domain, allocating sensors to an information system to minimize compromised information is also studied (e.g., Nandi et al. (2016)). Related to biological immune systems, Huang (2000) developed algorithms that mimic the body's immuno-response to disease or infection, leveraging those algorithms to solve other location-allocation problems.

Unlike resource location and allocation research, the published literature is relatively sparse as it pertains to enterprise resource models for sensor and interdictor location problems. Within the literature, there does exist a robust stream of research pertaining to enterprise resource planning (ERP) (e.g., see Shehab et al., 2004), a field of research focusing on the business processes within an organization, as well as the sub-discipline of material requirements planning (MRP) (e.g., see Morecroft, 1983), a production-focused examination of the materials, processes, and resources leveraged to attain a specific product. The frameworks for resource planning in these areas differ too much from the problem herein to inform a modeling approach, so we refer a reader interested in more information on ERP to the works of Umble et al. (2003) and Monk and Wagner (2012). The relative dearth of literature specific to this enterprise resource modeling results from a number of factors. Among these fac-

tors, it is challenging to represent disparate assets within an enterprise with accurate representation of their effects with respect to common performance metrics. One related work that applied this concept to anti-terrorism efforts is a study by Lunday et al. (2010), for which the goal was to model the application of defense resources to combat terrorism efforts and minimize the expected damage caused by a terrorist organization. Another study by Moghaddam and Nof (2014) examined the problem of making location-allocation decisions in collaborative networks of service enterprises. While similar, this differs too greatly from the scope of this work since it focuses mainly on meeting overall demands of completing tasks instead of a multi-stage location problem like the one studied herein. Beyond these studies, finding related works that apply a holistic approach to solving the sensor and interdictor location problem are elusive.

Because the current research problem seeks to detect and interdict intruders, the literature related to network interdiction can yield relevant modeling frameworks and insights. Within a military context, the concept of network interdiction originated in Ancient Roman times when the Persian cavalry cut Greek supply lines and routes to water sources in battle (Wood, 2010). The general problem can be stated easily in the context of a directed graph, in which an enemy attempts to traverse from node s to node t and an interdictor tries to "break" arcs in order to stop the enemy from being able to complete the journey (Wood, 1993). Beyond the scope of this work, there exist several surveys of this field of literature. Interested readers are referred to works by Cormican et al. (1998), Israeli and Wood (2002), and Wood (2010).

Several articles exist that specifically apply network interdiction to a sensor and interdictor location problem. For example, Brown et al. (2005) examined a two-sided approach to theater BMD and used network interdiction principles to set a framework for the problem. In a border security, Morton et al. (2007) formulated

models to interdict drug smugglers with nuclear material in the Former Soviet Union by locating radiation sensors. In the cybersecurity application, Nandi and Medal (2016) proposed four network interdiction models designed to aid in removing links in a computer network to minimize the spread of infections. Even using the human immune system as an application to network interdiction has been attempted in recent years. The author de Grey (2005) proposed interdicting (in this case, deleting) the genes required for telomere elongation from as many cells as possible, which is a large factor in cancerous growths reaching a life-threatening stage.

Because our problem consists of modeling two agents and their interactions, a game theoretic context has merit for consideration. To wit, the aforementioned network interdiction studies are *Stackelberg games* (Shoham and Leyton-Brown, 2008), a form of two-player, extensive form games with perfect information and complete information. Within the network interdiction literature, several works of note examine such games in the absence of either the perfect information assumption (e.g., Zheng and Castañón (2012), Yates (2013)) or the complete information game (e.g., Zhang and Ramirez-Marquez (2013), Borrero et al. (2016)). The work herein describes a framework in which one player is making decisions, and thus is not a game-theoretic framework. However, this literature motivates extensions to this research that inform the modeling structures. Herein, we seek a model for the underlying problem that achieves high quality solutions quickly, so it will portend practical tractability when embedded within a game theoretic framework in a sequel to this research.

This research makes three contributions to the literature. First, it sets forth a baseline mathematical programming model – and seven alternative variants – to address the underlying problem of allocating limited resources for the detection and interdiction of an intruder. Second, it conducts empirical testing to evaluate and compare the effect of alternative model variants on the efficacy and efficiency of a

leading commercial solver to identify optimal solutions. Third, it rigorously examines the impact of selected problem features on the ability of a leading commercial solver to address larger instances of the underlying problem, portending its utility for larger applications.

The remainder of this paper is organized as follows. Section 2.2 presents the modeling notation (e.g., sets, parameters, and decision variables) and the mathematical programming formulation variants. Section 2.3 presents the empirical testing, results, and analysis. Finally, Section 2.4 concludes the work and provides recommendations for future research.

2.2 Models and Solution Methodology

To formulate the mathematical program to address the underlying problem, it is necessary to define the following sets, parameters, and decision variables.

Sets.

- $N = \{1, 2, ..., \mathcal{N}\}$ is the number of distinguishable geo-spatial stages over which the intruder may be detected and interdicted by the defender's enterprise of sensors and interdictors, indexed by n. (If $\mathcal{N} = 1$, the following models remain valid, but the indexing of selected sets, parameters, decision variables, and constraints on n can be set aside.) Relative to the set of stages, two assumptions are made regarding the intruder's path. First, we assume that the intruder's intended path transits every stage. Second, the stages are numbered according to the order in which the intruder will attempt to transit them.

- $D = \{1, 2, ..., \mathcal{D}\}$ is the set of different detection resource types, indexed by d, each of which pertains to different capabilities (e.g., range, effectiveness).

- $J = \{1, 2, ..., \mathcal{J}\}$ is the set of possible locations at which detection resources can

be located, indexed by j.

  - The set J is partitioned by stage, i.e., $J = \bigcup_{n \in N} J_n$ and $\bigcap_{n \in N} J_n = \varnothing$.

- $I = \{1, 2, ..., \mathcal{I}\}$ is the set of different interdiction resource types, indexed by i, each of which has different capabilities (e.g., speed, range, probability of success).

- $K = \{1, 2, ..., \mathcal{K}\}$ is the set of possible locations at which interdiction resources can be located, indexed by k. Similar to set J, the set K is likewise partitioned over N.

Parameters.

- $u_d^{\mathbb{D}}$: The maximum number of detection resources of type d that can be emplaced.

- $u_i^{\mathbb{I}}$: The maximum number of interdiction resources of type i that can be emplaced.

- $p_{dj}^{\mathbb{D}}$: The probability that an intruder is detected by a detection resource of type d emplaced at location j.

- $p_{ik}^{\mathbb{I}}$: The conditional probability that an intruder is interdicted by an interdiction resource of type i emplaced at location k given it has been detected.

Decision Variables.

- x_{dj} : equals 1 if a detection resource of type d is emplaced at location j, and 0 otherwise.

- y_{ik} : equals 1 if an interdiction resource of type i is emplaced at location k, and 0 otherwise.

- α_{ik} : equals 1 if an interdiction resource of type i emplaced at location $k \in K_n$ is used to attempt to interdict the intruder in stage n, and 0 otherwise.
- $\pi_n^{\mathbb{D}}$: The conditional probability that an intruder is detected in stage n given it has successfully traversed previous stages, i.e., $1, \ldots, n-1$.
- $\pi_n^{\mathbb{I}}$: The conditional probability that an intruder is interdicted in stage n given it has successfully traversed previous stages, i.e., $1, \ldots, n-1$.
- $\pi_n^{\mathbb{D}\bigcap\mathbb{I}}$: The conditional probability of an intruder being detected and interdicted in stage n given it has successfully traversed previous stages, i.e., $1, \ldots, n-1$.
- $\pi^{\mathbb{D}\bigcap\mathbb{I}}$: The total probability of an intruder being detected and interdicted.

Leveraging the aforementioned notation, we formulate the **Resource Allocation for Intruder Detection and Interdiction** (**RAIDI**) model as follows.

$$\max \quad \pi^{\mathbb{D}\bigcap\mathbb{I}} \tag{1}$$

$$\text{s.t.} \quad \pi^{\mathbb{D}\bigcap\mathbb{I}} = 1 - \prod_{n\in N}\left(1-\pi_n^{\mathbb{D}\bigcap\mathbb{I}}\right), \tag{2}$$

$$\pi_n^{\mathbb{D}\bigcap\mathbb{I}} = \pi_n^{\mathbb{D}}\pi_n^{\mathbb{I}}, \qquad \forall\, n \in N, \tag{3}$$

$$\pi_n^{\mathbb{D}} = 1 - \prod_{d\in D}\prod_{j\in J_n}\left(1-p_{dj}^{\mathbb{D}}\right)^{x_{dj}}, \qquad \forall\, n \in N, \tag{4}$$

$$\sum_{d\in D} x_{dj} \le 1, \qquad \forall j \in J, \tag{5}$$

$$\pi_n^{\mathbb{I}} = 1 - \sum_{i\in I}\sum_{k\in K_n}\alpha_{ik}\left(1-p_{ik}^{\mathbb{I}}y_{ik}\right), \qquad \forall\, n \in N, \tag{6}$$

$$\sum_{i\in I}\sum_{k\in K_n}\alpha_{ik} = 1, \qquad \forall\, n \in N, \tag{7}$$

$$\alpha_{ik} \in \{0,1\}, \qquad \forall\, i \in I,\ k \in K_n,\ n \in N, \tag{8}$$

$$\sum_{j \in J} x_{dj} \leq u_d^{\mathbb{D}}, \qquad \forall d \in D, \qquad (9)$$

$$\sum_{k \in K} y_{ik} \leq u_i^{\mathbb{I}}, \qquad \forall i \in I, \qquad (10)$$

$$x_{dj} \in \{0, 1\}, \qquad \forall d \in D,\ j \in J \qquad (11)$$

$$y_{ik} \in \{0, 1\}, \qquad \forall i \in I,\ k \in K \qquad (12)$$

The decision maker seeks to maximize the probability of the detection and subsequent interdiction of an intruder via the objective function (1). Constraint (2) computes this probability as a function of the stage-specific probabilities of detection-and-interdiction, which are assumed to be independent. Likewise, Constraint (3) calculates each stage-specific probability as the product of the respective probabilities of detecting and interdicting the intruder in a given stage, each of which are also assumed to be independent. Constraint (4) calculates the overall probability that an intruder is not detected in stage n. For the purpose of computing stage-specific probabilities of interdiction, this model assumes one interdiction resource is to be used in each stage to interdict a possible intruder. Constraint (5) prevents the emplacement of more than one detection resource at any location. Constraint (6) calculates the probability of an intruder not being interdicted in each stage, which is calculated to be the smallest probability that the intruder is not interdicted over every interdiction resource type in stage n. Constraints (7) and (8) set limitations on the α_{ik}-variables such that at most one interdiction resource-location combination may be utilized in each stage. Lastly, Constraints (9) and (10) limit the number of resources that can be emplaced and Constraints (11) and (12) ensure that the decision variables are binary.

One can alternatively impose Constraint (13) in lieu of Constraint (4), provided it can be assumed that at most one detection resource of any type would be emplaced at any location $j \in J$, as enforced by Constraint (5). This limitation is not enforced

on the interdiction assets, so it is possible that more than one interdiction asset can be emplaced at a single location.

$$\pi_n^{\mathbb{D}} = 1 - \prod_{j \in J_n} \left(1 - \sum_{d \in D} p_{dj}^{\mathbb{D}} x_{dj}\right), \qquad \forall\ n \in N \tag{13}$$

Likewise, one can consider a linear set of constraints as an alternative to Constraint (6). Defining a new decision variable $\beta_{ik} = a_{ik} y_{ik}$, Constraint (6) is alternatively represented as Constraint (14) with the β_{ik}-variables restricted to binary values via Constraint (18). In lieu of the defined nonlinear transformation, the effective relationship is enforced linearly via Constraints (15)–(17).

$$\pi_n^{\mathbb{I}} = 1 - \sum_{i \in I} \sum_{k \in K_n} \left(\alpha_{ik} - p_{ik}^{\mathbb{I}} \beta_{ik}\right), \qquad \forall\ n \in N, \tag{14}$$

$$\beta_{ik} \geq \alpha_{ik} + y_{ik} - 1, \qquad \forall\ i \in I,\ k \in K_n,\ n \in N, \tag{15}$$

$$\beta_{ik} \leq \alpha_{ik}, \qquad \forall\ i \in I,\ k \in K_n,\ n \in N, \tag{16}$$

$$\beta_{ik} \leq y_{ik}, \qquad \forall\ i \in I,\ k \in K_n,\ n \in N, \tag{17}$$

$$\beta_{ik} \in \{0, 1\}, \qquad \forall\ i \in I,\ k \in K_n,\ n \in N \tag{18}$$

Given the two alternative constraint substitutions, we have four formulation variants to consider.

2.3 Testing, Results, and Analysis

This section details the design and conduct of empirical testing to evaluate and compare the efficacy of the alternative formulations corresponding to different combinations of methods for computing $\pi_n^{\mathbb{D}}$ and $\pi_n^{\mathbb{I}}$, respectively. Section 2.3.1 describes both the overall test design of specific scenarios and the methods by which individual test instances are generated for each scenario. Section 2.3.2 presents the testing

results over a set a baseline scenarios, along with an examination of the effects of alternative formulations on both the solution quality and the time to identify an optimal solution. Subsequent analysis in Section 2.3.3 identifies via an experimental design the effect of selected problem features on both the efficacy and efficiency of the best performing model.

Each instance of the model variants was solved on a 2.8 GHz PC with 64 GB of RAM and an Intel(R) Xeon X5660 processor, and using GAMS modeling language (Version 30.1.0) to invoke BARON (Version 19.12.7), a commercial solver designed for global optimization of nonconvex math programs. BARON was applied with a time limit of 15 minutes and a relative optimality gap of 0% for each instance. To solve subproblems, BARON invoked IBM ILOG CPLEX (Version 12.10.0) and/or MINOS (Version 5.5), as appropriate. Testing was completed using the NEOS solver (Gropp and Moré, 1997; Czyzyk et al., 1998; Dolan, 2001), and batch runs were resubmitted as necessary to ensure all testing was conducted on a platform having the aforementioned performance specifications, to facilitate equitable comparison of empirical testing results.

Because BARON leverages a branch and bound framework with the imposition of both feasibility and optimality cuts, conventional wisdom indicates that solver convergence is enhanced via the imposition of tight lower and upper bounds on all decision variables to reduce the volume of the hyperrectangle BARON will iteratively decompose (Ryoo and Sahinidis, 1995, 1996; Sahinidis, 1996; Tawarmalani and Sahinidis, 2004, 2005). As such, each of the four model variants is also examined both with and without the imposition of simple bounds of $[0, 1]$ on each of the $\pi_n^{\mathbb{D}}$-, $\pi_n^{\mathbb{I}}$-, $\pi_n^{\mathbb{D} \bigcap \mathbb{I}}$-, and $\pi^{\mathbb{D} \bigcap \mathbb{I}}$-variables. Table 1 depicts the eight RAIDI model variants tested in subsequent sections and, for each model, how its construction differs.

Table 1. RAIDI Model Variants Tested

Model Variant	$\pi_n^{\mathbb{D}}$ Constraints	$\pi_n^{\mathbb{I}}$ Constraints	$[0,1]$ Variable Bounding
default	(4)	(6)	NO
altdet	(13)	(6)	NO
altint	(4)	(14)-(18)	NO
altdetint	(13)	(14)-(18)	NO
default-b	(4)	(6)	YES
altdet-b	(13)	(6)	YES
altint-b	(4)	(14)-(18)	YES
altdetint-b	(13)	(14)-(18)	YES

2.3.1 Test Instance Generation

The relative performance of model variants may differ due both to problem features and instance features. Within this context, we define problem features for RAIDI formulations as the number of stages, $\mathcal{N}$; the number of types of detection resources, $\mathcal{D}$; the number of possible locations at which detection resources can be located, $\mathcal{J}$; the number of types of interdiction resources, $\mathcal{I}$; and the number of possible locations at which interdiction resources can be located, $\mathcal{K}$. Hereafter, we define a RAIDI *scenario* as a specific set of values for the $(\mathcal{N}, \mathcal{D}, \mathcal{J}, \mathcal{I}, \mathcal{K})$-features.

Scenario generation for testing in Sections 2.3.2 and 2.3.3 is determined via combinations of low, medium, and high levels for each of the problem features. Table 2 presents the respective problem feature levels.

Table 2. Feature Levels Examined for RAIDI Problem Scenarios

Problem	Feature Levels		
Feature	Low	Medium	High
$\mathcal{N}$	3	4	5
$\mathcal{D}$	2	3	4
$\mathcal{J}$	12	16	20
$\mathcal{I}$	2	3	4
$\mathcal{K}$	6	8	10

In contrast, we define an *instance* of a RAIDI problem to be specific to a given scenario. Each instance may vary with respect to the respective partitions of possible detection and interdiction resource locations over stages (i.e., J_n and K_n, $\forall\, n \in N$);

the respective numbers of detection and interdiction resources, by type (i.e., $u_d^{\mathbb{D}},\ \forall\ d \in D$ and $u_i^{\mathbb{I}},\ \forall\ i \in I$); and the respective detection and interdiction probabilities (i.e., $p_{dj}^{\mathbb{D}}$- and $p_{ik}^{\mathbb{I}}$-parameters, indexed accordingly).

In testing throughout Sections 2.3.2 and 2.3.3, this research generates RAIDI instances for a given scenario in the following manner. A batch of 30 instances is iteratively generated for the scenario via fixed pseudo-random number generation seeds in GAMS, and each instance is iteratively solved for each of the model variants in Table 1. For the purpose of instance generation and testing, the partitions of possible locations for respective detection and interdiction resources over stages (i.e., J_n and K_n) are stochastically generated within the GAMS model, with the provision that at least one of each location type is assigned to each stage.

Informing specific parameter values, we assume that higher values of the indices d and i correspond to more capable detection and interdiction resource types, which are likely more expensive and, hence, available in lesser amounts. In general, we expect detection resources to be more prolific than interdiction resources, given for each stage $n \in N$ the RAIDI model considers the effect of all detection resources but only allows for one interdiction resource to be assigned to the intruder. Accordingly, we generate a $u_d^{\mathbb{D}}$- and $u_i^{\mathbb{I}}$-parameters as a function of the scenario parameters. Logical lower and upper bounds on $u_d^{\mathbb{D}}$ are respectively induced by (i) an assumption that at least one detection resource will be emplaced in every stage and (ii) the combination of the number of resource locations and an assumption that detection resources will not be co-located. Equation (19) illustrates for $u_d^{\mathbb{D}},\ \forall\ d \in D$, the generation of a value from a uniform distribution on $[\mathcal{N}, \mathcal{J}]$ and the allocation of an integer-valued proportion of that value by resource type. For example, if $\mathcal{D} = 3$, a ratio of (1/2), (1/3), and (1/6) of the respectively generated values would be used to compute $u_d^{\mathbb{D}}$ for $d = 1, 2, 3$, respectively. Similarly, for interdiction types, Equation (20) illustrates

for $u_i^{\mathbb{I}},\ \forall i \in I$, the generation of a value from a uniform distribution on $[\mathcal{N}, 2\mathcal{N}]$. This upper bound for each type of interdiction resources differs from that of the detection resources because, as previously mentioned, we expect the amount of interdiction resources to be less than that of the detection resources. This upper bound limits each type of interdiction resource to no more than an average of two per stage.

$$u_d^{\mathbb{D}} = \left\lceil \left[\Big((\mathcal{D}+1) - d \Big) \Big/ \sum_{d \in D} d \right] U\left[\mathcal{N}, \mathcal{J}\right] \right\rceil,\ \forall\ d \in D \tag{19}$$

$$u_i^{\mathbb{I}} = \left\lceil \left[\Big((\mathcal{I}+1) - i \Big) \Big/ \sum_{i \in I} i \right] U\left[\mathcal{N}, 2\mathcal{N}\right] \right\rceil,\ \forall\ i \in I \tag{20}$$

Probability parameters also vary by resource type (i.e., d and i, respectively), assuming that types having higher indices are the more capable (and more expensive, hence less available) resources. Accordingly, the instance probabilities are generated in a manner that assigns higher probabilities of detection and interdiction to the higher-valued indices of detection and interdiction type, respectively. For detection assets, a probability range of $\left[p_{\min}^{\mathbb{D}}, p_{\max}^{\mathbb{D}}\right] = [0.2, 0.8]$ is partitioned into $\mathcal{D}$ intervals having equal width, assigning the intervals to resource types with the highest-valued types having the highest probability interval, and so forth. As a modification of the aforementioned procedure to prevent a completely hierarchical partition of resources by type, we modify the partitions of the probability range so the individual intervals overlap one another by 10% of their interval widths. The probabilities of interdiction by type are generated in an identical manner and with an identical probability range of $\left[p_{\min}^{\mathbb{I}}, p_{\max}^{\mathbb{I}}\right] = [0.2, 0.8]$. Equations (21) and (22) detail the instance-specific generation of the by-resource-type probabilities of detection and interdiction, respectively, via uniform distributions, and wherein $\Delta^{\mathbb{D}} = \frac{p_{UB}^{\mathbb{D}} - p_{LB}^{\mathbb{D}}}{\mathcal{D}}$ and $\Delta^{\mathbb{I}} = \frac{p_{UB}^{\mathbb{I}} - p_{LB}^{\mathbb{I}}}{\mathcal{I}}$.

$$p_{dj}^{\mathbb{D}} = U\left[p_{LB}^{\mathbb{D}} + (d-1)\cdot\Delta^{\mathbb{D}}, p_{LB}^{\mathbb{D}} + d\cdot\Delta^{\mathbb{D}}\right],\ \forall\ d \in D,\ j \in J \tag{21}$$

$$p^{\mathbb{I}}_{ik} = U\left[p^{\mathbb{I}}_{LB} + (i-1)\cdot\Delta^{\mathbb{I}}, p^{\mathbb{I}}_{LB} + i\cdot\Delta^{\mathbb{I}}\right],\ \forall\ i \in I,\ k \in K \tag{22}$$

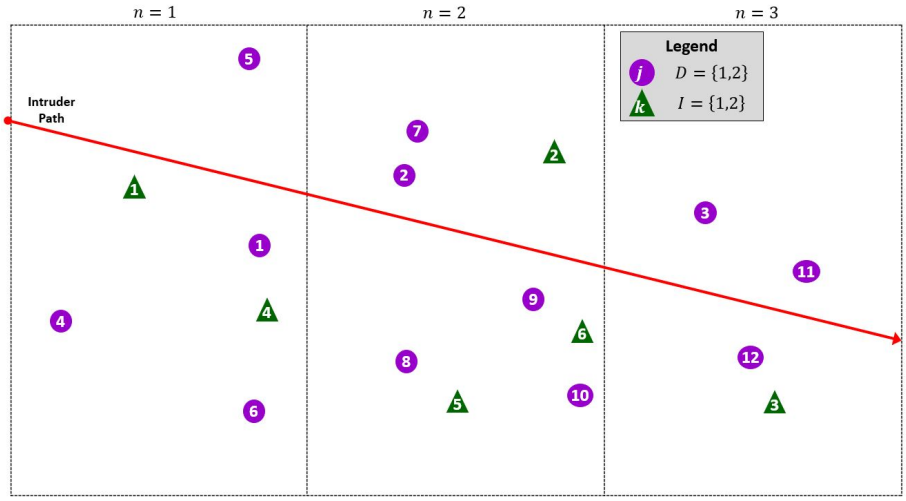

Figure 3. Illustrative Instance of RAIDI

Figure 3 is an illustrative instance of RAIDI in which $\mathcal{N} = 3$, $\mathcal{D} = 2$, $\mathcal{J} = 12$, $\mathcal{I} = 2$, and $\mathcal{K} = 6$.

2.3.2 RAIDI Model Variant Testing and Comparison for Baseline Scenarios

In this section, empirical testing compares the solution quality and computational effort required of the BARON commercial solver to identify a globally optimal solution to each of the RAIDI model variants in Table 1. Each model variant is tested for 30 instances of each baseline scenario presented in Table 3, wherein the scenarios respectively consist of the low, medium, and high levels for features from Table 2.

Table 3. Baseline RAIDI Problem Scenarios

Scenario	$(\mathcal{N}, \mathcal{D}, \mathcal{J}, \mathcal{I}, \mathcal{K})$
Low Feature Level (LFL)	$(3, 2, 12, 2, 6)$
Medium Feature Level (MFL)	$(4, 3, 16, 3, 8)$
High Feature Level (HFL)	$(5, 4, 20, 4, 10)$

The hierarchy of metrics to assess the relative performance of model variants are, in order, the objective function value(s) attained, the relative optimality gap (%) identified upon termination, the required computational effort, and the number of instances on which the solver terminated due to limits on the computational effort. Whereas the solution quality is of foremost importance, we use the relative optimality gap as the second criterion; if the solver does not find a solution it identifies as globally optimal – even though it may be globally optimal – upon termination due to time limitations, it is of notable interest to assess how well the model variant enables a solver assessment of the solution quality. Of tertiary importance is the required computational effort. Finally, we report and consider the number of instances for which the solver terminated due to time limitations (i.e., "terminated early") before identifying a global optimum.

Tables 4–6 report the testing results for the LFL, MFL, and HFL scenarios, respectively. The first column in each table presents the model variant tested. For each variant, the second through fourth columns present the average and standard deviations for the objective function value upon termination, the relative optimality gap (%) identified upon termination, and the computational effort required (seconds) by the commercial solver BARON. The final column reports the number of instances (out of 30) for which the solver terminated due to the 900 second limit on computational effort.

The results shown in Table 4 indicate that, when feature levels are set to low values, there is little differentiation in the performance of BARON among the model variants. The average objective function values obtained for each model variant across the 30 instances were identical, as were the relative optimality gaps. The only difference between the model variants was the average computational effort required to solve the instances, although it is worth noting that BARON did not terminate any

Table 4. RAIDI Model Variant Performances (Means and Standard Deviations) for Selected Performance Metrics over 30 Synthetic Instances of the LFL Scenario

Model	Objective Fn. Value.	Relative Gap (%)	Req'd Comp. Effort (sec)	No. Instances Terminated Early
default	0.955 ± 0.029	0 ± 0	4.413 ± 10.619	0
default-b	0.955 ± 0.029	0 ± 0	2.579 ± 8.998	0
altdet	0.955 ± 0.029	0 ± 0	1.083 ± 3.060	0
altdet-b	0.955 ± 0.029	0 ± 0	0.419 ± 1.062	0
altint	0.955 ± 0.029	0 ± 0	2.360 ± 6.650	0
altint-b	0.955 ± 0.029	0 ± 0	1.150 ± 2.079	0
altdetint	0.955 ± 0.029	0 ± 0	1.635 ± 6.841	0
altdetint-b	0.955 ± 0.029	0 ± 0	0.751 ± 1.961	0

instances early for any model variant.

Table 5. RAIDI Model Variant Performances (Means and Standard Deviations) for Selected Performance Metrics over 30 Synthetic Instances of the MFL Scenario

Model	Objective Fn. Value.	Relative Gap (%)	Req'd Comp. Effort (sec)	No. Instances Terminated Early
default	0.981 ± 0.018	0.016 ± 0.017	750.631 ± 334.090	25
default-b	0.981 ± 0.019	0.017 ± 0.017	800.990 ± 270.612	26
altdet	0.982 ± 0.018	0.012 ± 0.015	653.572 ± 341.859	18
altdet-b	0.982 ± 0.018	0.013 ± 0.018	633.950 ± 347.910	18
altint	0.981 ± 0.018	0.022 ± 0.083	334.852 ± 430.298	11
altint-b	0.981 ± 0.018	0.016 ± 0.018	738.775 ± 328.320	24
altdetint	0.982 ± 0.018	0.008 ± 0.013	570.398 ± 383.965	16
altdetint-b	0.982 ± 0.018	0.014 ± 0.017	751.583 ± 285.066	23

Table 5 describes the results found when the model variants are used to solve 30 instances of the MFL scenario. The average objective function values obtained were nearly the same, as was the case in the LFL scenario. The average relative gaps obtained by the eight model variants were also similar in value, ranging from 0.008% to 0.022%. Similar to the LFL scenario results, the average computational effort required to solve the 30 instances provided the most differentiation between model variants. The *altint* model variant yielded the lowest average computational effort required. However, *altint* also had the worst relative optimality gap results, whereas the *default-b* model variant yielded the highest average computational effort required. As expected, these two model variants respectively had the least (11) and most (26) number of instances for which the solver terminated early due to the limit

on run time.

Table 6. RAIDI Model Variant Performances (Means and Standard Deviations) for Selected Performance Metrics over 30 Synthetic Instances of the HFL Scenario

Model	Objective Fn. Value.	Relative Gap (%)	Req'd Comp. Effort (sec)	No. Instances Terminated Early
default	0.990 ± 0.011	0.010 ± 0.010	900.015 ± 0.095	30
default-b	0.990 ± 0.010	0.010 ± 0.010	870.275 ± 160.966	29
altdet	0.990 ± 0.010	0.286 ± 0.021	900.050 ± 0.259	30
altdet-b	0.991 ± 0.009	0.009 ± 0.009	900.019 ± 0.116	30
altint	0.990 ± 0.010	0.020 ± 0.059	481.369 ± 446.079	15
altint-b	0.991 ± 0.009	0.009 ± 0.009	870.011 ± 161.447	29
altdetint	0.991 ± 0.009	0.271 ± 0.206	899.990 ± 0.044	30
altdetint-b	0.991 ± 0.009	0.049 ± 0.009	899.994 ± 0.049	30

Table 6 shows the HFL scenario results, which indicate that as the feature levels rise, there are more differences for the performance of BARON between the model variants. As was the case in the LFL and MFL scenario results, the objective function values were nearly identical across all model variants. The average relative optimality gap amongst the eight model variants varied more widely. Notably, *altdet*, *altdetint*, and *altboth* had the highest average relative optimality gaps and *default*, *default-b*, *altdet-b*, and *altint-b* had the lowest. Collectively, these results indicate that introducing bounds on decision variables leads to lower relative optimality gaps, as expected. The average computational effort required to solve the instances are all similarly high (and close to or effectively 900 seconds), with the exception of the *altint* model variant, for which the average computational effort required is much lower. This result corresponds with the number of instances for which BARON terminated early for each model variant; for the *altint* model variant, BARON only terminated early on 15 of the 30 instances, whereas it terminated early on at least 29 of the 30 instances for every other model variant. The *altint* model variant finds a solution faster than the other model variants but at the slight expense of solution quality. In contrast, BARON terminated early for the other model variants quite often, but the solutions found are of high quality. Another notable result is that BARON terminated early

on all 30 instances for model variants that include the alternative constraints for detection.

Overall, model variants achieved objective function values that were nearly identical, but both the relative optimality gaps and computational effort required differed across model variants as the scenario feature levels increased in magnitude. The relative optimality gaps, required computational effort, and the number of instances terminated due to the limit on run time increased as the scenario feature levels increased. A common pattern across scenarios is that the *altint* model variant performed better with respect to the computational effort required and worse with respect to relative gap obtained. Another important pattern is that the model variants that included the variable bounding performed well in all scenarios with respect to relative optimality gap. The time limit of 900 seconds was not relevant in the LFL scenario, but it became more relevant in the MFL and HFL scenarios. It is important to note that even though BARON may terminate early on a high number of instances for a particular model variant, it does not mean that it obtained poor objective function values. Overall, the *default, default-b, altdet-b,* and *altint-b* model variants are identified as higher-quality model variants across these baseline scenarios due to having both a high average objective function value and a low average relative optimality gap over the set of three baseline scenarios. Even when these model variants had a high number of instances terminate early (e.g., in the HFL scenario), they still achieved the low relative optimality gaps and high objective function values. These five model variants are examined further in Section 2.3.2.

Insights from Early Model Convergence

The commercial solver BARON terminated prematurely for the *majority* of HFL instances for *each* model variant, and it likewise terminated due to the imposed time

limit for *all* HFL instances for a *majority* of model variants. As such, it is of interest to examine the degree to which the model variants enabled *early* vis-á-vis *terminal* solution quality.

Table 7 describes the average quality, in terms of average reported objective function value, for the first and last solutions found by BARON for the model variants in the HFL scenario. Informing the second column of Table 7, the root node objective function value corresponds to the solution found by BARON at the first node in its branch-and-bound sequence; it provides a lower bound upon which the solver seeks improvement over the 900 second time limit allowed for each instance. In the third column of Table 7, the terminal objective function values are the same ones reported in Table 6; the entries denote the average objective function value that BARON reported upon termination, whether due to reaching the allotted time limit or by identifying a global optimal solution.

Table 7. RAIDI Model Variant Performance (Means and Standard Deviations) for Terminal Objective Function Value and Root Node Objective Function Value over 30 Synthetic Instances of the HFL Scenario

	Root Node	**Terminal**
Model	**Obj. Fn. Val.**	**Obj. Fn. Val.**
default	0.984 ± 0.013	0.990 ± 0.011
default-b	0.985 ± 0.013	0.990 ± 0.010
altdet	0.985 ± 0.012	0.990 ± 0.010
altdet-b	0.986 ± 0.012	0.991 ± 0.009
altint	0.990 ± 0.010	0.990 ± 0.010
altint-b	0.990 ± 0.011	0.991 ± 0.009
altdetint	0.990 ± 0.010	0.991 ± 0.009
altdetint-b	0.991 ± 0.009	0.991 ± 0.009

The results shown in Table 7 indicate the model variants using alternative constraints for interdiction have average objective function values that do not improve much at all as BARON runs longer, whereas each of the other four model variants' performance improves noticeably over time. That is, the model variants *altint, altint-b, altdetint,* and *altdetint-b* provide slightly higher quality solutions early during BARON's solution methodology. Since the model variants *default, default-b, altdet-b,*

and *altint-b* were recommended as higher-quality model variants in Section 2.3.2, and Table 7 reinforces that these models are still of high quality with respect to root node solutions, these are the recommended model variants to be explored hereafter.

2.3.3 RAIDI Scenario Feature Examination

Given the superlative performance of the RAIDI model variants *default, default-b, altdet-b* and *altint-b* in Section 2.3.2, herein we test the effects of different scenario features on the efficacy and efficiency of BARON for solving RAIDI problem instances.

A full factorial experiment on the different feature levels presented in Table 2 to determine which problem features are most influential to the RAIDI problem would require examining $3^5 = 243$ unique scenarios (with multiple instances for each scenario). The model parameters $\mathcal{N}, \mathcal{D}, \mathcal{J}, \mathcal{I}$, and $\mathcal{K}$ are the five factors. Such an endeavor is computationally expensive, if not prohibitive; therefore, a fractional factorial design is preferable to garner useful insights with lesser computational effort. A full factorial design has no aliasing between factor combinations; in contrast, the effects of higher-order interactions between problem features is elusive when examining a fractional factorial design. However, either is preferable to one-factor-at-a-time (OFAT) analyses for the greater insights that can be obtained.

The fractional factorial experiment used herein is a 3^{5-2}_{III} design with 30 trials in each treatment combination as depicted in each of Tables 8-11. This experiment is conducted for each of the model variants *default*, *default-b*, *altdet-b*, and *altint-b*, collectively chosen for their relatively superior performance in Section 2.3.2. The primary response variable of interest is the relative optimality gap (%) achieved within the 15-minute time limit, whereas the second response variable is the required computational effort, subject to the same upper bound. Tables 8-10 report both the average and standard deviation for each of these response variables, for each treat-

ment combination within the experiment, computed over 30 trials. For each of the three mathematical programming formulation variants, two regression models are developed – one for each response variable – to garner greater insights via an exploration individual and combined effects.

Table 8. Treatment Levels and Relative Optimality Gap Metrics (Mean and Standard Deviation) - *default* Model Variant

	Factors					Relative	Req'd Comp.
Run	$\mathcal{N}$	$\mathcal{D}$	$\mathcal{J}$	$\mathcal{I}$	$\mathcal{K}$	**Gap (%)**	**Effort (sec)**
1	3	2	12	2	6	0.000 ± 0.000	0.843 ± 0.921
2	3	2	16	3	6	0.007 ± 0.018	146.223 ± 324.842
3	3	2	20	4	6	0.006 ± 0.017	200.860 ± 364.287
4	3	3	12	3	10	0.003 ± 0.011	98.732 ± 247.415
5	3	3	16	4	10	0.004 ± 0.012	131.411 ± 307.137
6	3	3	20	2	10	0.003 ± 0.008	151.971 ± 336.498
7	3	4	12	4	8	0.005 ± 0.016	122.275 ± 281.447
8	3	4	16	2	8	0.005 ± 0.011	182.823 ± 360.520
9	3	4	20	3	8	0.014 ± 0.011	724.639 ± 360.539
10	4	2	12	3	8	0.002 ± 0.009	266.818 ± 338.228
11	4	2	16	4	8	0.009 ± 0.016	629.090 ± 412.902
12	4	2	20	2	8	0.007 ± 0.008	629.803 ± 414.645
13	4	3	12	4	6	0.011 ± 0.015	686.517 ± 368.972
14	4	3	16	2	6	0.006 ± 0.009	408.144 ± 429.615
15	4	3	20	3	6	0.008 ± 0.010	606.524 ± 417.374
16	4	4	12	2	10	0.007 ± 0.010	702.288 ± 356.076
17	4	4	16	3	10	0.007 ± 0.013	733.427 ± 365.907
18	4	4	20	4	10	0.007 ± 0.007	906.841 ± 13.588
19	5	2	12	4	10	0.008 ± 0.014	825.638 ± 239.568
20	5	2	16	2	10	0.004 ± 0.008	801.153 ± 272.499
21	5	2	20	3	10	0.004 ± 0.006	925.267 ± 21.824
22	5	3	12	2	8	0.004 ± 0.007	768.663 ± 313.157
23	5	3	16	3	8	0.005 ± 0.008	921.277 ± 24.706
24	5	3	20	4	8	0.005 ± 0.009	935.088 ± 20.213
25	5	4	12	3	6	0.009 ± 0.013	885.132 ± 163.099
26	5	4	16	4	6	0.008 ± 0.010	905.427 ± 169.272
27	5	4	20	2	6	0.007 ± 0.007	857.212 ± 228.591

Table 12 presents the feature coefficients corresponding to each of two standard least squares (SLS) regression models (i.e., one for each of the two responses) for the respective data summarized in Table 8. For the relative optimality gap response, only $\mathcal{D}$ and $\mathcal{I}$ were significant factors. This outcome indicates that only the number of types of detection and interdiction resources, respectively, are significant predictors of the relative optimality gap. For the required computational effort response,

Table 9. Treatment Levels and Relative Optimality Gap Metrics (Mean and Standard Deviation) - *default-b* Model Variant

	Factors					Relative	Req'd Comp.
Run	$\mathcal{N}$	$\mathcal{D}$	$\mathcal{J}$	$\mathcal{I}$	$\mathcal{K}$	**Gap (%)**	**Effort (sec)**
1	3	2	12	2	6	0.000 ± 0.000	0.606 ± 0.505
2	3	2	16	3	6	0.007 ± 0.019	121.138 ± 306.765
3	3	2	20	4	6	0.007 ± 0.015	208.996 ± 365.032
4	3	3	12	3	10	0.005 ± 0.012	162.788 ± 340.757
5	3	3	16	4	10	0.006 ± 0.014	212.016 ± 381.700
6	3	3	20	2	10	0.002 ± 0.005	142.214 ± 318.971
7	3	4	12	4	8	0.008 ± 0.015	220.054 ± 381.469
8	3	4	16	2	8	0.002 ± 0.007	147.795 ± 310.475
9	3	4	20	3	8	0.018 ± 0.015	754.672 ± 336.486
10	4	2	12	3	8	0.001 ± 0.003	175.213 ± 306.946
11	4	2	16	4	8	0.007 ± 0.014	480.915 ± 429.437
12	4	2	20	2	8	0.004 ± 0.007	432.141 ± 438.654
13	4	3	12	4	6	0.010 ± 0.014	586.550 ± 407.299
14	4	3	16	2	6	0.008 ± 0.010	550.945 ± 422.964
15	4	3	20	3	6	0.011 ± 0.009	842.138 ± 223.983
16	4	4	12	2	10	0.006 ± 0.011	642.040 ± 371.868
17	4	4	16	3	10	0.007 ± 0.012	822.627 ± 272.632
18	4	4	20	4	10	0.007 ± 0.007	906.627 ± 14.334
19	5	2	12	4	10	0.007 ± 0.014	855.459 ± 195.352
20	5	2	16	2	10	0.004 ± 0.009	812.452 ± 232.288
21	5	2	20	3	10	0.003 ± 0.006	925.904 ± 22.789
22	5	3	12	2	8	0.004 ± 0.008	681.418 ± 382.602
23	5	3	16	3	8	0.004 ± 0.008	890.671 ± 164.932
24	5	3	20	4	8	0.005 ± 0.009	932.129 ± 19.226
25	5	4	12	3	6	0.009 ± 0.013	865.085 ± 194.279
26	5	4	16	4	6	0.008 ± 0.010	904.935 ± 169.255
27	5	4	20	2	6	0.006 ± 0.006	858.293 ± 228.813

Table 10. Treatment Levels and Relative Optimality Gap Metrics (Mean and Standard Deviation) - *altdet-b* Model Variant

	Factors					Relative	Req'd Comp.
Run	$\mathcal{N}$	$\mathcal{D}$	$\mathcal{J}$	$\mathcal{I}$	$\mathcal{K}$	**Gap (%)**	**Effort (sec)**
1	3	2	12	2	6	0.000 ± 0.000	0.610 ± 0.616
2	3	2	16	3	6	0.000 ± 0.000	43.488 ± 123.951
3	3	2	20	4	6	0.004 ± 0.012	207.874 ± 322.127
4	3	3	12	3	10	0.003 ± 0.011	157.088 ± 279.502
5	3	3	16	4	10	0.003 ± 0.010	622.626 ± 352.349
6	3	3	20	2	10	0.002 ± 0.007	671.558 ± 353.811
7	3	4	12	4	8	0.006 ± 0.015	647.802 ± 306.568
8	3	4	16	2	8	0.008 ± 0.015	886.947 ± 95.039
9	3	4	20	3	8	0.013 ± 0.018	819.566 ± 260.331
10	4	2	12	3	8	0.000 ± 0.000	25.318 ± 59.210
11	4	2	16	4	8	0.006 ± 0.015	347.192 ± 404.525
12	4	2	20	2	8	0.001 ± 0.002	245.071 ± 361.482
13	4	3	12	4	6	0.008 ± 0.014	533.362 ± 380.427
14	4	3	16	2	6	0.006 ± 0.009	590.541 ± 374.052
15	4	3	20	3	6	0.011 ± 0.010	827.984 ± 199.184
16	4	4	12	2	10	0.005 ± 0.013	573.218 ± 342.702
17	4	4	16	3	10	0.008 ± 0.012	913.665 ± 17.731
18	4	4	20	4	10	0.007 ± 0.007	906.601 ± 13.839
19	5	2	12	4	10	0.005 ± 0.012	574.409 ± 352.286
20	5	2	16	2	10	0.001 ± 0.001	553.374 ± 380.002
21	5	2	20	3	10	0.003 ± 0.005	808.234 ± 282.980
22	5	3	12	2	8	0.004 ± 0.009	606.436 ± 377.640
23	5	3	16	3	8	0.004 ± 0.008	919.060 ± 24.276
24	5	3	20	4	8	0.005 ± 0.009	933.915 ± 20.225
25	5	4	12	3	6	0.008 ± 0.013	736.235 ± 304.141
26	5	4	16	4	6	0.007 ± 0.009	935.999 ± 21.709
27	5	4	20	2	6	0.005 ± 0.005	918.517 ± 18.751

Table 11. Treatment Levels and Relative Optimality Gap Metrics (Mean and Standard Deviation) - *altint-b* Model Variant

	Factors					Relative	Req'd Comp.
Run	$\mathcal{N}$	$\mathcal{D}$	$\mathcal{J}$	$\mathcal{I}$	$\mathcal{K}$	**Gap (%)**	**Effort (sec)**
1	3	2	12	2	6	0.000 ± 0.000	0.903 ± 1.681
2	3	2	16	3	6	0.000 ± 0.000	68.852 ± 176.649
3	3	2	20	4	6	0.003 ± 0.015	74.916 ± 231.779
4	3	3	12	3	10	0.001 ± 0.002	70.736 ± 227.376
5	3	3	16	4	10	0.003 ± 0.011	156.538 ± 302.386
6	3	3	20	2	10	0.007 ± 0.010	362.492 ± 438.878
7	3	4	12	4	8	0.006 ± 0.011	334.629 ± 430.462
8	3	4	16	2	8	0.007 ± 0.010	471.976 ± 440.406
9	3	4	20	3	8	0.010 ± 0.012	565.953 ± 414.857
10	4	2	12	3	8	0.000 ± 0.002	133.974 ± 233.585
11	4	2	16	4	8	0.005 ± 0.015	574.707 ± 407.565
12	4	2	20	2	8	0.005 ± 0.005	641.559 ± 392.923
13	4	3	12	4	6	0.003 ± 0.004	616.691 ± 395.580
14	4	3	16	2	6	0.009 ± 0.011	764.026 ± 309.313
15	4	3	20	3	6	0.007 ± 0.007	870.058 ± 161.228
16	4	4	12	2	10	0.005 ± 0.010	743.093 ± 325.828
17	4	4	16	3	10	0.004 ± 0.005	842.304 ± 216.073
18	4	4	20	4	10	0.003 ± 0.005	785.997 ± 265.275
19	5	2	12	4	10	0.003 ± 0.011	693.944 ± 348.971
20	5	2	16	2	10	0.003 ± 0.007	834.082 ± 217.443
21	5	2	20	3	10	0.002 ± 0.004	900.021 ± 0.007
22	5	3	12	2	8	0.003 ± 0.004	806.637 ± 245.533
23	5	3	16	3	8	0.002 ± 0.007	873.805 ± 141.187
24	5	3	20	4	8	0.002 ± 0.006	899.970 ± 0.156
25	5	4	12	3	6	0.005 ± 0.007	869.986 ± 161.407
26	5	4	16	4	6	0.004 ± 0.006	900.001 ± 0.012
27	5	4	20	2	6	0.006 ± 0.007	870.054 ± 160.962

Table 12. Standard Least Squares Regression Coefficient Estimates for Relative Optimality Gap and Req. Comp. Effort Responses - *default* Model Variant

	Relative Gap (%)				Req'd Comp. Effort (sec)			
Term	**Estimate**	**Std Error**	t **Ratio**	**Prob**$>\|t\|$	**Estimate**	**Std Error**	t **Ratio**	**Prob**$>\|t\|$
$\mathcal{N}$	0.00035	0.00049	0.70	0.4822	336.94898	13.78550	24.44	<.0001
$\mathcal{D}$	0.00116	0.00049	2.35	0.0190	88.57609	13.78550	6.43	<.0001
$\mathcal{J}$	0.00017	0.00012	1.36	0.1727	21.96248	3.44638	6.37	<.0001
$\mathcal{I}$	0.00111	0.00049	2.26	0.0241	46.68033	13.78550	3.39	0.0007
$\mathcal{K}$	-0.00040	0.00025	-1.61	0.1088	16.10684	6.89275	2.34	0.0197

every factor is significant in the regression model, with a positive regression coefficient. As expected, increasing the scenario feature levels leads to a higher required computational effort. Most notably, $\mathcal{N}$ has the largest coefficient; an increase in the number of stages will most rapidly increase the required computational effort when using the *default* model variant to solve RAIDI scenario instances. In addition to the SLS regression models described by Table 12, another regression model that included two-factor interactions was constructed for each response. Although not reported here for the sake of brevity, the $\mathcal{N} \times \mathcal{D}$ was a significant factor in the relative gap response model, and most of the two-factor interactions were significant in the required computational effort response model, indicating the sensitivity of the latter response to all features.

Table 13. Standard Least Squares Regression Coefficient Estimates for Relative Optimality Gap and Req. Comp. Effort Responses - *default-b* Model Variant

	Relative Gap (%)				Req'd Comp. Effort (sec)			
Term	**Estimate**	**Std Error**	**t Ratio**	**Prob$>\|t\|$**	**Estimate**	**Std Error**	**t Ratio**	**Prob$>\|t\|$**
$\mathcal{N}$	-0.00019	0.00049	-0.39	0.6979	319.78156	13.81994	23.14	<.0001
$\mathcal{D}$	0.00161	0.00049	3.30	0.0010	117.18359	13.81994	8.48	<.0001
$\mathcal{J}$	0.00017	0.00012	1.43	0.1538	25.19310	3.45499	7.29	<.0001
$\mathcal{I}$	0.00162	0.00049	3.34	0.0009	57.76532	13.81994	4.18	<.0001
$\mathcal{K}$	-0.00049	0.00024	-2.02	0.0439	15.09557	6.90997	2.18	0.0292

Table 13 reports the results attained when fitting two SLS models to the *default-b* data that is summarized in Table 9. Similarly to the *default* model variant, the results for the relative optimality gap response show that $\mathcal{D}$ and $\mathcal{I}$ are the significant predictors and that every factor is significant for the required computational effort response. Moreover, for the required computational effort response, $\mathcal{N}$ has by far the highest regression coefficient. Indicated is that an increase in the number of stages correlates to an increase in the required computational effort for the *default-b* model variant to solve a RAIDI scenario instance. In addition to the modeling results in Table 13, additional regression models including two-factor interactions were also constructed for each response. None of the two-factor interactions were

significant in the relative gap response model, but most of the two-factor interactions were significant in the required computational effort response model.

Table 14. Standard Least Squares Regression Coefficient Estimates for Relative Optimality Gap and Req. Comp. Effort Responses - *altdet-b* Model Variant

	Relative Gap (%)				Req'd Comp. Effort (sec)			
Term	**Estimate**	**Std Error**	**t Ratio**	**Prob> $\|t\|$**	**Estimate**	**Std Error**	**t Ratio**	**Prob> $\|t\|$**
$\mathcal{N}$	0.00018	0.00045	0.41	0.6813	162.70111	12.66573	12.85	<.0001
$\mathcal{D}$	0.00264	0.00045	5.91	<.0001	251.83217	12.66573	19.88	<.0001
$\mathcal{J}$	0.00016	0.00011	1.47	0.1411	34.51169	3.16643	10.90	<.0001
$\mathcal{I}$	0.00111	0.00045	2.48	0.0133	36.86161	12.66573	2.91	0.0037
$\mathcal{K}$	-0.00034	0.00022	-1.54	0.1251	27.39340	6.33287	4.33	<.0001

Table 14 presents the results attained when fitting two SLS models to the *altdet-b* data that is summarized in Table 10. These results likewise show that $\mathcal{D}$ and $\mathcal{I}$ are significant factors in the SLS regression model for the relative optimality gap response. Every factor is significant in the regression model for the required computational effort response. One difference for the *altdet-b* model variant is that $\mathcal{D}$ has the highest regression coefficient estimate for the required computational effort response, whereas $\mathcal{N}$ has the highest regression coefficient for the other model variants, even though it is still second highest for the *default-b* model variant.

Table 15. Standard Least Squares Regression Coefficient Estimates for Relative Optimality Gap and Req. Comp. Effort Responses - *altint-b* Model Variant

	Relative Gap (%)				Req'd Comp. Effort (sec)			
Term	**Estimate**	**Std Error**	**t Ratio**	**Prob> $\|t\|$**	**Estimate**	**Std Error**	**t Ratio**	**Prob> $\|t\|$**
$\mathcal{N}$	-0.00041	0.00036	-1.14	0.2552	307.86141	13.11231	23.48	<.0001
$\mathcal{D}$	0.00169	0.00036	4.71	<.0001	136.72409	13.11231	10.43	<.0001
$\mathcal{J}$	0.00025	0.00009	2.82	0.0050	23.61705	3.27808	7.20	<.0001
$\mathcal{I}$	-0.00060	0.00036	-1.66	0.0975	-25.41281	13.11231	-1.94	0.053
$\mathcal{K}$	-0.00016	0.00018	-0.89	0.3731	9.82555	6.55616	1.50	0.1344

Table 15 presents the results attained when fitting two SLS models to the *altint-b* data that is summarized in Table 11. These results show that $\mathcal{D}$ and $\mathcal{J}$ are significant factors in the SLS regression model for the relative optimality gap response. Only $\mathcal{N}$, $\mathcal{D}$, $\mathcal{J}$ are significant in the regression model for the required computational effort response. Similar to the other model variants' SLS regression models, $\mathcal{N}$ and $\mathcal{D}$ have

the highest regression coefficient estimates for the required computational effort response. Similar to the methodology for the other three higher-quality model variants, another regression model that included two-factor interactions was constructed for each response. The collective results were almost identical to those of the *default* model variant.

In summary, the results shown in Tables 12-15 indicate that subdividing a defender's area into more stages will likely result in a higher required computational effort. Since the defender only uses one interdiction resource per stage, such a decomposition into a greater number of stages yields more opportunities for a defender's assets to interdict an intruder, and so a natural tradespace exists. Whereas the number of stages is significant in each of the three required computational effort regression models, the other four main effects are significant as well. Thus, increasing the size of any scenario feature will likely lead to an increase in the computational effort required to solve it. Second, the results show that $\mathcal{D}$ and $\mathcal{I}$ are usually significant predictors to the relative optimality gap response, depending on the model variant employed. That is, the more *types* of assets that the defender is able to place, the better each of the three model variants perform with respect to the relative optimality gap achieved.

2.4 Conclusions

Given two respective sets of detection resources and interdiction resources, each having different types of resources with heterogeneous capabilities, this research addresses the problem of locating and allocating them over a sequence of spatially-defined stages to effectively detect and intercept an intruder. We set forth a mixed-integer nonlinear mathematical programming model – and seven alternative variants – to address the underlying problem using a leading commercial solver for global optimization.

For the Resource Allocation for Intruder Detection and Interdiction (RAIDI) model variants, this work formalizes the definition of problem scenarios as they relate to key parameters relating to instance size with the intent of determining the relative effectiveness of model variants to attain high-quality solutions quickly to RAIDI scenario instances. Development of the model variants allows for a structural examination of scenarios of many different sizes, and which enables a study to identify which factors in the RAIDI scenarios influence the solution quality found by the model variants.

We first use three baseline scenarios to determine which of the eight model variants are of high quality as indicated by the relative optimality gaps achieved and computational effort required to solve instances, and the ones that are deemed ineffective are set aside. The model variants of higher quality are examined further by using a fractional factorial design to fit simple linear regression (SLS) regression models to two responses: relative optimality gap and required computational effort. The main effects in these regression models are simply the feature levels in the RAIDI scenarios. Testing results identified that the number of types of detection and interdiction resources are the significant factors in determining the relative optimality gap obtained by the model variants, and that every feature level is a significant factor in determining the computational effort required to solve an instance of a RAIDI scenario.

After examining the results of both the baseline scenario metrics and the fractional factorial experiments for the higher-quality model variants, it is useful to note that there is a natural tradespace between solution quality (measured by relative optimality gap obtained) and the computational effort required to solve an instance. Generally speaking, the better a model variant performs when solving an instance in one of those two metrics, the worse it performs in the other. Another finding shows

that when the number of stages in a scenario is increased, there is a significant increase in the computational effort required for a model variant to solve the scenario instance. Last, results in Section 2.3.3 show that both the number of detection asset types, $\mathcal{D}$, and interdiction asset types, $\mathcal{I}$, are statistically significant to the relative optimality gap response obtained by the model variants to solve RAIDI scenario instances.

The superlative RAIDI model variant identified via two phases of empirical testing is the *default-b* model, which augments the default model with simple upper and lower bounds on each of the probability calculations to enhance the performance of the commercial solver's branch-and-bound procedure. This model variant performed extremely well across all scenario sizes with respect to both of the responses measured. Moreover, the results of fitting two SLS regression models (i.e., one each for the relative optimality gap response and the required computational effort response) show that $\mathcal{D}$ and $\mathcal{I}$ are very significant predictors of the relative optimality gap, especially $\mathcal{D}$. These outcomes, combined with the relatively low standard deviations for both relative optimality gap obtained and computational effort required when solving 30 randomly generated instances each for 27 different RAIDI scenarios, indicate that the *default-b* model variant performs consistently across many differently-sized RAIDI scenarios and is worthy of use when considering more complicated modeling frameworks in a sequel to this work. It is clear that this model variant is no different than the others in the sense that there is a clear trade off between solution quality and the computational effort required to solve an instance of a RAIDI scenario, but the benefits mentioned above are unique to the *default-b* model variant.

There are multiple areas of future research for this problem thread. Regarding the intruder, introducing the concept of multiple intruder paths would create several possibilities for further study in contrast to the single intruder path studied herein. First, there may exist uncertainty regarding where a single intruder will travel, given

a set of possible intruder paths. This new problem may be addressed via robust optimization or stochastic programming, depending on the information available to the decision maker. Second, a problem that introduces uncertainty about the paths over which multiple intruders will travel has merit for study. In either case, a refined version of the RAIDI model can be used, and the superlative RAIDI model variant identified herein provides a foundational framework for modeling such interactions.

III. The Weighted Intruder Path Covering Problem

3.1 Introduction

Nations, states, and territories must protect their sovereignty against would-be intruders, and that protection often entails the location and use of disparate resources to detect and intercept those intruders. One such problem is ballistic missile defense (BMD), wherein a defender must array sets of radars and interceptors to defend cities (i.e., population centers) against an adversary's launch of ballistic missiles. Other problems such as border protection, the interdiction of refugee movement, cybersecurity, and even the prevention of spread by natural biological immune systems are likewise characterized by similar objectives, resource-outcome relationships, and constraints. Each of these motivating scenarios entails a defender emplacing fixed detection and interdiction resources and an intruder or multiple intruders attempting to traverse the spatial region via a set of paths, where the path used may be unknown to the defender. As it relates to a BMD scenario, intruder missiles seek to destroy high value assets (HVAs) in the defender's territory. In the refugee and border protection scenario, refugees seek safe havens and resources such as shelter and water within the territory of the defender (Mahecic, 2020). In the cybersecurity scenario, a hacker may attempt to steal sensitive user data from a server within a computer network (Schlesinger and Solomon, 2020). In the immune system scenario, an infection may attempt to spread to attack vital organs via the bloodstream (O'Connell and Cafasso, 2018). Moreover, an enterprise modeling approach for defensive asset location is worth examination; subject to the tractability of solution methods, it is preferable to a "systems of systems" approach that decomposes the enterprise and inherently tolerates assumed suboptimality of solutions. An enterprise model to identify the appropriate use of disparate resources to detect and intercept intruders with unknown

path(s) for intrusion has merit for a variety of applications.

In a preceding work to this study, Haywood et al. (2020) examined a related problem wherein a single intruder attempts to traverse a region partitioned by the defender into physical stages (i.e., subregions). In each stage, the defender can place limited detection and interdiction resources to intercept an intruder traveling on a predetermined path. This research extends the previous work, both with respect to modeling and solution methodology. From a modeling perspective, it improves the level of fidelity in three aspects: 1) adopting a multi-objective optimization framework that accounts for the cost of resources used by a defender, 2) introducing uncertainty regarding which path(s) intruder(s) will traverse, and 3) modeling the ability of selected defensive resources to serve the dual-purpose of both detecting and interdicting intruders. Haywood et al. (2020) addressed the cost of the limited resources by making the higher-quality detection and interdiction resources less plentiful for the defender's use. This research seeks to introduce a cost objective for detection and interdiction resources and more thoroughly investigate the tradespace between cost and effectiveness for various defense configurations. Haywood et al. (2020) did not address the concept of intruder path uncertainty, and this research attempts to address it by also seeking, as an objective, to minimize the maximum expected damage over any of the intruder paths under consideration. For the resulting mixed-integer, nonlinear programming formulation this research tests selected metaheuristics designed for multi-objective optimization vis-à-vis a leading commercial solver for global optimization, the latter of which we demonstrate has limited efficacy for solving instances of the underlying problem.

The motivating scenarios illustrate a need to examine the uncertainty in the path an intruder will traverse. In the BMD scenario, a realistic view is that the actual path of an intruder missile is unknown, but there are some paths are more likely to

be traversed than others. In the border scenario, the actual path refugees may take is unknown, but historical data may indicate more commonly traveled paths and inform probabilities with which each path may be used. In the cybersecurity scenario, many firewalls may be in place, but the actual path an intruder may attempt to use when breaching the servers is unknown. In an immune system, an infection will generally travel along the path of least resistance but, if there are multiple avenues of least resistance, then the path is not known with certainty. Figure 4 in Section 3.3.2 provides a graphical depiction of a representative spatial relationship between stages, intruder paths, and possible locations for detection assets, interdiction assets, or dual-purpose assets; the notation therein will be formally defined in Section 3.2.

In each of these different scenarios, a multi-objective optimization approach is appropriate; there is a natural trade-space to examine between the cost and performance of an enterprise. For the aforementioned motivating problems and their structural similarities, this research seeks to address the following problem:

> *Given an intruder attempting to traverse a spatially-decomposed region via multiple possible paths, effectively and cost-efficiently identify a defensive strategy that locates sets of detection resources and interdiction resources, each of which has different types of resources that vary by cost and capability.*

Within the context of the related literature, this research makes two contributions. In its first contribution to address the underlying problem, this research sets forth a mathematical programming model having several collectively complicating aspects that differentiate it from other research in the literature, as reviewed in Section 3.1.1. The model addresses the location of assets across an enterprise comprised of different asset types (i.e., detection and interdiction assets) and capabilities, including dual-purpose assets representing actual assets for certain motivating scenarios (e.g., AEGIS class destroyers in a BMD scenario). The enterprise approach of the

model considers the location of these assets in a defender's territory organized into multiple stages, better representing the geographic boundaries often used to organize defenses for related applications (e.g., border patrol). Finally, the model employs a multi-objective approach to enable the examination of the tradeoffs between the effectiveness and cost of defensive asset configurations. In its second contribution, this research identifies and empirically tests alternative, conceptually sound solution methodologies for instances of the underlying problem. Empirical testing first identifies the instance size-specific limitations of a leading commercial, global optimization solver, motivating the examination of metaheuristics. Subsequent testing compares the relative efficacy of two metaheuristics for solving larger-sized instances, identifying the superlative technique that provides practical utility to the relevant mathematical programming model presented in the first contribution.

The remainder of this paper is organized as follows. Section 3.1.1 reviews the relevant literature to the application of interest, as well as the literature that informs either the modeling approach or solution methodologies examined herein. Section 3.2 presents the mathematical programming formulation, and Section 3.3 validates the model for an illustrative instance and conducts the aforementioned empirical testing. Section 3.4 concludes the paper with a summary of resulting insights and suggestions for future research.

3.1.1 Literature Review

Although the published literature does not address the underlying problem examined herein, it does both inform our modeling approach and provide alternative, candidate solution methodologies that we consider and empirically test.

Several related modeling techniques from the literature provide insight, yet none we identified embraces the complexity of the problem examined in this research. On

a superficial level, most resource location problems emplace a single asset type to address a particular demand (e.g., Bell et al. (2011), Basciftci et al. (2021)). Even works that emplace multiple asset types (e.g., Serafino and Ventre (2016), Paul et al. (2017)) typically consider each asset type to have homogenous capabilities, whereas the problem studied herein allows a defender to emplace multiple asset types – with each type having a range of specific asset capabilities – and with asset types contributing differently to the objective of interdicting an intruder. Moreover, this research utilizes an enterprise model in which the stated goal is to detect and subsequently interdict an intruder using dedicated assets for each task and over all potential intruder paths and multiple stages. The complexity embraced by this modeling endeavor improves upon the literature that considers a single intruder path and/or stage (e.g., Hausken (2010), Karabulut et al. (2017), Lessin et al. (2019)).

Path covering research is a literature thread more closely related to the motivation for this research, wherein a user seeks to emplace facilities that cover paths or routes rather than fixed-point demands. In one such example, Capar et al. (2013) examined the Flow-Refueling Location Model (FRLM) to locate alternative-fuel station locations for use by vehicles along their routes. In related extensions, Upchurch et al. (2009) considered a capacitated variant of the problem, and Capar et al. (2013) studied heuristic solution methods. An abundance of literature pertaining to similar applications (e.g., vehicle recharging stations, aircraft refueling locations) exists, but it typically considers vehicle ranges and adopts a cooperative approach for facility location. Moreover, the research herein differs in complexity from traditional path covering problems due to the path of an intruder being decomposed into multiple stages as opposed to a single path. As such, this literature motivates but does not directly inform the research herein; it is more strongly informed by traditional facility location models.

Given this work seeks to defend (i.e., cover) numerous intruder paths with a (cost-)limited number of resources, both the Maximal Covering Location Problem (MCLP) and the Maximal Expected Covering Location Problem (MEXCLP) provide useful modeling perspectives. Church and ReVelle (1974) introduced the MCLP, which seeks to cover the maximum amount of demands, subject to bounds on the number (or cost) of emplaced facilities, wherein *coverage* of a demand is a binary characterization. Daskin (1983) extended the MCLP via the MEXCLP modeling framework, wherein each resource has a probability of being busy (i.e., unable to provide coverage), and the expected coverage of demands is maximized. A reader interested in the greater context of location theory would benefit from examining works by Daskin (2011) and Church and Murray (2018). Of relevance to this research is the general MEXCLP framework that considers probabilities of coverage and the expected coverage attained across the enterprise of resource emplacement. An interesting optimization problem that allocates rectangular strips across a rectangular region is studied by Hu et al. (2021), an example of a coverage problem wherein a user locates assets to cover a spatial demand rather than traditional point-based demands. In comparison, the research herein seeks to provide coverage to paths rather than point demands via located resources.

Regarding solution methodologies utilized herein pertaining to multi-objective optimization, this research considers both the effectiveness of system performance and the efficient use of limited resources. There are two different frameworks for multi-objective optimization regarding the preferences of a decision-maker over objectives: *a priori* and *a posteriori* (Marler and Arora, 2004). An *a priori* framework entails a decision-maker articulation of priorities before identifying a solution, and only one Pareto optimal solution need be identified. Alternatively, an *a posteriori* framework identifies the set (or a subset) of Pareto solutions, characterizing the tradespace for a

decision maker to consider and possibly discriminating among the solutions to develop a recommendation (e.g., via proximity to an ideal point) (Marler and Arora, 2004). Given the potential benefit of deriving insights attainable by examining the tradeoffs between across a Pareto front, this research embraces an *a posteriori* framework.

Among the methods used to identify Pareto optimal solutions to multi-objective optimization problems are the Weighted Sum Method, the ε-constraint Method, compromise programming (i.e., a method of weighted metrics), and scalarizing functions (Ehrgott (2005); Deb (2014)). The literature in the field of multi-objective optimization is vast, and we recommend the works by Deb (2001), Marler and Arora (2004), and Ehrgott (2005) for an interested reader. When solving multi-objective optimization models that are computationally challenging, the precise identification of Pareto optimal solutions may be challenging to traditional optimization methods, motivating the use of metaheuristics. Gonzalez et al. (2020) explored the use of a simulation algorithm to solve multi-objective optimization problems instead of a global solver, with mixed results. Talbi et al. (2012) review this developing area of the multi-objective optimization literature, including the use of both non-evolutionary approaches (e.g., local search, Simulated Annealing, Tabu Search) as well as hybrid metaheuristics (e.g., Multi-objective Genetic Local Search).

This work solves instances of the problem using selected variants of a class of metaheuristics known as genetic algorithms (GA). Holland et al. (1975) pioneered GA development, embedding concepts from nature and evolutionary processes. For the purpose of multi-objective optimization (MOO), many variations of GAs have been developed to explore Pareto fronts. Two noteworthy algorithmic components developed within the MOO GA research thread are *elitism* and the use of an external population. Elitism ranks population members by fitness level and ensures the most fit members survive to the next generation, and it is employed in notable works by

Murata and Ishibuchi (1995) and Deb et al. (2002). Murata and Ishibuchi (1995) invented the MOO GA that randomly assigns weights to each objective function for each population member in an effort to more thoroughly explore the Pareto front, addressing the shortcoming they identified with other MOO GAs. Adopting the naming convention used by Konak et al. (2006), the algorithm developed by Murata and Ishibuchi (1995) will hereafter be referred to as the Random Weight Genetic Algorithm (RWGA). In another major development, Srinivas and Deb (1994) created the Nondominated Sorting Genetic Algorithm (NSGA), which does not use elitism but instead simply makes use of fitness sharing via niching to generate diversity among subsequent populations. Deb et al. (2002) improved upon this algorithm in creating NSGA-II, which is widely considered to be one of, if not the, best MOO GAs in the literature. NSGA-II has success beyond traditional use, as evidenced in research by (Rabbani et al., 2019), wherein the authors used it in conjunction with Monte Carlo simulation to create a *simheuristic* to solve MINLP models. Similarly, research conducted by Drake et al. (2020) employs NSGA-II as one of several metaheuristics to solve a multi-objective optimization problem involving deployment of resources for infrastructure networks, and NSGA-II emerges as the best of the tested MOO GAs. Traditionally, NSGA-II uses elitism when iterating through generations and, as the name suggests, assigns fitness values via the use of rankings by examining whether a solution in the population is dominated. The second major component, the use of external populations, consists of maintaining a distinct, secondary population and introducing members of it to the main population when creating the subsequent generation. This concept is observable in RWGA (Murata and Ishibuchi, 1995); the algorithm maintains a separate population consisting of solutions heretofore identified as Pareto optimal and introduces a subset of them to the next generation. For a detailed comparisons between various MOO GAs, we refer an interested reader to

comprehensive surveys by Zitzler et al. (2000) and Konak et al. (2006).

Within this research, we apply and compare both the RWGA and NSGA-II multi-objective GAs. We selected these metaheuristics due to the persistently high performance each method has exhibited when tested on various problems, both within the literature and during preliminary testing on instances of the underlying problem for this research, and because they are quite different in the mechanisms employed for diversity, elitism, use of external populations, and fitness assignment.

3.2 Model and Solution Methodology

First, we develop a model of the problem in which a single intruder attempts to traverse the stages of a defensive region where the intruder's path is unknown but is limited to a finite set of possible paths. Moreover, this model is initially tri-objective, with the objectives respectively seeking to 1) minimize the expected damage caused by the intruder, 2) minimize the maximum expected damage done by the intruder, and 3) minimize the cost of the defense configuration.

To formulate the mathematical program to address the underlying problem, it is necessary to define the following sets, parameters, and decision variables.

Sets

- $P = \{1, 2, ..., \mathcal{P}\}$ is the set of paths over which the intruder may traverse through the defender's territory, indexed by ψ.

- $N = \{1, 2, ..., \mathcal{N}\}$ is the set of distinguishable stages over which the intruder may be detected and interdicted by the defender's enterprise of sensors and interdictors, indexed by n. (If $\mathcal{N} = 1$, the following models remain valid, but the indexing of selected sets, parameters, decision variables, and constraints on n can be set aside.) Relative to the set of stages, two assumptions are made regarding the intruder's path. First, we assume that each path transits every

stage. Second, the stages are numbered in ascending order, as an intruder would encounter them when traversing any path.

- $D = \{1, 2, ..., \mathcal{D}\}$ is the set of different detection resource types, indexed by d, each of which pertains to different capabilities (e.g., range, effectiveness).

- $J = \{1, 2, ..., \mathcal{J}\}$ is the set of possible locations at which detection resources can be located, indexed by j. J is partitioned by stage, where $\bigcup_{n \in N} J_n = J$.

- $I = \{1, 2, ..., \mathcal{I}\}$ is the set of different interdiction resource types, indexed by i, each of which has different capabilities (e.g., speed, range, probability of success).

- $K = \{1, 2, ..., \mathcal{K}\}$ is the set of possible locations at which interdiction resources can be located, indexed by k. Similar to set J, the set K is likewise partitioned over N.

- $B = \{1, 2, ..., \mathcal{B}\}$ is the set of dual-purpose resource types (i.e., resources that can both detect *and* interdict an intruder), indexed by b, each of which pertains to different capabilities (e.g., speed, range, probability of interdiction, probability of detection).

- $L = \{1, 2, ..., \mathcal{L}\}$ is the set of possible locations at which dual-purpose resources can be located, indexed by l. Similar to sets J and K, the set L is likewise partitioned over N.

Parameters

- v_ψ: the expected damage that an intruder on path ψ would inflict if not interdicted. As formulated, the math program considers an intruder seeking to traverse each of the paths $\psi \in P$ simultaneously. It may also address a single

intruder considering which one of the paths to traverse. For such a case, assuming a probability distribution of the intruder over the paths, v_ψ is the likelihood the intruder will traverse that path, multiplied by the damage induced if they successfully traverse it.

- $u_d^{\mathbb{D}}, u_i^{\mathbb{I}}, u_b^{\mathbb{B}}$: the maximum number of detection, interdiction, and dual-purpose resources that can be emplaced, respectively of types d, i, and b.

- $c_d^{\mathbb{D}}, c_i^{\mathbb{I}}, c_b^{\mathbb{B}}$: the cost of emplacing a detection, interdiction, and dual-purpose resources, respectively of types d, i, and b.

- $p_{dj\psi}^{\mathbb{D}}$: the probability that an intruder on path ψ is detected by a detection resource of type d emplaced at location j.

- $p_{ik\psi}^{\mathbb{I}}$: the probability that an intruder on path ψ is interdicted by an interdiction resource of type i emplaced at location k.

- $p_{bl\psi}^{\mathbb{BD}}, p_{bl\psi}^{\mathbb{BI}}$: the probability that an intruder on path ψ is detected or interdicted, respectively, by a dual-purpose resource of type b emplaced at location l.

- $w_e, w_{\max}, w_c$: the relative weights assigned, respectively, to the expected total damage, the worst-case path-specific damage, and the enterprise cost when solving an instance of the problem with a commercial solver via the Weighted Sum Method for MOO

Decision Variables

- x_{dj} : a binary variable equal to 1 if a detection resource of type d is emplaced at location j, and 0 otherwise.

- y_{ik} : a binary variable equal to 1 if an interdiction resource of type i is emplaced at location k, and 0 otherwise.

- z_{bl} : a binary variable equal to 1 if a dual-purpose resource of type b is emplaced at location l, and 0 otherwise.

- $\pi^{\mathbb{D}}_{\psi n}$: the conditional probability that an intruder on path ψ is detected in stage n given it has successfully traversed previous stages, i.e., $1, \ldots, n-1$.

- $\pi^{\mathbb{I}}_{\psi n}$: the conditional probability that an intruder on path ψ is interdicted in stage n given it has successfully traversed previous stages, i.e., $1, \ldots, n-1$ and has been detected in stage n.

- $\pi^{\mathbb{D}\bigcap\mathbb{I}}_{\psi n}$: the conditional probability of an intruder on path ψ being detected *and* subsequently interdicted in stage n given it has successfully traversed previous stages, i.e., $1, \ldots, n-1$.

- $\pi^{\mathbb{D}\bigcap\mathbb{I}}_{\psi}$: the probability of an intruder on path ψ being detected and subsequently interdicted.

- f_e: the total expected damage done by the intruder.

- f_{max}: the worst-case expected damage done by the intruder.

- f_c: the cost of the defense configuration employed by the defender.

Leveraging the aforementioned notation, we formulate the **Weighted Intruder Path Covering** (**WIPC**) model as follows.

$$\min \quad (f_e, f_{max}, f_c) \tag{23}$$

$$\text{s.t.} \quad f_e = \sum_{\psi \in P} v_\psi \left(1 - \pi^{\mathbb{D}\bigcap\mathbb{I}}_{\psi}\right), \tag{24}$$

$$f_{max} \geq v_\psi \left(1 - \pi^{\mathbb{D}\bigcap\mathbb{I}}_{\psi}\right), \qquad \forall \psi \in P \tag{25}$$

$$f_c = \sum_{d \in D} \sum_{j \in J} c^{\mathbb{D}}_d x_{dj} + \sum_{i \in I} \sum_{k \in K} c^{\mathbb{I}}_i y_{ik} + \sum_{b \in B} \sum_{l \in L} c^{\mathbb{B}}_b z_{bl}, \tag{26}$$

$$\pi_{\psi}^{\mathbb{D}\bigcap\mathbb{I}} = 1 - \prod_{n\in N}\left(1 - \pi_{\psi n}^{\mathbb{D}\bigcap\mathbb{I}}\right), \qquad \forall\ \psi \in P \quad (27)$$

$$\pi_{\psi n}^{\mathbb{D}\bigcap\mathbb{I}} = \pi_{\psi n}^{\mathbb{D}}\pi_{\psi n}^{\mathbb{I}}, \qquad \forall\ \psi \in P,\ n \in N, \quad (28)$$

$$\pi_{\psi n}^{\mathbb{D}} = 1 - \prod_{d\in D}\prod_{j\in J_n}\left(1 - p_{dj\psi}^{\mathbb{D}}\right)^{x_{dj}}\prod_{b\in B}\prod_{l\in L_n}\left(1 - p_{bl\psi}^{\mathbb{B}\mathbb{D}}\right)^{z_{bl}}, \quad \forall\ \psi \in P,\ \forall\ n \in N, \quad (29)$$

$$\sum_{d\in D} x_{dj} \leq 1, \qquad \forall j \in J, \quad (30)$$

$$\pi_{\psi n}^{\mathbb{I}} = 1 - \prod_{i\in I}\prod_{k\in K_n}\left(1 - p_{ik\psi}^{\mathbb{I}}\right)^{y_{ik}}\prod_{b\in B}\prod_{l\in L_n}\left(1 - p_{bl\psi}^{\mathbb{B}\mathbb{I}}\right)^{z_{bl}}, \quad \forall\ \psi \in P, \forall\ n \in N, \quad (31)$$

$$\sum_{i\in I}\sum_{k\in K_n} y_{ik} + \sum_{b\in B}\sum_{l\in L_n} z_{bl} \leq 2, \qquad \forall\ n \in N, \quad (32)$$

$$\sum_{j\in J} x_{dj} \leq u_d^{\mathbb{D}}, \qquad \forall d \in D, \quad (33)$$

$$\sum_{k\in K} y_{ik} \leq u_i^{\mathbb{I}}, \qquad \forall i \in I, \quad (34)$$

$$\sum_{l\in L} z_{bl} \leq u_b^{\mathbb{B}}, \qquad \forall b \in B, \quad (35)$$

$$x_{dj} \in \{0, 1\}, \qquad \forall d \in D,\ j \in J \quad (36)$$

$$y_{ik} \in \{0, 1\}, \qquad \forall i \in I,\ k \in K \quad (37)$$

$$z_{bl} \in \{0, 1\}, \qquad \forall b \in B,\ l \in L \quad (38)$$

$$0 \leq \pi_{\psi n}^{\mathbb{D}} \leq 1, \qquad \forall\ \psi \in P,\ \forall\ n \in N, \quad (39)$$

$$0 \leq \pi_{\psi n}^{\mathbb{I}} \leq 1, \qquad \forall\ \psi \in P, \forall\ n \in N, \quad (40)$$

$$0 \leq \pi_{\psi n}^{\mathbb{D}\bigcap\mathbb{I}} \leq 1, \qquad \forall\ \psi \in P, \forall\ n \in N, \quad (41)$$

$$0 \leq \pi_{\psi}^{\mathbb{D}\bigcap\mathbb{I}} \leq 1, \qquad \forall\ \psi \in P, \quad (42)$$

$$0 \leq f_e \leq \sum_{\psi\in P} v_{\psi}, \quad (43)$$

$$f_{\max} \leq \max_{\psi\in P}\{v_{\psi}\}, \quad (44)$$

$$0 \leq f_c \leq \mathcal{J} \cdot \max_{d}\{c_d^{\mathbb{D}}\} + \mathcal{K} \cdot \max_{i}\{c_i^{\mathbb{I}}\} + \mathcal{L} \cdot \max_{b}\{c_b^{\mathbb{B}}\}, \quad (45)$$

The objective (23) of this formulation minimizes each of three objective functions, as respectively either calculated or bound via Constraints (24)-(26). Constraint (24) calculates the expected damage inflicted by intruders over all paths $\psi \in P$. Constraint (25) bounds the maximum expected damage from below by the expected damage that will occur on each of the intruder paths $\psi \in P$. Note that, although this model is initially tri-objective, we later simplify to a bi-objective model after initial testing indicates a high correlation between two of the objectives, as further discussed in Section 3.3.2. Constraint (26) calculates the cost of the emplaced resources. Constraint (27) calculates the probability of detection-and-interdiction of an intruder on each path $\psi \in P$ using the stage-specific conditional probabilities of detection-and-interdiction, assuming independence between stages. Constraint (28) computes the stage-specific conditional probabilities of detection-and-interdiction for an intruder on each path $\psi \in P$, likewise assuming independence between these probabilities. To calculate the in-stage probability of detection for an intruder on each of the paths $\psi \in P$, it is assumed that *every* detection resource and dual-purpose resource emplaced in the stage contributes to the overall in-stage probability of detection. Constraint (29) leverages this assumption in calculating each path-and-stage-specific probability of detection as a function of the detection and dual-purpose resources emplaced in stage n and the respective probabilities specific to each type of resource. Constraint (30) limits the number of detection resources that can be emplaced at each location $j \in J$ to at most one. To calculate each path-and-stage-specific probability of interdiction via Constraint (31), a different assumption is made regarding the interdiction and dual-purpose assets; at most two interdiction or dual-purpose assets in each stage can be utilized to attempt to interdict an intruder on each path. This assumption is motivated by the literature for the BMD scenario, which indicates firing more than

two interceptor missiles is not effective (Wilkening, 2000). Accordingly, Constraint (32) enforces that at most two interdiction or dual-purpose assets can be utilized for interdiction on a given intruder path within a stage. Constraints (33)-(35) ensure the total number of detection, interdiction, and dual-purpose resources, respectively, do not exceed the allotted amount. Constraints (36)-(38) enforce binary restrictions on selected decision variables, and Constraints (39)-(42) bound each of the computed probabilities to impose a hypercube of constraints on the related decision variables to support the application of a global optimization (i.e., branch-and-bound) algorithm via a commercial solver. For similar reasons, Constraints (43)-(45) enforce lower and upper bounds on the objective function calculations.

Three solution methods are considered and empirically tested in Section 3.3 to solve instances of the WIPC. The research first uses the Branch-And-Reduce Optimization Navigator (BARON) (Sahinidis and Tawarmalani, 2004), a commercial, global solver designed for the global optimal solution of mixed-integer nonlinear programs (MINLPs) such as WIPC. The commercial global optimization solver BARON was selected from among several alternatives (i.e., Bonmin, COUENNE, LindoGlobal, and SCIP) based on its superlative performance during preliminary empirical testing for instances of the problem. Additionally, testing examines the multi-objective genetic algorithms RWGA and NSGA-II, as discussed in Section 3.1.1.

3.3 Testing, Results, and Analysis

Before examining the limitations of the commercial solver BARON, Section 3.3.1 describes the method utilized to stochastically generate parameters for test instances, and Section 3.3.2 validates the WIPC model for a small, illustrative instance. Section 3.3.3 empirically examines the limitations of the commercial solver BARON and motivates the development and use of a metaheuristic to solve larger instances of WIPC.

Section 3.3.4 demonstrates the superior effectiveness and efficiency of RWGA and NSGA-II to BARON. Finally, Section 3.3.5 assesses the effectiveness and efficiency of RWGA and NSGA-II for larger-sized instances.

3.3.1 Test Instance Generation

To enable a relatively focused testing design, set sizes and selected parameters for WIPC instances are user-defined and deterministic, with the remainder of parameters stochastically generated based on the aforementioned user-defined values. For a given test instance, specified are the number of paths $\mathcal{P}$, the number of stages $\mathcal{N}$, and the respective numbers and types of each defensive resource (i.e., $\mathcal{D}, \mathcal{J}, \mathcal{I}, \mathcal{K}, \mathcal{B}, \mathcal{L}$). Although not a direct set or parameter of WIPC, testing also considers a user-determined size of the defended region; herein, a rectangular region is assumed with an intruder traversing (w.l.o.g.) from left to right. The rectangular region we consider has a width of 6400 units and a height of 2300 units, which roughly emulates, from the perspective of BMD, the aspect ratio and approximate dimensions (in miles) of a region of interest in the Northern Pacific Ocean. Using this width, the respective widths of stages are assumed to be uniform (i.e., with each stage width equal to $6400/\mathcal{N}$).

For each set of these affixed values, which we hereafter refer to as a *scenario*, testing considers multiple instances, wherein each instance differs via selected, stochastically generated parameters. In testing throughout Sections 3.3.2- 3.3.4, this research generates WIPC instances for a given scenario in the following manner. First, instances are generated using a fixed pseudo-random number generation seed in GAMS for Sections 3.3.2 and 3.3.3, and in R for Section 3.3.4. The allocation of possible resource locations to stages (i.e., J_n, K_n, and L_n) are respectively identified with a discrete uniform distribution with the proviso that at least one location option for

each category and type of resource exists in each stage. Once the detection, interdiction, and dual-purpose resource locations are assigned to stages, their respective locations within the stages are calculated via a uniform distribution to designate a two-dimensional Cartesian coordinate. Intruder paths are calculated by assuming that intruders will traverse the territory in a straight line from an origin (i.e., launch site) on the left edge of the rectangular region to a destination (i.e., target) on the right edge of the rectangular region. The vertical coordinate for each of these points (i.e., on the left or right boundary or the region) for a given path is generated using a uniform distribution $U(0, 2300)$.

Another parameter generated to create an instance of WIPC are the values of v_ψ. These values are sampled randomly from the set of the populations of the 20 most populated cities on the US West Coast. For example, if $\mathcal{P} = 7$ for an instance of WIPC, seven numbers are chosen at random (with replacement) from the aforementioned set to generate the v_ψ-values. The larger population values are interpreted as larger values of v_ψ because an intruder targeting a highly populated city is assumed to have the ability to cause more damage.

When generating specific parameter values, we assume that higher values of the indicies d, i, and b correspond to more capable detection, interdiction, and dual-purpose resource types, respectively, which are also assumed to be more expensive and available in lesser amounts. We also expect detection resources to be more abundant than interdiction and dual-purpose resources and we expect that dual-purpose resources are inherently more expensive than the other two resource types. We thus generate $u_d^{\mathbb{D}}$-, $u_i^{\mathbb{I}}$-, and $u_b^{\mathbb{B}}$-parameters as a function of the scenario parameters using a uniform random variable. The lower bound on $u_d^{\mathbb{D}}$ is induced by an assumption that at least one detection resource will be emplaced in each stage, and an upper bound is induced by a combination of the policy that detection resources will not

be co-located and the number of total detection resource locations. Equation (46) illustrates for $u_d^{\mathbb{D}}$, $\forall\ d \in D$, the generation of a value from a uniform distribution and the allocation allocation of an integer-valued proportion of that value by resource type. Similarly, for interdiction and dual-purpose resource types, Equations (47) and (48) illustrate the generation of a value from uniform distributions for $u_i^{\mathbb{I}}$, $\forall\ i \in I$, and $u_b^{\mathbb{B}}$, $\forall\ b \in B$, respectively. The upper bound for the uniform distribution in Equation (47) differs from that in Equation (46) because we expect the amount of interdiction resources to be less than that of the detection resources. The upper bound in Equation (48) is analogous to that in Equation (46) since the assumptions for the placement of dual-purpose resources mirror that of detection resources in that each resource improves detection within a stage.

$$u_d^{\mathbb{D}} = \left\lceil \left((\mathcal{D} + 1 - d) \,/ \sum_{d \in D} d \right) U(\mathcal{N}, \mathcal{J}) \right\rceil,\ \forall\ d \in D \tag{46}$$

$$u_i^{\mathbb{I}} = \left\lceil \left((\mathcal{I} + 1 - i) \,/ \sum_{i \in I} i \right) U(\mathcal{N}, 2\mathcal{N}) \right\rceil,\ \forall\ i \in I \tag{47}$$

$$u_b^{\mathbb{B}} = \left\lceil \left((\mathcal{B} + 1 - b) \,/ \sum_{b \in B} b \right) U(\mathcal{N}, \mathcal{L}) \right\rceil,\ \forall\ b \in B \tag{48}$$

Cost parameters also vary by resource type (i.e., d, i, and b, respectively), assuming that types with higher indices are the more capable and expensive resources. Accordingly, the costs for each resource type are generated in a manner that assigns higher costs to resource types with higher-valued indices. For detection assets, a cost range of $[c_{LB}^D, c_{UB}^D] = [1, 10]$ is partitioned into $\mathcal{D}$ intervals having equal width, assigning the higher cost intervals to resource types with highest-valued indices, and so forth. We slightly modify the intervals so they overlap by 10% of the interval widths; the result is not a completely hierarchical partition. The costs of interdiction

resources by type are generated in an identical manner with identical lower and upper bounds for cost, but the costs of dual-purpose resources by type are generated using a cost range of $[c^B_{LB}, c^B_{UB}] = [5, 20]$. The reason for the larger bounds of dual-purpose costs is simply due to the dual-purpose nature of the resources. Equations (49)-(51) illustrate the specific generation of the costs by resource type of detection, interdiction, and dual-purpose resources, respectively, wherein $\Delta^{\mathbb{D}} = \frac{c^{\mathbb{D}}_{UB} - c^{\mathbb{D}}_{LB}}{\mathcal{D}}$, $\Delta^{\mathbb{I}} = \frac{c^{\mathbb{I}}_{UB} - c^{\mathbb{I}}_{LB}}{\mathcal{I}}$, and $\Delta^{\mathbb{B}} = \frac{c^{\mathbb{B}}_{UB} - c^{\mathbb{B}}_{LB}}{\mathcal{B}}$.

$$c^{\mathbb{D}}_d = U\left(c^{\mathbb{D}}_{LB} + (d-1)\cdot\Delta^{\mathbb{D}}, c^{\mathbb{D}}_{LB} + d\cdot\Delta^{\mathbb{D}}\right),\ \forall\ d \in D \tag{49}$$

$$c^{\mathbb{I}}_i = U\left(c^{\mathbb{I}}_{LB} + (i-1)\cdot\Delta^{\mathbb{I}}, c^{\mathbb{I}}_{LB} + i\cdot\Delta^{\mathbb{I}}\right),\ \forall\ i \in I \tag{50}$$

$$c^{\mathbb{B}}_b = U\left(c^{\mathbb{B}}_{LB} + (b-1)\cdot\Delta^{\mathbb{B}}, c^{\mathbb{B}}_{LB} + b\cdot\Delta^{\mathbb{B}}\right),\ \forall\ b \in B \tag{51}$$

Probability parameters vary by resource type and are calculated as a function of distance. Specifically, the distance used to calculate probability of detection and interdiction of an intruder is the shortest distance from the resource's location to the intruder path, henceforth referred to as $dist_{\min}$. For example, the minimum distance from a detection location j to an intruder path ψ reads $dist^{\mathbb{D}}_{\min}(j, \psi)$. The probability of detection (which can be accomplished by a detection or dual-purpose resource) from a resource location for a particular intruder is calculated using a logistic decay function of $dist_{\min}$, as mentioned previously. The probability of interdiction (which can be accomplished by an interdiction or a dual-purpose resource) from a resource location for a particular intruder is calculated using an exponential decay function of the aforementioned $dist_{\min}$. Equations (53)-(56) depict the probability functions used to parameterize WIPC instances as functions of $dist_{\min}$ with the appropriate indices,

where $p_{\max} = 0.9$ is the maximum probability of an emplaced resource successfully detecting or interdicting the intruder, and $dist_{0.5}$ indicates the distance at which a resource's probability of successfully detecting or interdicting an intruder is equal to 0.5 and is a function of the resource type index. The $dist_{0.5}$ calculations for detection and interdiction resource types are such that the largest index (i.e., the most effective) resource types have a 0.5 probability of success when the distance to the intruder is 25% of the stage width, and this percentage is reduced for lower indices as shown in Equation (52). This distance is adjusted to 15% of the stage width for dual-purpose resource types, where we assume that dual-purpose resource types will not accomplish the same level of effectiveness at the same distance as a dedicated detection or interdiction resource type.

$$dist^{\mathbb{D}}_{0.5}(d) = \frac{d}{\mathcal{D}} \cdot 0.25 \cdot \text{stagewidth} \tag{52}$$

$$p^{\mathbb{D}}_{dj\psi}\left(dist^{\mathbb{D}}_{\min}(j,\psi)\right) = \frac{\left(\frac{p_{\max}}{1-p_{\max}}\right)\exp\left\{\frac{dist^{\mathbb{D}}_{\min}(j,\psi)}{dist^{\mathbb{D}}_{0.5}(d)}\ln\left(\frac{1-p_{\max}}{p_{\max}}\right)\right\}}{1+\left(\frac{p_{\max}}{1-p_{\max}}\right)\exp\left\{\frac{dist^{\mathbb{D}}_{\min}(j,\psi)}{dist^{\mathbb{D}}_{0.5}(d)}\ln\left(\frac{1-p_{\max}}{p_{\max}}\right)\right\}} \tag{53}$$

$$p^{\mathbb{I}}_{ik\psi}\left(dist^{\mathbb{I}}_{\min}(k,\psi)\right) = p_{\max}\exp\left\{\frac{dist^{\mathbb{I}}_{\min}(k,\psi)}{dist^{\mathbb{I}}_{0.5}(i)}\ln\left(\frac{0.5}{p_{\max}}\right)\right\} \tag{54}$$

$$p^{\mathbb{BD}}_{bl\psi}\left(dist^{\mathbb{B}}_{\min}(l,\psi)\right) = \frac{\left(\frac{p_{\max}}{1-p_{\max}}\right)\exp\left\{\frac{dist^{\mathbb{B}}_{\min}(l,\psi)}{dist^{\mathbb{B}}_{0.5}(b)}\ln\left(\frac{1-p_{\max}}{p_{\max}}\right)\right\}}{1+\left(\frac{p_{\max}}{1-p_{\max}}\right)\exp\left\{\frac{dist^{\mathbb{B}}_{\min}(l,\psi)}{dist^{\mathbb{B}}_{0.5}(b)}\ln\left(\frac{1-p_{\max}}{p_{\max}}\right)\right\}} \tag{55}$$

$$p^{\mathbb{BI}}_{bl\psi}\left(dist^{\mathbb{B}}_{\min}(l,\psi)\right) = p_{\max}\exp\left\{\frac{dist^{\mathbb{B}}_{\min}(l,\psi)}{dist^{\mathbb{B}}_{0.5}(b)}\ln\left(\frac{0.5}{p_{\max}}\right)\right\} \tag{56}$$

3.3.2 Validating the Model with an Illustrative Instance

Figure 4 depicts a small, illustrative instance used to validate WIPC and to illustrate how various objective function weighting combinations affect the optimal solutions attained. The depicted instance consists of $n = 3$ stages with $\mathcal{P} = 3$ possible intruder paths, each having an associated possible damage v_p. For the defender, there are $\mathcal{J} = 4$ possible locations for detection assets, $\mathcal{K} = 4$ possible locations for interdiction assets, and $\mathcal{L} = 5$ possible locations for dual-purpose assets. The defender also has two types of each type of asset from which to choose when deciding which assets to use and where to emplace them.

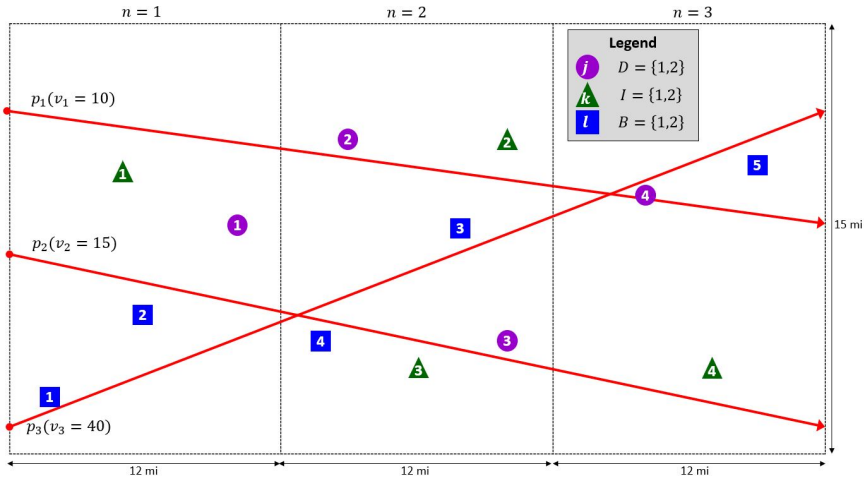

Figure 4. Illustrative Instance of WIPC

As shown in Figure 4, the dual-purpose (and thus, more expensive) assets can only be placed at locations in relatively close proximity to path p_3, which has the highest value of v_ψ. That is, an intruder on path p_3 can inflict the most damage, and it will cost more money to properly defend that path. In contrast, the maximum amount of damage an intruder traversing both paths p_1 and p_2 can inflict is lower, and those paths collectively require less money to defend using a combination of detection and interception assets.

Testing applied the Weighted Sum Method and invoked the commercial solver

BARON to identify multiple Pareto Optimal (PO) solutions for this WIPC instance, examining non-zero objective function weight combinations of $(w_c, w_e, w_{\max})$ in increments of 0.1, such that $w_c + w_{\max} + w_e = 1$. Of note, the GAMS implementation of the WIPC formulation minimizes an intermediate decision variable defined as $z = w_c f_c + w_{\max} f_{\max} + w_e f_e$ and additionally imposes the constraint $z \leq f_c + f_{\max} + f_e$ to bound further any solver-generated relaxations. For increasing values of w_c, Figure 5 plots the optimal values of f_e and $f_{\max}$ identified. For some weight combinations (e.g., $w_c = 0.4$), Figure 5 depicts only one point; this result indicates that all five of the optimal solutions attained for the varying values of $(w_e, w_{\max})$ yielded the same objective function values at optimality.

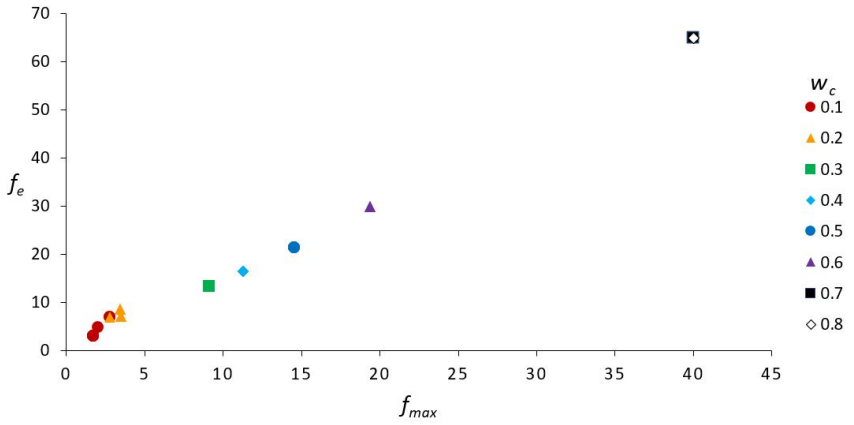

Figure 5. Optimal $(f_{\max}, f_e)$-values via the Weighted Sum Method to the Illustrative WIPC Instance for Various w_c-values

The results depicted in Figure 5 yield two important insights. First, a tradeoff between system cost and effectiveness is evident for this illustrative instance. As w_c increases and the cost of the defense configuration becomes a higher relative priority, the optimal values of both f_e and $f_{\max}$ increase. Second, the optimal values of f_e and $f_{\max}$ are highly correlated ($r = 0.995$) over all combinations of objective function weights examined; there exists redundancy in examining both of these objective functions in the WIPC formulation. Although there is not a guarantee that this result

is generalizable to all instances of WIPC, we find its existence motivation to reduce the number of objective functions considered in an effort to improve tractability. Beginning in Section 3.3.3, only one system effectiveness objective will be considered, rendering WIPC as a bi-objective rather than a tri-objective formulation. Because f_e is a direct calculation via Constraint (29), whereas $f_{\max}$ is determined at optimality relative to the lower bounding supports imposed via Constraint (39), $f_{\max}$ will be set aside as an explicit objective function.

To further validate the formulation for this instance, we restrict our attention to the optimal solutions for $(w_c, w_e, w_{\max}) \in \{(0.1, 0.5, 0.4), (0.4, 0.3, 0.3), (0.7, 0.2, 0.1)\}$, respectively corresponding to low, medium, and high relative priorities on minimizing defender costs with roughly similar priorities over the remaining objective functions (at the 0.1 granularity of objective function weights). Table 16 presents the optimal solutions for each combination of objective function weights. The second column reports the optimal objective function values, and the subsequent columns respectively identify the combination of indices and asset type where detection, interdiction, and dual-purpose assets are emplaced.

Table 16. Optimal solutions for three sample weight combinations

$(w_c, w_e, w_{\max})$	$(f_c^*, f_e^*, f_{\max}^*)$	$(d,j)\|x_{dj}^*=1$	$(i,k)\|y_{ik}^*=1$	$(b,l)\|z_{bl}^*=1$
$(0.1, 0.5, 0.4)$	$(69.9, 3.0, 1.7)$	$(1,4),(2,2),(2,3)$	$(1,3),(2,2),(2,4)$	$(2,4)$
$(0.4, 0.3, 0.3)$	$(25.1, 16.5, 11.3)$	$(2,2),(2,3)$	$(1,3),(2,2)$	–
$(0.7, 0.2, 0.1)$	$(0, 65, 40)$	–	–	–

Within Table 16, as w_c increases and the combined weights for effectiveness measures decrease, fewer dual-purpose resources are used and, eventually, no resources are emplaced. These results are expected for this instance because the dual-purpose resources are more expensive and their location sites are clustered around the path capable of inflicting the most damage. Via the solutions to these different objective function weights, the tradeoff between cost and effectiveness is further evident, both between types of resources and whether to emplace resources at all.

3.3.3 Identifying the Limitations of a Commercial Solver for Global Optimization

It remains of interest to identify an appropriate solution methodology that efficiently finds high quality solutions. Given the WIPC is a non-convex, mixed-integer nonlinear math programming (MINLP) formulation, we test a leading commercial solver (i.e., BARON) designed for global optimization. We considered instances having combinations of sets of size $\mathcal{P} \in \{20, 40, 60, 80\}$ and, for the sake of simplicity, common parametric values of $\mathcal{J}/\mathcal{K}/\mathcal{L}$ in the set $\{5, 10, 15, 20, 25\}$. BARON was invoked to solve 10 stochastically generated instances for each parametric combination, each with alternative termination criteria of a 0% relative optimality gap and a time limit of 2700 seconds of computational effort.

Table 17 reports the average relative optimality gap (%) attained for the 10 instances at each parametric combination, as well as the number of instances for which a suboptimal solution was reported by BARON. Of note, the average relative optimality gap is strictly increasing for values of $\mathcal{J}/\mathcal{K}/\mathcal{L}$ for a given $\mathcal{P}$-value, and the same relationship holds for increasing values of $\mathcal{P}$ for a given $\mathcal{J}/\mathcal{K}/\mathcal{L}$-parameter. These general trends conform to intuition; solver performance degrades with an increasing size of instances of WIPC. (Roughly similar trends exist for the number of suboptimal solutions reported by BARON.) More interesting is that the degradation of the average relative optimality gap attained is greater with increases to the number of locations available for defense asset emplacement than the number of intruder paths.

Table 18 reports the average computational effort (seconds) required by BARON for the 10 instances at each parametric combination, as well as the number of instances for which BARON terminated due to the 2700 second limit on computational effort. Although there did exist two instances (i.e., at $(\mathcal{J}/\mathcal{K}/\mathcal{L}, \mathcal{P}) = (15, 20)$) for which BARON identified a global optimal solution upon termination at ~2700 sec-

Table 17. Average relative optimality gap (%) attained and number of instances (out of 10) for which a suboptimal solution was identified using commercial solver BARON for various instance sizes of WIPC

	Number of Intruder Paths ($\mathcal{P}$)			
$\mathcal{J}/\mathcal{K}/\mathcal{L}$	20	40	60	80
5	0.00 (0)	0.00 (0)	0.00 (0)	8.12 (1)
10	0.09 (1)	15.23 (3)	18.08 (3)	44.52 (7)
15	9.73 (2)	36.94 (5)	41.54 (5)	68.88 (8)
20	31.67 (4)	86.89 (10)	80.47 (9)	90.79 (10)
25	49.52 (6)	91.93 (10)	94.15 (10)	91.78 (10)

onds of computational effort, these results were anomalous; every suboptimal solution reported in Table 17 resulted from a termination of BARON due to the limit on computational effort, as indicated in Table 18. Therefore, it may be possible to improve the quality of solutions identified by BARON in Table 17, but doing so would be relatively inefficient.

Table 18. Average computational effort (seconds) required and the number of instances (out of 10) for which the commercial solver BARON terminated due to a 2700 second time limitation, for various instance sizes of WIPC

	Number of Intruder Paths ($\mathcal{P}$)			
$\mathcal{J}/\mathcal{K}/\mathcal{L}$	20	40	60	80
5	1.9 (0)	8.0 (0)	47.9 (0)	389.1 (1)
10	108.7 (1)	1207.9 (3)	1391.3 (3)	2113.2 (7)
15	1099.0 (4)	2115.0 (5)	2066.3 (5)	2409.9 (8)
20	1378.6 (4)	2702.0 (10)	2484.4 (9)	2700.4 (10)
25	2116.0 (7)	2705.6 (10)	2702.2 (10)	2701.7 (10)

Considering the collective testing results, a commercial solver designed for global optimization remains capable of identifying high quality solutions when considering a greater number of intruder paths, but its use to consider instances having a larger number of options for locating defensive assets is limited when the efficiency of a solution method is important. Moreover, the general trends observed portend yet greater challenges to solver efficiency with increased instance size(s). Such a limitation is challenging to accept for practical applications of the WIPC, motivating the exploration of a metaheuristic capable of efficiently addressing larger instances of the problem.

3.3.4 Metaheuristics as an Alternative to a Commercial Solver for Global Optimization

In this section, we show that GAs are a viable substitute for a commercial global solver for solving instances of WIPC. As discussed in Section 1.2, we use NSGA-II and RWGA as GAs for MOO, and BARON as the commercial solver used to solve the same randomly generated instance with 5-, 20-, and 45-minute run-time limits. For both GAs, the common parameters are population size ($n = 100$) and mutation probability during crossover ($p = 0.3$). RWGA has a specific parameter called N_{elite}, which is the number of previously discovered PO solutions that are reintroduced to the current population during each iteration. After completing pre-testing tuning, $N_{elite} = 5$ was chosen for this instance. All testing was completed on a 2.5 GHz with 16 GB of RAM and an Intel(R) Core(TM) i7-6500U processor. GAMS modeling language (Version 30.1.0) was used to invoke the commercial solver BARON (Version 19.12.7). To solve subproblems, BARON invoked IBM ILOG CPLEX (Version 12.10.0) and/or MINOS (Version 5.5), as appropriate. RWGA and NSGA-II were coded in RStudio (Version 3.3.2). BARON testing was completed by solving the randomly generated WIPC instance for nine weight combinations where $w_c \in \{0.1, 0.2, \ldots, 0.9\}$ and $w_e = 1 - w_c$, as an attempt to explore a wider range of solutions on the Pareto front. For each of the weight combinations, BARON was allowed the same run-time provided to RWGA and NSGA-II, wherein they could complete as many iterations as possible. That is, BARON solved each of the nine weight combinations with a 5-minute limit whereas NSGA-II and RWGA compiled and returned entire Pareto fronts within a single 5-minute limit on computational effort. This time-based termination criterion was imposed to ensure a more fair comparison for BARON, since it could be used to simultaneously solve multiple instances via parallel processing.

Figure 6 presents the PO solutions obtained by the three solution methods within

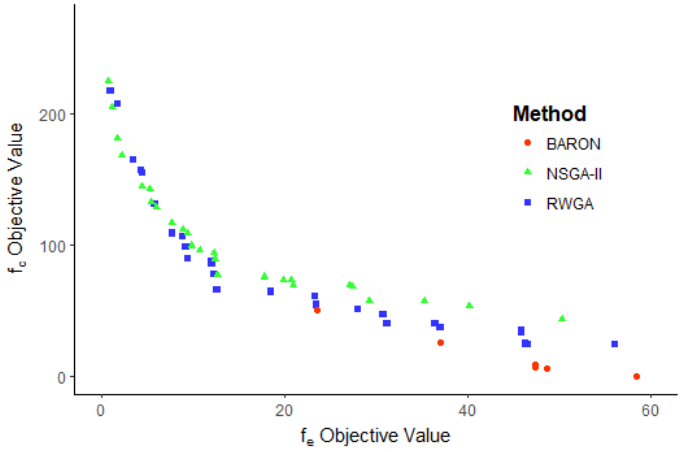

Figure 6. Pareto fronts generated by BARON, NSGA-II, and RWGA with a 5-minute run-time limit

Table 19. Comparison of Commercial Solver (BARON), NSGA-II, and RWGA regarding the solutions returned after 5 minutes of run-time

Solution Method	No. of Solutions Reported	No. of PO Solutions Reported	No. of PO Solutions Reported *Relative* to all Reported Solutions
BARON	9	6	6
NSGA-II	100	26	9
RWGA	27	27	21

the 5-minute run-time limit. Readily observable is that each of the three solution methods produce some non-dominated solutions (with respect to all solutions reported) at various regions along the front. For smaller values of f_e, NSGA-II produces PO solutions. For larger values of f_e, BARON produces PO solutions and, for values of f_e between 10 and 20, RWGA produces PO solutions. Table 19 reports the number of PO solutions identified by each method, both within each method's final set of solutions and with respect to the collective set of solutions identified by all three methods. The first column of Table 19 indicates the number of total solutions returned by each method. Note that NSGA-II returns the entire population upon termination, and only afterwards are the PO solutions identified; in contrast, RWGA reports the external population where only PO solutions are stored, which explains the differences (or lack thereof) between reported values in the first and second columns. The third column of Table 19 identifies the number of solutions identified by a given method that are non-dominated when compared to the collective set of solutions reported by all three methods. Clearly, RWGA reports more PO solutions relative to the other methods when the computational effort was limited to 5 minutes.

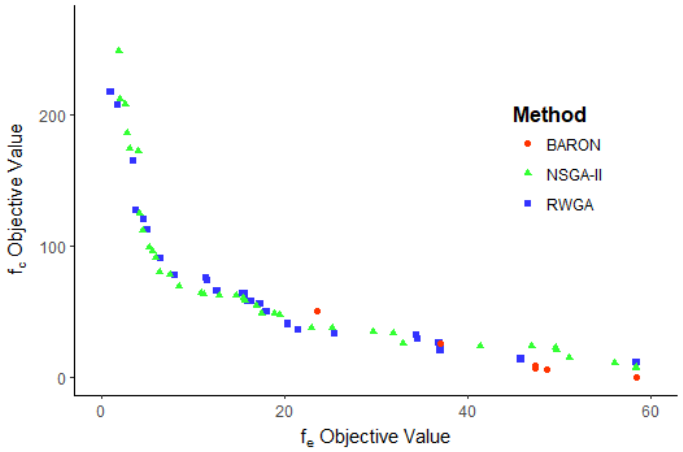

Figure 7. Pareto fronts generated by BARON, NSGA-II, and RWGA with a 20-minute run-time limit

Figure 7 displays the Pareto fronts generated by all three solution methods when

Table 20. Comparison of Commercial Solver (BARON), NSGA-II, and RWGA regarding the solutions returned after 20 minutes of run-time

Solution Method	No. of Solutions Reported	No. of PO Solutions Reported	No. of PO Solutions Reported *Relative* to all Reported Solutions
BARON	9	6	4
NSGA-II	100	38	22
RWGA	29	29	11

they are limited to a 20-minute run-time. The Pareto front generated by NSGA-II almost completely dominates the fronts generated by RWGA and BARON. Notably, the solutions reported by BARON are the exact same as those reported after 5 minutes, indicating that BARON's reported solutions did not improve with time, and they are on an extreme with respect to weights for f_e and f_c (i.e., the objective functions are not well scaled for the use of the Weighted Sum Method). Table 20 updates the results from Table 19 for the 20-minute time limit. Relative to the 5-minute results, NSGA-II improves the most, as evidenced by the number of PO solutions reported relative to the collective set of solutions identified by all three methods. Of the 38 solutions that NSGA-II reported after 20 minutes, 22 (57%) of them were still PO when compared to the solutions reported by RWGA and BARON.

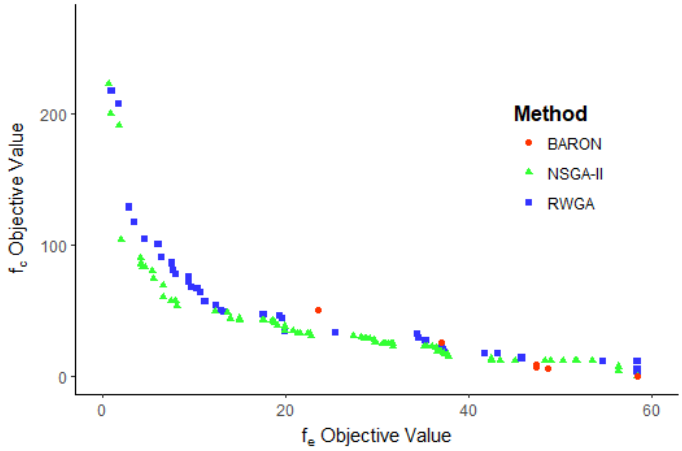

Figure 8. Pareto fronts generated by BARON, NSGA-II, and RWGA with a 45-minute run-time limit

Figure 8 displays the Pareto fronts generated by all three solutions methods when

Table 21. Comparison of Commercial Solver (BARON), NSGA-II, and RWGA regarding the solutions returned after 45 minutes of run-time

Solution Method	No. of Solutions Reported	No. of PO Solutions Reported	No. of PO Solutions Reported *Relative* to all Reported Solutions
BARON	9	6	4
NSGA-II	100	78	70
RWGA	41	41	4

using a 45-minute time limit. Similar to the results after 20 minutes, NSGA-II exhibits a strong Pareto front that dominates most of the solutions returned by RWGA as well as some solutions produced by BARON, but it does not dominate the BARON solutions produced for high values of w_e. Table 21 further demonstrates NSGA-II's dominance; 70 out of 78 solutions (i.e., 90%) produced by NSGA-II are still PO when compared to all solutions identified by the other two methods. Again, BARON produced the exact same nine solutions, of which only six were PO; as before, the solutions are not improving when allowing more time for solver convergence. Over all three time limits allowed, RWGA and NSGA-II returned a higher number and quality of solutions, but NSGA-II solutions dominated more RWGA solutions as the run-time limit increased.

Table 22. Comparison of convergence over time between BARON, NSGA-II, and RWGA

Method	No. of PO Solutions Reported After 5 min.	No. of PO Solutions Reported at 5 min. that are still present at 20 min.	No. of PO Solutions Reported at 20 min. that are still present at 45 min.
BARON	6	6	6
NSGA-II	26	0	0
RWGA	27	4	8

Table 22 compares the convergence of BARON, NSGA-II, and RWGA by examining the number of PO solutions that are returned with smaller run-time limits and the degree to which they "survive" to the next largest run-time limit. BARON did not evolve with time at all, as evidenced by the fact that the six PO solutions it returned

are identical after 5, 20, and 45 minutes of run-time. NSGA-II on the other hand, reported 26 PO solutions after 5 minutes of run-time, of which 0 were returned again after 20 minutes of run-time, indicating improvement in the set of solutions identified with greater computational effort. Likewise, none of the solutions NSGA-II reported after 45 minutes of run-time had been identified after 20 minutes. RWGA reported a similar number of solutions after 5 minutes and, while only four of them were still present at the 20-minute mark, eight of the solutions reported at 20 minutes were still present after 45 minutes had passed. This result indicates that NSGA-II exhibited the superlative convergence of solutions between the 20- and 45-minute time limits.

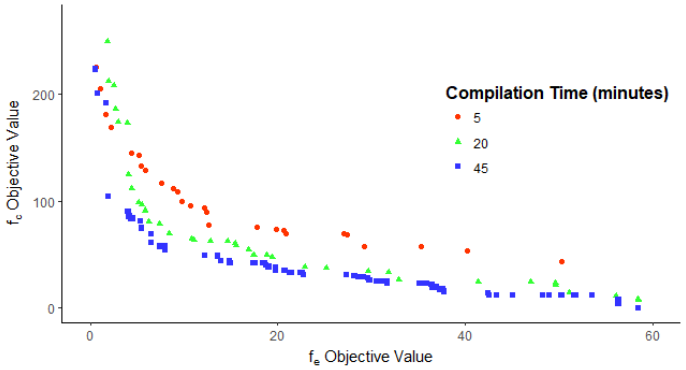

Figure 9. Pareto fronts generated by NSGA-II with 5-, 20-, and 45-minute run-time limits

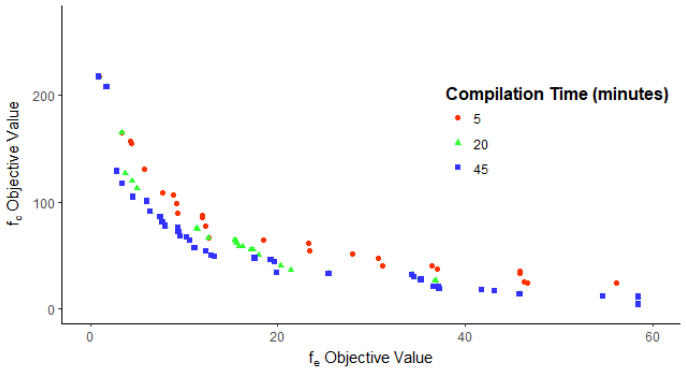

Figure 10. Pareto fronts generated by RWGA with 5-, 20-, and 45-minute run-time limits

Figures 9 and 10 display the Pareto fronts generated after 5, 20, and 45 minutes

by NSGA-II and RWGA, respectively. NSGA-II clearly improves with time and, combined with the results shown in Table 22, NSGA-II produced better Pareto fronts that dominated previously identified fronts. RWGA also returns better Pareto fronts as the run-time limit increases, but RWGA's improvement is not as stark as the improvement shown by NSGA-II. These results, combined with the data in Table 22, indicate that NSGA-II and RWGA both outperform BARON when solving this WIPC instance, with NSGA-II exhibiting a better performance than RWGA.

3.3.5 RWGA vs. NSGA-II as a Solution Method for Larger WIPC Instances

This section compares RWGA and NSGA-II as an extension of the testing reported in Table 17. Allowing the number of intruder paths to equal 25, 30, and 35, while also increasing the number of locations for each type of resource to 80, 100, and 120 allows for an examination of RWGA and NSGA-II for larger, more challenging instances of WIPC.

Table 23. Mean and Standard Deviation of PO solutions reported *relative* to all reported solutions for 10 instances of WIPC solved using RWGA and NSGA-II (2700-second time limitation)

		Number of Intruder Paths ($\mathcal{P}$)		
$\mathcal{J}/\mathcal{K}/\mathcal{L}$	**GA**	80	100	120
25	RWGA	9.8 ± 5.7	11.2 ± 7.7	11.4 ± 7.8
	NSGA-II	74.7 ± 9.3	66.1 ± 13.3	64.6 ± 9.6
30	RWGA	7.1 ± 3.8	8.3 ± 4.1	7.8 ± 2.5
	NSGA-II	69.7 ± 8.7	63.4 ± 11.1	49.5 ± 8.4
35	RWGA	11.9 ± 7.7	10.1 ± 6.0	10.4 ± 3.8
	NSGA-II	58 ± 13.1	46.3 ± 9.6	38.8 ± 6.7

Table 23 details the results of the direct comparison between RWGA and NSGA-II on nine different problem sizes. In each problem size, 10 random instances were randomly generated and solved by both RWGA and NSGA-II with a 2700-second run-time limit. Once the PO solutions for each metaheuristic were returned, the solutions were examined in aggregate for a given instance, and the number of solutions that

were still PO for each metaheuristic were recorded. The data in each cell is the mean ± s.d. of the number of solutions reported that are PO relative to all reported solutions across the 10 random instances. NSGA-II outperforms RWGA at every instance size. Via paired t-tests with $\alpha = 0.01$, NSGA-II reported a significantly larger number of PO solutions than RWGA, leading us to recommend it as a method for solving instances of the WIPC.

3.4 Conclusions

Given an intruder attempting to traverse a spatially-decomposed region via multiple possible paths, this research aims to effectively and cost-efficiently identify a defensive strategy that locates sets of detection resources and interdiction resources, each of which has different types of resources that vary by cost and capability. This research formulated and validated a mixed-integer nonlinear programming model formulation to solve the underlying problem using a leading commercial solver and two different metaheuristics as solution methods.

In comparing the solution methodologies, limitations for identifying a global optimal solution via a leading commercial solver (BARON) were identified during computational testing. Given a 2700-second limit for run-time, BARON was able to identify a feasible solution quickly but failed to identify an optimal solution for most test instances, especially as scenario parameters increased in size. The relative optimality gap achieved by BARON in these test instances was 91% in the largest-sized instance tested, which was not a particularly large instance. This result motivated the consideration of metaheuristics as an alternative solution method.

For the two multi-objective GAs, RWGA and NSGA-II, each of which was selected based on different conceptual performance characteristics, empirical testing demonstrated their superior performance in comparison with BARON, with respect to the

both the quantity of non-dominated solutions identified (individually or relative to all methods tested) and the required computational effort to do so. Subsequent testing of the two GAs over a designed set of test instances identified NSGA-II as the recommended technique to solve larger-sized instances of the underlying problem. Even as instance sizes increased, NSGA-II produced more non-dominated solutions relative to the solutions returned by RWGA at a statistically significant level.

A sequel to this research will examine this problem within a game theoretic context, wherein a rational intruder can observe the defender's asset location decisions prior to commencing an intrusion. The resulting extensive form game motivates the exploration of a bilevel programming model framework and a corresponding examination of solution methodologies to not only identify the intruder's optimal second-stage response(s) to a defender's asset location decisions, but to also identify the optimal first-stage decisions by the defender.

IV. Intruder Detection and Interdiction Modeling: A Bilevel Programming Approach for Ballistic Missile Defense Asset Location

4.1 Introduction

The rapid and recent proliferation of adversary missile threats induces a need for a similar evolution in missile defense of the United States (US) and it's North Atlantic Treaty Organization (NATO) partner, Canada. Potential threats arise from the Democratic People's Republic of Korea (DPRK) and Iran, two nation-states developing and expanding their intercontinental ballistic missile (ICBM) technology (United States Department of Defense, 2019). The developing capabilities augmented with the repeated threats to use them against the US (Heinrichs, 2020; Martin, 2021) has compelled the US to prioritize the development, procurement, and fielding of an every-increasingly complex missile defense enterprise (United States Department of Defense, 2019).

In 1944 during World War II (WWII), Germany fired the first long-range, guided ballistic missile, the V-2 rocket (Missile Defense Agency, 2013). The V-2 had a range of only 200 miles and was quite inaccurate compared to contemporary missile technology but, due to the lack of a defense in place for such a threat outside of bombing the launch sites, it still managed to significantly damage sites in Great Britain (Missile Defense Agency, 2013). After WWII, the US and the Union of Soviet Socialist Republics (USSR) engaged in the Cold War, during which the US and USSR simultaneously expanded their ballistic missile and ballistic missile defense (BMD) technology over the course of approximately 45 years. Presently, there are at least eight countries capable of launching ICBMs and are considered adversaries or potential adversaries of the US (United States Department of Defense, 2019). Among these countries, and particularly aggressive with missile development and threats to

use them, the DPRK has successfully tested its ICBM, the Hwasong-15, as recently as 2017, and it is thought to have the potential to reach the entire US (United States Department of Defense, 2019).

A ballistic missile's flight is decomposed into three phases of flight: boost, midcourse, and terminal. In the boost phase, the missile is being propelled into Earth's upper atmosphere by a rocket (National Research Council, 2008). Once there, the missile is in the midcourse phase and being powered by only gravity as it falls toward its target on Earth's surface and re-enters the atmosphere. The missile enters the terminal phase once it is back in Earth's atmosphere and makes its final approach to the target.

The current US BMD enterprise is comprised of detection and interdiction assets for stopping adversary missiles from reaching the US, and these existing assets are generally developed to address detection or interdiction of missiles in a particular phase of flight. Since the advent of radar in WWII, the US has developed radar technology such as the Sea-Based X-band Radar (SBX), an ocean-going semi-submersible platform equipped with an X-Band radar that can be positioned as needed in the ocean (Missile Defense Agency, 2018*a*). Modern missile defense leverages surface-based missiles in order to interdict BMs in flight. Examples of this include the Patriot missile defense system used in Operation Desert Storm, Iron Dome developed by Israel and used in the Gaza-Israel Conflict, and Ground-Based Interceptors currently in use by the US.

The proliferation of adversary missile technology has given rise to the development of new missile defense assets over time. The Missile Defense Agency (MDA) requested $9.1 billion for Fiscal Year 2021 to continue the development of new technology to aid in this defense (Missile Defense Agency, 2020). This budget marks an increase from $7.6 billion just five years ago (Missile Defense Agency, 2017*a*). Many assets have yet

to be fully developed and fielded, but research remains very active. For example, the Department of Defense has stated that the F-35 is slated to be equipped with technology that will aid in the attempt to intercept ballistic missiles in the boost phase, which is currently identified as a very difficult and expensive task (United States Department of Defense, 2019). Moreover, the MDA has allocated approximately $250 million to space-based sensors that will aid in the tracking of ballistic missiles in flight (Missile Defense Agency, 2019).

With both the current and evolving threats fielded by adversaries, the US BMD enterprise faces the daunting task of developing and integrating new detection and interdiction resources effectively. With the use of existing resources and the anticipation of new resources in coming years, an enterprise approach to the BMD problem is necessary and appropriate to ensure a cohesive response to adversary threats. Given this motivating problem, this research seeks to address the following problem statement:

> *Effectively defend a set of population centers against attack by a limited number of ICBMs by locating sets of BMD resources to detect and interdict ICBMs over a range of launch-to-target missile paths and their respective, spatio-temporally defined flight stages, assuming an adversary will observe the defensive asset location decisions and respond with an ICBM targeting strategy that maximizes the expected damage of an attack.*

4.1.1 Literature Review

This research is informed by four threads of research within the literature. The modeling herein leverages concepts and approaches from general missile defense studies, intruder detection and interdiction models, game-theoretic frameworks for defense problems, and defender-attacker models for missile defense from both the defender and intruder point of view. With respect to its solution methodologies, this research

applies bilevel program modeling techniques techniques as well as a metaheuristic approach to explore the solution space.

Given this work seeks to examine a ballistic missile defense problem, a brief examination of historical missile defense studies is warranted. Garwin and Bethe (1968) studied the "light" anti-ballistic missile (ABM) defense system that Defense Secretary McNamara authorized, arguing that it would prove ineffective against the current adversary missile technology. Johnson (1970) subsequently studied the current state of US radar technology and provided a detailed explanation of how US radar is used in the larger US BMD system. Following President Reagan's declaration that a space-based ballistic missile defense system would be a useful asset, Bethe et al. (1984) published work suggesting that such a system would prove ineffective and unlikely to protect the US from an adversarial nuclear attack. This early perspective on space-based interception of ballistic missiles differs greatly from current studies and opinions on space-based BMD, such as the most recent Missile Defense Review, which states that space-based BMD will be a necessary component of the future architecture providing boost-phase defense (United States Department of Defense, 2019). In a somewhat recent study, Wilkening (2000) provided a more concrete approach to examining probabilistic models, employing various adversary shooting philosophies and providing optimal interceptor allocations for each.

The second major thread of research that informs this work is intruder detection and interdiction. Lessin et al. (2018) developed a bilevel programming model to optimally allocate sensors to aid in the detection of an intruder. Similarly, Eliş et al. (2021) modeled the defense of a region of terrain using guards deployed such that each piece of the terrain is observed by at least one guard. Scheiper et al. (2019) solved an electric network design problem to support demand by electric vehicles. Although not an intruder detection and interdiction problem, the authors' work was its conceptual

dual; their research seeks to enable travel on a network rather than interdict it. Haywood et al. (2020) created a model that, once solved with a global solver, provides the defender with optimal location decisions for detection and interdiction resources in order to maximize the success of interdicting an intruder on a known path of intrusion. In an extension of this work, Haywood et al. (2021) solved a similar problem with multiple intruders on a set of paths with a known probability of use, employing a genetic algorithm to solve a more complex model. Components of these models are incorporated within the bilevel programming architecture of the current work.

The third thread of literature that lends insight to this work is game theory, specifically Stackelberg games. This is a game in which the amount won by one player is exactly equal to the amount lost of the other player (i.e., "zero-sum"), and players take turns making their moves (i.e., "extensive-form") (Shoham and Leyton-Brown, 2008). At each turn, a player is aware of what action other players took on the previous turn as well as the value of their current move in terms of how it affects both themselves and their opponent (i.e., "complete and perfect information") (Shoham and Leyton-Brown, 2008). This type of game-theoretic framework is especially applicable to BMD scenarios because the adversaries in the scenario are making location or launch decisions in turn, informed by observed adversary decisions. Additionally, the BMD scenario can be modeled appropriately as a zero-sum game with the assumption that the intruder and defender are maximizing and minimizing, respectively, the damage inflicted by intruder missiles.

The fourth thread of literature that is relevant to this research includes missile defense studies that employ a defender-attacker model. These types of studies utilize a framework in which a defender makes the first decision, usually allocating their resources in anticipation of an attack, and subsequently an intruder observes this decision and reacts accordingly when deciding their best course of action. An example of

this can be seen in work by Brown et al. (2006), wherein the authors applied defender-attacker and defender-attacker-defender modeling techniques to problems related to terrorists attempting to attack critical infrastructure. Brown et al. (2005) applied this modeling technique specifically to theater ballistic missile defense, and develop a decision-support tool for decision-makers to aid with the positioning of defense assets to prepare for missile attacks. The research presented herein differs from the work by Brown et al. (2005) in that an enterprise view is adopted; both detection and interdiction resources hosted on various platforms are considered. Boardman et al. (2017) present a defender-attacker-defender model for the location of surface-to-air missile batteries, wherein a defender first locates their batteries and an intruder observes these decisions and launches their missile attack. The defender then observes this attack and makes decisions regarding the assignment of interceptor missiles in batteries to incoming attacker missiles. Han et al. (2016) preceded Boardman et al. (2017) and studied the problem with homogeneous interceptor missiles. This research will not explicitly model the latter part of this problem, instead focusing on the initial defender decision of locating assets to minimize damage done by intruder missile attacks, assuming the capacity of an interdiction resource is not overwhelmed by the number of ballistic missiles encountered.

Within the context of the related literature, this research makes three contributions. First, this research sets forth a game theoretic, bilevel program modeling framework for the problem of allocating missile defense resources to detect and interdict intruder ballistic missiles attempting to destroy valuable targets. Second, it applies a series of transformations that reformulate the model as a single-level mathematical program that is shown to be convex and, hence, readily solvable to optimality by any of a number of commercial optimization solvers. Third, the research conducts testing to both illustrate its efficacy and empirically examine its practical tractabil-

ity, both of which are sound for application on large-scale instances of the underlying problem.

The remainder of this paper is organized as follows. Section 2 presents the mathematical programming formulation and solution methodology; Section 3 validates the model for an illustrative instance and conducts the aforementioned empirical testing; and Section 4 summarizes the resulting insights and identifies logical extensions to this research.

4.2 Models and Solution Methodology

First, we develop a bilevel programming model of the problem in which a single defender locates resources, after which an attacker observes the defender's actions and routes its missiles accordingly to inflict the most damage. That is, the defender aims to minimize the maximum amount of expected damage inflicted by attacker missiles. We assume the attacker and defender have good intelligence on each others' capabilities (i.e., complete information), and the attacker can observe defender locations and reasonably infer their allocation of resources (i.e., perfect information).

4.2.1 Bilevel Mathematical Programming Model

To formulate the mathematical program to address the underlying problem, it is necessary to define the following sets, parameters, and decision variables.

Sets

- $U = \{1, 2, ..., \mathcal{U}\}$ is the set of launch sites from which the attacker may launch missiles, indexed by u.
- $V = \{1, 2, ..., \mathcal{V}\}$ is the set of target sites to which the attacker may aim missiles, indexed by v.

- $P = \{(1,1),(1,2),\ldots,(\mathcal{U},\mathcal{V})\}$ is the set of paths over which the attacker may traverse through the defender's territory, indexed by ψ. The size of this set is denoted $\mathcal{U} \cdot \mathcal{V} = \mathcal{P}$.

- $S = \{1,2,...,\mathcal{S}\}$ is the set of distinguishable stages over which the attacker may be detected and interdicted by the defender's enterprise of sensors and interdictors, indexed by s. Relative to the set of stages, two assumptions are made regarding the attacker's path. First, we assume that each path transits every stage. Second, the stages are numbered in ascending order, as an attacker would encounter them when traversing any path. For the research application herein, $\mathcal{S} = 3$ to represent the boost, midcourse, and terminal stages of adversary missile flight, but we retain the parameter-based representation within the formulation to support its generalizability for other, related problems pertaining to attacker detection and interception.

- $D = \{1,2,...,\mathcal{D}\}$ is the set of different detection resource types, indexed by d, each of which pertains to different capabilities (e.g., range, effectiveness).

- $J = \{1,2,...,\mathcal{J}\}$ is the set of possible locations at which detection resources can be located, indexed by j. J is partitioned by stage, where $\bigcup_{s \in S} J_s = J$.

- $I = \{1,2,...,\mathcal{I}\}$ is the set of different interdiction resource types, indexed by i, each of which has different capabilities (e.g., speed, range, probability of success).

- $K = \{1,2,...,\mathcal{K}\}$ is the set of possible locations at which interdiction resources can be located, indexed by k. Similar to set J, the set K is likewise partitioned over S.

- $B = \{1,2,...,\mathcal{B}\}$ is the set of dual-purpose resource types (i.e., resources that

can both detect *and* interdict an attacker), indexed by b (or b'), each of which pertains to different capabilities (e.g., speed, range, probability of interdiction, probability of detection).

- $L = \{1, 2, ..., \mathcal{L}\}$ is the set of possible locations at which dual-purpose resources can be located, indexed by l (or l'). Similar to sets J and K, the set L is likewise partitioned over S.

Parameters

- $r_\psi > 0$: the expected damage that a missile on path ψ would inflict if not interdicted.

- $u_d^{\mathbb{D}}, u_i^{\mathbb{I}}, u_b^{\mathbb{B}}$: the maximum number of detection, interdiction, and dual-purpose resources that can be emplaced, respectively of types d, i, and b.

- m_s : the maximum number of interception engagements that can be attempted within each stage s.

- m_ψ: the maximum number of engagements of an intruder path ψ by a given interceptor.

- λ: an integer value equal to the maximum number of missiles launched by the attacker.

- $range_d^{\mathbb{D}}, range_b^{\mathbb{BD}}, range_i^{\mathbb{I}}, range_b^{\mathbb{BI}}$: the detection ranges for detection resources of type d and dual-purpose resources of type b, and interdiction ranges for interdiction resources of type i, and dual-purpose resources of type b, respectively.

- $a_{dj\psi s}^{\mathbb{D}}$: a binary parameter equal to 1 if the closest point in stage s on path ψ to location j is less than or equal to $range_d^{\mathbb{D}}$.

- $a^{\mathbb{BD}}_{bl\psi s}$: a binary parameter equal to 1 if the closest point in stage s on path ψ to location l is less than or equal to $range^{\mathbb{BD}}_{b}$.

- $p^{\mathbb{I}}_{ik\psi s}$: the probability that an attacker on path ψ is interdicted by an interdiction resource of type i emplaced at location k during flight stage s.

- $p^{\mathbb{BI}}_{bl\psi s}$: the probability that an attacker on path ψ is interdicted by a dual-purpose resource of type b emplaced at location l during flight stage s.

- $\gamma^{\mathbb{I}}_{ik\psi s}$: a binary parameter equal to 1 if the closest point in stage s on path ψ to location k is less than or equal to $range^{\mathbb{I}}_{i}$.

- $\gamma^{\mathbb{BI}}_{bl\psi s}$: a binary parameter equal to 1 if the closest point in stage s on path ψ to location l is less than or equal to $range^{\mathbb{BI}}_{b}$.

Decision Variables

- x_{dj} : a binary variable equal to 1 if a detection resource of type d is emplaced at location j, and 0 otherwise.

- y_{ik} : a binary variable equal to 1 if an interdiction resource of type i is emplaced at location k, and 0 otherwise.

- z_{bl} : a binary variable equal to 1 if a dual-purpose resource of type b is emplaced at location l, and 0 otherwise.

- δ_{ψ} : a binary variable equal to 1 if path ψ is used by the attacker and 0 otherwise.

- $\theta^{\mathbb{I}}_{ik\psi s}$: a non-negative integer variable equal to 1 if an interdiction resource of type i emplaced at location k is employed to engage an attacker on path ψ during stage s.

- $\theta^{\mathbb{BI}}_{bl\psi s}$: a non-negative integer variable equal to 1 if a dual-purpose resource of type b emplaced at location l is employed to engage an attacker on path ψ during stage s.

- $\pi^{\mathbb{I}}_{\psi}$: the probability of an attacker on path ψ being detected and subsequently interdicted.

Leveraging the aforementioned notation, we formulate the model **P1** as follows.

$$\min_{\substack{\mathbf{x},\mathbf{y},\mathbf{z},\\ \theta^{\mathbb{I}},\theta^{\mathbb{BI}},\pi^{\mathbb{I}}}} \max_{\delta} \sum_{\psi\in P} \delta_{\psi} r_{\psi}(1-\pi^{\mathbb{I}}_{\psi}) \tag{57}$$

$$\text{s.t. } (1-\pi^{\mathbb{I}}_{\psi}) = \prod_{s\in S}\left(\prod_{i\in I}\prod_{k\in K}\left(1-p^{\mathbb{I}}_{ik\psi s}\right)^{\theta^{\mathbb{I}}_{ik\psi s}}\prod_{b\in B}\prod_{l\in L}\left(1-p^{\mathbb{BI}}_{bl\psi s}\right)^{\theta^{\mathbb{BI}}_{bl\psi s}}\right),\ \forall\ \psi\in P \tag{58}$$

$$\theta^{\mathbb{I}}_{ik\psi s} \le m_{\psi}\gamma_{ik\psi s}\left(\sum_{d\in D}\sum_{j\in J} a^{\mathbb{D}}_{dj\psi s}x_{dj} + \sum_{b\in B}\sum_{l\in L} a^{\mathbb{BD}}_{bl\psi s}z_{bl}\right),\ \forall\ i\in I,\ k\in K, \psi\in P, s\in S, \tag{59}$$

$$\theta^{\mathbb{BI}}_{bl\psi s} \le m_{\psi}\gamma_{bl\psi s}\left(\sum_{d\in D}\sum_{j\in J} a^{\mathbb{D}}_{dj\psi s}x_{dj} + \sum_{b'\in B}\sum_{l'\in L} a^{\mathbb{BD}}_{b'l'\psi s}z_{b'l'}\right),\ \forall\ b\in B,\ l\in L, \psi\in P, s\in S, \tag{60}$$

$$\theta^{\mathbb{I}}_{ik\psi s} \le m_{\psi}y_{ik},\ \forall\ i\in I,\ k\in K, \psi\in P, s\in S, \tag{61}$$

$$\theta^{\mathbb{BI}}_{bl\psi s} \le m_{\psi}z_{bl},\ \forall\ b\in B,\ l\in L, \psi\in P, s\in S, \tag{62}$$

$$\sum_{i\in I}\sum_{k\in K}\theta^{\mathbb{I}}_{ik\psi s} + \sum_{b\in B}\sum_{l\in L}\theta^{\mathbb{BI}}_{bl\psi s} \le m_s,\ \forall\ \psi\in P,\ \forall\ s\in S, \tag{63}$$

$$\sum_{s\in S}\theta^{\mathbb{I}}_{ik\psi s} \le m_{\psi},\ \forall\ i\in I, k\in K, \psi\in P, \tag{64}$$

$$\sum_{s\in S}\theta^{\mathbb{BI}}_{bl\psi s} \le m_{\psi},\ \forall\ b\in B, l\in L, \psi\in P, \tag{65}$$

$$\sum_{j\in J} x_{dj} \le u^{\mathbb{D}}_{d},\ \forall\ d\in D, \tag{66}$$

$$\sum_{k \in K} y_{ik} \leq u_i^{\mathbb{I}}, \ \forall \ i \in I, \tag{67}$$

$$\sum_{l \in L} z_{bl} \leq u_b^{\mathbb{B}}, \ \forall \ b \in B, \tag{68}$$

$$\sum_{\psi \in P} \delta_\psi \leq \lambda, \tag{69}$$

$$0 \leq \pi_\psi^{\mathbb{I}} \leq 1, \ \forall \ \psi \in P, \tag{70}$$

$$x_{dj} \in \{0, 1\}, \ \forall \ d \in D, \ j \in J \tag{71}$$

$$y_{ik} \in \{0, 1\}, \ \forall \ i \in I, \ k \in K \tag{72}$$

$$z_{bl} \in \{0, 1\}, \ \forall \ b \in B, \ l \in L \tag{73}$$

$$\delta_\psi \in \{0, 1\}, \ \forall \ \psi \in P, \tag{74}$$

$$\theta_{ik\psi s}^{\mathbb{I}} \in \mathbb{Z}_+, \ \forall \ i \in I, \ k \in K, \psi \in P, s \in S, \tag{75}$$

$$\theta_{bl\psi s}^{\mathbb{BI}} \in \mathbb{Z}_+, \ \forall \ b \in B, \ l \in L, \psi \in P, s \in S, \tag{76}$$

The objective (57) of this formulation reflects the zero-sum nature of the game being played between attacker and defender. The attacker's objective is to maximize the expected damage done by their missiles and the defender's objective is to minimize the same expression. The order of operators in Equation (57) indicates the defender first making the decisions to locate detection and interdiction resources, assigning them to identified, potential missile paths (i.e., inferred from known missile launch sites and possible missile targets), after which the attacker observes the defender's actions and selects targets to inflict maximal cumulative expected damage. Constraint (58) calculates the probability an attacker missile is detected and subsequently interdicted by defender assets, with an underlying assumption that the probability of detection and subsequent interdiction between stages is independent. Additionally, we assume that, if a detection asset is within range of an attacker missile during a

given stage, then it is detected with certainty. Constraint (59) ensures interdiction resources can only be deployed by the defender if the attacker missile can be detected by an emplaced asset. Similarly, Constraint (60) ensures that dual-purpose interdiction resources can be deployed to intercept an attacker missile only if the attacker missile is being detected by defender resources. Constraints (61)-(62) are assignment constraints in which interdiction and dual-purpose assets are only assigned to be launched from a location if they have been placed at that location by the defender. That is, the defender may not employ interdiction resources from locations that they have not been placed. Constraint (63) places an upper bound on the engagements that can be made against each attacker per stage in intruder paths. For example, the defender may choose to limit the number of engagements in the ballistic stage of each attacker path to two. Constraints (64)-(65) place an upper bound on the number of engagements from each interdiction location to each path utilizing interdiction and dual-purpose resources, respectively. Constraints (66)-(68) limit the number of detection, interdiction, and dual-purpose resources that can be emplaced by type, respectively, and Constraint (69) limits the number of missiles that can be launched by the attacker. Finally, Constraint (70) places 0-1 bounds on the probability of detection and subsequent interdiction on each path by the defender, Constraints (71)-(74) detail the binary nature of defender location and attacker path decision variables, and Constraints (75)-(76) ensure the assignment variables for engagements are positive integers.

Noting the nonlinearity observed in Constraint (58), a logarithmic transformation is performed by introducing a new variable $\phi_{\psi}^{\mathbb{I}} = \ln\left(1 - \pi_{\psi}^{\mathbb{I}}\right)$. Constraint (78) will replace Constraint (58), and the transformed objective function is represented in Equation (77).

$$\min_{\substack{\mathbf{x},\mathbf{y},\mathbf{z},\\ \theta^{\mathbb{I}},\theta^{\mathbb{BI}},\phi^{\mathbb{I}}_{\psi}}} \max_{\delta} \quad \sum_{\psi\in P} \delta_{\psi} r_{\psi} e^{\phi^{\mathbb{I}}_{\psi}} \tag{77}$$

$$\phi^{\mathbb{I}}_{\psi} = \sum_{s\in S}\left(\sum_{i\in I}\sum_{k\in K(i,\psi,s)} \theta^{\mathbb{I}}_{ik\psi s}\ln\left(1-p^{\mathbb{I}}_{ik\psi s}\right) + \sum_{b\in B}\sum_{l\in L(b,\psi,s)} \theta^{\mathbb{BI}}_{bl\psi s}\ln\left(1-p^{\mathbb{BI}}_{bl\psi s}\right)\right), \quad \forall\ \psi\in P. \tag{78}$$

We denote the reformulated model as **P2** with the objective function as depicted in Equation (77), bounded by Constraints (59)-(76) and (78). The main advantage of P2 over P1 is the reformulated constraints are linear, yielding a polytope as its feasible region. We observe that P1 consists of bilinear terms (i.e., products of the attacker and defender decision variables), whereas P2 has defender variables in exponentiation. Section 4.2.2 further examines P2 and presents a solution method that allows the use of a commercial solver.

4.2.2 Solution Methodology

The special structure of the lower-level attacker problem can be exploited in a manner that allows for a reformulation of P2 into a single-level minimization problem. Within the bilevel structure of Problem P2, note that, for a fixed defender solution, the attacker is solving a knapsack problem with items having a value of $r_{\psi}e^{\phi_{\psi}}$ and equal costs. Thus, the binary restriction on the δ_{ψ}-variables may be relaxed and, for $\lambda \in \mathbb{Z}_{+}$, a binary-valued solution to the relaxed lower-level problem will yield the optimal objective function value. We denote **P3** as the model obtained by relaxing the binary constraints on δ_{ψ} (i.e., replacing Constraint (74) in P2 with Constraints (79) and (80)).

$$\delta_\psi \leq 1, \qquad \forall\ \psi \in P \tag{79}$$

$$\delta_\psi \geq 0, \qquad \forall\ \psi \in P \tag{80}$$

Proposition 1. *For a fixed upper-level (i.e., defender) solution to P3, a binary-valued optimal solution to the lower-level (i.e., attacker) problem exists.*

Proof. As seen in the objective (77) and Constraints (69), (79) and (80), the lower-level problem of P3 is a relaxation of a simple knapsack problem in which the value of each "knapsack" item (i.e., expected damage inflicted by each attacker missile) is $r_\psi e^{\phi^{\mathbb{I}}_\psi}$, and only λ items can be placed in the knapsack (i.e., fired at target sites). Therefore, for a cardinality weighted knapsack problem, an optimal solution uses a strict prioritization of the items based on their value, subject to total capacity. □

The optimal solution to P3, (z^*_{P3}), which exists by Proposition 1, is clearly an upper bound to the P2 solution (z^*_{P2}) since P3 is employing a relaxation on δ_ψ in the attacker problem. Due to the lower-level attacker problem consisting of a relaxation to a knapsack problem, the solution to P3 portends a solution to P2 in terms of δ_ψ consisting of a binary 0-1 vector. By the properties of a knapsack problem with cardinality weights on items, this binary valued solution is the attacker's best response to the defender's component of the optimal solution to P3.

The bilevel structure of P3 contains the relaxation of the attacker problem, which is simply a (integer-relaxed) knapsack problem for which the dual is quite useful. Adopting a technique employed by Wood (1993) for a bilevel program having the same objective function for both decision-makers (i.e., a zero-sum, extensive form game with perfect and complete information, in the game theoretic context), we can take the dual of the lower-level (i.e., inner) problem to yield a single-level optimization

problem. Noting the linear form of the inner problem in P3, we assign dual variables α and β_ψ to Constraints (69) and (79), respectively. For the zero-sum Stackelberg game represented in P3, we can take the dual of the inner problem (e.g., see Wood (1993); Lunday et al. (2010); González-Díaz et al. (2021)) and solve the resulting single-level optimization problem using a commercial solver.

The resulting formulation, denoted Problem **P4**, is represented in (81)–(85) and obtained by replacing the lower-level (attacker) problem in P3 with its dual.

$$\min_{\substack{\mathbf{x},\mathbf{y},\mathbf{z},\theta^{\mathbb{I}} \\ \theta^{\mathbb{B}\mathbb{I}},\phi^{\mathbb{I}}_{\psi},\alpha,\beta_\psi}} \quad \alpha\lambda + \sum_{\psi\in P}\beta_\psi \tag{81}$$

$$\text{s.t.} \quad \alpha + \beta_\psi \geq r_\psi e^{\phi^{\mathbb{I}}_{\psi}},\ \forall\ \psi \in P, \tag{82}$$

$$\alpha \geq 0, \tag{83}$$

$$\beta_\psi \geq 0,\ \forall\ \psi \in P, \tag{84}$$

$$\text{Constraints (59)-(68), (70)-(76), and (78).} \tag{85}$$

The only nonlinear constraint in P4 is Constraint (82). We propose that Constraint (82) is, in fact, convex, and therefore P4 is a convex program (i.e., it has a linear (convex) objective function with a convex feasible region). To recall, Bazaraa et al. (2013) define a convex program as one in which the goal is to minimize a function $g(\boldsymbol{x})$ such that $\boldsymbol{x} \in S$, where g is a convex function and S is a convex set. The structure of convex programs is enticing because a local minimum of the program is also a global minimum, so an interior point algorithm will converge to a global optimal solution.

Proposition 2. *P4 is a convex program.*

Proof. Clearly, the linear objective (81) is a convex function. Constraints (82)-(85) induce the feasible region of P4, and all but Constraint (82) are linear and, hence,

convex. A brief analysis identifies that Constraint (82) is also convex.

Considering Constraint (82) in canonical form (e.g., see Bazaraa et al. (2013)), where $g(\boldsymbol{x}) = r_\psi e^{\phi^{\mathbb{I}}_\psi} - \alpha - \beta_\psi \leq 0$ and $\boldsymbol{x} = \left[\phi^{\mathbb{I}}_\psi, \alpha, \beta_\psi\right]$, it has a Hessian of the form

$$\boldsymbol{H}(\boldsymbol{x}) = \begin{bmatrix} r_\psi e^{\phi^{\mathbb{I}}_\psi} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The only non-zero-valued eigenvector of $\boldsymbol{H}(\boldsymbol{x})$ is $r_\psi e^{\phi^{\mathbb{I}}_\psi}$, which is strictly positive because $r_\psi > 0$ by construction and $e^{\phi^{\mathbb{I}}_\psi} > 0$. Thus, $\boldsymbol{H}(\boldsymbol{x})$ is positive semi-definite, Constraint (82) is convex, and the feasible region defined by the intersection of convex sets is also convex (Bazaraa et al., 2013). □

Resulting from Proposition (2), one can solve the single-level optimization problem P4 and find a global optimum utilizing an interior point method common to any of a number of alternative, readily-available commercial solvers designed for solving mixed-integer nonlinear programs for which the integer relaxation is a convex program. The resulting optimal solution to P4 also solves P3, where $z^*_{P4} = z^*_{P3}$, and from which the attacker's solution to P3 can be recovered from the optimal solution to P4 via the optimal dual variable values corresponding to Constraint (82).

4.3 Testing, Results, and Analysis

Before examining the limitations of a commercial solver, Section 4.3.1 tests the solution method detailed in Section 4.2.2 model for a small, illustrative instance. Section 4.3.2 details the method used to generate random instances for testing in Sections 4.3.3 and 4.3.4. Section 4.3.3 reports the results for a realistically-sized instance, and Section 4.3.4 examines the tractability of a commercial solver to address

larger instances of the underlying problem, under selective parametric increases in problem size.

4.3.1 Illustrative Test Instance

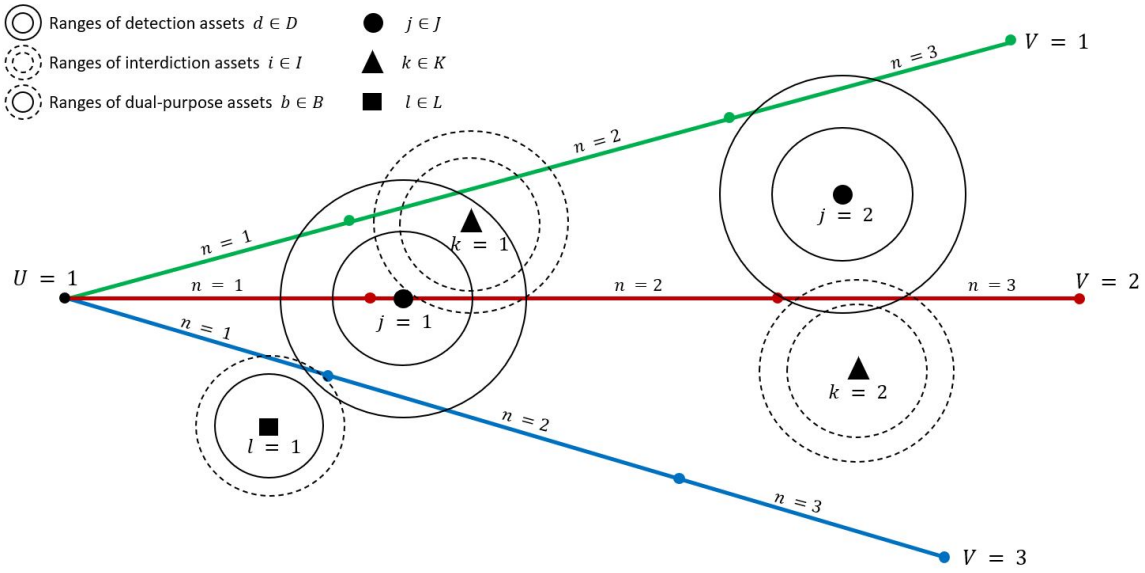

Figure 11. Small Illustrative Instance

For an illustrative test instance designed to validate the model and accompanying solution methodology, we employ a scenario of $(\mathcal{U}, \mathcal{V}, \mathcal{P}, \lambda) = (1, 3, 4, 2)$, $(\mathcal{S}) = (3)$, and $(\mathcal{D}, \mathcal{J}, \mathcal{I}, \mathcal{K}, \mathcal{B}, \mathcal{L},) = (2, 2, 2, 2, 1, 1)$. Paths 1-3 correspond to terminal points $v = 1, 2, 3$ with r_ψ values equal to 20, 30, and 40, respectfully. We allow $u_d^{\mathbb{D}} = u_i^{\mathbb{I}} = u_b^{\mathbb{B}} = 1$ for all asset types, and set $m_s = m_\psi = 2$. That is, only one type of each asset can be placed in total, and at most two interception engagements are allowed per stage and per path, respectively. Within the set of two detection assets, let the one having the larger index be the asset with the larger range (e.g., detection asset 2 has a larger range than asset 1). This characteristic also holds for the sets of two interdiction assets. Figure 11 depicts the locations $\mathcal{J}$, $\mathcal{K}$, and $\mathcal{B}$ at which these assets can be placed. Within this figure, dashed circles represent interdiction asset ranges, and solid circles represent detection asset ranges. Asset locations, ranges,

and proximity to each path inform the parameters $a^{\mathbb{D}}_{dj\psi s}$, $a^{\mathbb{BD}}_{bl\psi s}$, $\gamma^{\mathbb{I}}_{ik\psi s}$, and $\gamma^{\mathbb{BI}}_{bl\psi s}$ for the instance depicted in Figure 11. If a detection resource of type d located at position j is able to detect path ψ in stage s, then $a^{\mathbb{D}}_{dj\psi s} = 1$, indicating a detection asset of type d located at j is able to detect an intruder traversing stage s of path ψ with a probability equal to one. For example, $a^{\mathbb{D}}_{2121} = 1$ in the instance in Figure 11. Additionally, the probabilities of interdiction are set to $p^{\mathbb{I}}_{i1\psi s} = 0.5$, $p^{\mathbb{I}}_{i2\psi s} = 0.3$, and $p^{\mathbb{BI}}_{bl\psi s} = 0.2$. For this illustrative example, $\lambda = 2$ so the attacker is forced to choose a path on which no missile is deployed.

The optimal objective function value for this small instance is equal to 33.1. Within the unique optimal solution, the defender chooses to locate respective detection and interdiction assets of type 2 at locations $j = 1$ and $k = 1$ (i.e., $x^*_{21} = y^*_{21} = 1$). The defender also chooses to locate its only dual-purpose asset at the only possible location, $l = 1$ (i.e., $z^*_{11} = 1$). Alternative locations for detection assets were not utilized because they would not detect any path. For similar reasons, it is not beneficial to locate an interdiction asset at alternative locations. Given this optimal solution, the defender is able to shoot twice at stage 2 of both paths 1 and 2 from location $k = 1$ (i.e., $\theta^{\mathbb{I}}_{2112} = \theta^{\mathbb{I}}_{2122} = 2$), and to shoot twice at stage 1 of path 3 (i.e., $\theta^{\mathbb{BI}}_{1131} = 2$). As a result, the vector of missile interdiction probabilities on paths 1, 2, and 3 is $\left(\boldsymbol{\pi}^{\mathbb{I}}\right)^* = (0.75, 0.75, 0.36)$. For this problem instance, the attacker's optimal solution vector is $\boldsymbol{\delta}^* = (0, 1, 1)$, indicating decisions to deploy a missile along each of paths 2 and 3, where the r_ψ values equal 30 and 40, respectively. In doing so, the attacker's missile attack decisions maximize the objective function value in Equation (57). This illustrative instance validates that model P4 operates as expected and identifies globally minimizing asset location and assignment decisions for instances of the underlying problem. In a confirmatory experiment, the same instance was solved for Problem P3 via the commercial solver BARON, attaining the same result, and

a visual examination of alternative solutions for Problems P1 and P2 confirm the optimality of the attained solution.

4.3.2 Test Instance Generation

To test an instance of P4, we first define a "scenario" of P4 to pertain to the set of 3 user-defined tuples of $(\mathcal{U}, \mathcal{V}, \mathcal{P}, \lambda)$, $(\mathcal{S})$, and $(\mathcal{D}, \mathcal{J}, \mathcal{I}, \mathcal{K}, \mathcal{B}, \mathcal{L},)$. Once a scenario "size" has been determined by the aforementioned tuple, an instance is defined by parameterizing the scenario (i.e., generating and assigning values to $r_\psi, u_d^{\mathbb{D}}, \ldots, \gamma_{bl\psi s}^{\mathbb{BI}}$), dependent on scenario size as an input for some parameters.

For all instances examined herein, the common region in which the defender can place assets is a rectangular region 2300 units in width and 6400 units in height, roughly mimicking the expanse of the northern Pacific Ocean. The attacker's launch and target regions are rectangular regions 100 units in width and 6400 units in height, without loss of generality, appended to the respective western and eastern ends of the region wherein the defender can place assets. The values of r_ψ are randomly chosen from a set of 20 values equal to the populations of the largest 20 cities on the North American West Coast. For example, in the case where $|\mathcal{P}| = 8$, eight numbers are randomly chosen with replacement from the set of city populations to be used as r_ψ-values.

Haywood et al. (2021) detail a formula for generating values for $u_d^{\mathbb{D}}$, $u_i^{\mathbb{I}}$, and $u_b^{\mathbb{B}}$, wherein the values are identified from a random uniform distribution having lower and upper bounds dependent on the number of stages, locations available for asset placement, and the index of the asset type. We emulate the authors' convention, assuming the larger indices correspond to more effective assets (e.g., $u_2^{\mathbb{D}}$ has a larger

detection range than $u_1^{\mathbb{D}}$). These range formulas are shown in Equations (86)-(89).

$$range_d^{\mathbb{D}} = d \cdot U\,(0, 0.5) \cdot \frac{6400}{\mathcal{S}} \tag{86}$$

$$range_b^{\mathbb{BD}} = b \cdot U\,(0, 0.4) \cdot \frac{6400}{\mathcal{S}} \tag{87}$$

$$range_i^{\mathbb{I}} = i \cdot U\,(0, 0.5) \cdot \frac{6400}{\mathcal{S}} \tag{88}$$

$$range_b^{\mathbb{BI}} = b \cdot U\,(0, 0.5) \cdot \frac{6400}{\mathcal{S}} \tag{89}$$

We assume that, if a detection or dual-purpose asset is within range of a stage of a path, the probability of *detecting* an intruder missile during that stage of the path is equal to one. However, if an interdiction or dual-purpose asset is within range of a particular stage of a path, Equations (90) and (91) display the formula used to determine the probability of *interdiction* by those assets on the path during that stage.

$$p_{ik\psi s}^{\mathbb{I}} = \frac{i}{\mathcal{I}} \cdot U\,(0.5, 0.9) \tag{90}$$

$$p_{bl\psi s}^{\mathbb{BI}} = \frac{b}{\mathcal{B}} \cdot U\,(0.5, 0.9) \tag{91}$$

Finally, m_s and m_ψ are generated as functions of the number of stages and paths, respectively, as seen in Equations (92) and (93).

$$m_s = \lceil U(0.15, 0.25) \cdot \mathcal{S} \rceil \tag{92}$$

$$m_\psi = \lceil U(0.01, 0.1) \cdot \mathcal{P} \rceil \tag{93}$$

Using this procedure to develop a random instance for any scenario size of P4, Section 4.3.3 examines the solution to a larger-sized instance of P4, and Section 4.3.4 will examine the effect of scenario features on the ability of a solver to reach a high-

quality solution within a given run-time limit.

4.3.3 Illustration of Relevant Analysis and Insights

Herein, we demonstrate for a larger-sized instance the resulting insights that can be garnered when the details of an instance are too numerous and varied to enable a detailed visualization. The specific scenario size examined has features $(\mathcal{U}, \mathcal{V}, \mathcal{P}, \lambda) = (5, 5, 25, 10)$, $\mathcal{S} = 3$, and $(\mathcal{D}, \mathcal{J}, \mathcal{I}, \mathcal{K}, \mathcal{B}, \mathcal{L},) = (5, 10, 5, 10, 5, 10)$.

Table 24. Solution metrics for a larger-sized instance of P4, sorted in decreasing order according to expected damage per path

		Engagements per stage					
ψ	Target Value (r_ψ)	$s = 1$	$s = 2$	$s = 3$	$1 - \pi_\psi^I$	Expected Damage	δ_ψ
13	4	1	1	1	0.012	0.049	1
3	1.4	1	1	1	0.010	0.015	1
23	1.4	1	1	1	0.008	0.012	1
20	1.4	1	1	1	0.008	0.011	1
24	0.43	1	1	1	0.026	0.011	1
16	0.22	1	1	1	0.045	0.010	1
6	0.88	1	1	1	0.011	0.010	1
9	0.88	1	1	1	0.010	0.009	1
11	1	1	1	1	0.007	0.007	1
25	0.27	1	1	1	0.025	0.007	1
22	0.23	1	0	1	0.030	0.007	0
18	0.29	1	1	1	0.021	0.006	0
15	0.31	1	1	1	0.018	0.006	0
4	0.22	0	1	1	0.025	0.005	0
21	0.27	1	1	1	0.020	0.005	0
8	0.52	1	1	1	0.009	0.005	0
19	0.38	1	0	1	0.012	0.005	0
1	0.23	1	1	1	0.018	0.004	0
14	0.35	1	1	1	0.011	0.004	0
10	0.28	1	1	1	0.013	0.004	0
17	0.5	1	1	1	0.006	0.003	0
2	0.31	1	1	1	0.009	0.003	0
12	0.23	1	1	1	0.012	0.003	0
7	0.47	1	1	1	0.005	0.003	0
5	0.38	1	1	1	0.002	0.001	0

Table 24 depicts selected, relevant solution metrics for a single random instance

generated with the aforementioned scenario feature levels. The first column contains the respective indices of 25 paths, the second column displays the randomly generated target value for each path. The third, fourth, and fifth columns tabulate the number of defender engagements in each stage due to location decisions regarding detection and interdiction assets, and the sixth column contains the probability the attacker successfully navigates each path. The seventh column calculates the expected damage caused by the intruder by multiplying the values in the second and sixth columns, and the eighth column reports the attacker's decision whether to utilize the path. Table 24 is sorted by the seventh column in decreasing order, and confirms that the attacker will deploy its 10 missiles along the 10 paths having the highest expected damage.

4.3.4 Main Testing

The scenario feature levels examined for their effect on the computational effort required by a leading commercial solver to identify an optimal solution are the number of each type of asset allowed, the number of locations assets can be placed, the number of launch and target sites, the total number of stages, and the number of intruder missiles targeting defender sites. Table 25 depicts the low, medium, and high levels for each of these scenario features, each of which was identified during preliminary testing. Due to their similar meaning and to reduce the size of experimental designs to manageable sizes, testing herein assumes common factor levels for both the number of possible locations for asset placement (i.e., $\mathcal{J} = \mathcal{K} = \mathcal{L}$) and the number of asset types ($\mathcal{D} = \mathcal{I} = \mathcal{B}$).

Prior to conducting the intended empirical testing, preliminary testing examined four commercial solvers (i.e., BARON, Bonmin, Couenne, and scip) when solving a set of 30 randomly generated instances of P4 having medium-sized scenario features, with

Table 25. Scenario Feature Levels for Instances of P4

Scenario Feature	Low	Medium	High
$\mathcal{D} = \mathcal{I} = \mathcal{B}$	2	5	8
$\mathcal{J} = \mathcal{K} = \mathcal{L}$	5	10	15
$(\mathcal{U}, \mathcal{V})$	$(3, 5)$	$(5, 5)$	$(5, 7)$
$\mathcal{S}$	2	3	4
λ	5	10	15

the goal of identifying the superlative commercial solver for this problem. Testing was conducted using the NEOS Server (Gropp and Moré, 1997; Czyzyk et al., 1998; Dolan, 2001), and processed on a Dell PowerEdge R430 with an Intel Xeon E5-2698 processor and 192 GB of RAM. Each solver was terminated for an instance when either a global optimal solution was identified (i.e., a 0% relative optimal gap) or the computational time exceeded 300 seconds. Table 26 reports the number of optimal solutions found by each solver in the first column, the number of instances not solved to optimality in the second column, the average absolute optimality gap for the instances in which suboptimal solutions were reached in the third column, the number of instances in which the solver failed to return a feasible solution in the fourth column, and the average run-time for the 30 instances in the fifth column.

Table 26. Solver performance for 30 random instances of P4 with medium-sized scenario features and run-time limit of 300 seconds

Solver invoked	No. optimal solns.	No. suboptimal solns.	Avg. abs. opt. gap for suboptimal solns.	No. instances w/ no feasible soln. found	Avg. comp. effort (s)
BARON	20	8	0.035 ± 0.084	2	256.627 ± 57.407
Bonmin	0	0	N/A	30	300.000 ± 0.000
Couenne	0	27	1.217 ± 1.667	3	300.000 ± 0.000
scip	27	1	0.001 ± 0.000	2	57.600 ± 68.075

Over the 30 instances solved, scip found a feasible solution in 28 instances, 27 of which were optimal. In the sole instance that scip failed to find the optimal solution, an extremely small absolute optimality gap of 0.001 was still achieved. BARON was the second best performer to scip; however, BARON had a longer run-time on

average and failed to find a feasible solution for two instances. The other two solvers, while capable and respected in the literature, failed quite often to either identify a feasible solution or, if a feasible solution was identified, solve the instance to optimality within the 300-second time limit. Accordingly, subsequent testing invokes scip as the commercial solver to solve instances of Problem P4.

To determine which scenario features in Table 25 are most influential to the computational effort required by scip to find an optimal solution, a fractional factorial experiment is conducted. A full factorial design was not chosen due to the prohibitive size of the experiment requiring $3^5 = 243$ runs to complete. The fractional factorial experimental design employed is 3^{5-2}_{III} with 30 trials of each run, with the required computational effort as the response variable. A fractional factorial design is a reasonable design for this research, because the scenario features are the variables of interest, not the interactions between these variables. The 30 trials at each level are randomly generated instances using the methodology described in Section 4.3.2. The solver scip was terminated for an instance when either a global optimal solution was identified (i.e., a 0% relative optimal gap) or the computational time exceeded 1800 seconds, allowing for the longer times needed to solve instances at runs having high feature levels. Table 27 reports the results of this experiment to solve 30 randomly generated instances of P4 at each of 27 different treatment combinations of scenario feature levels, tabulating the average required computational effort for each run. For runs wherein scip terminated due to the 30-minute time limit for at least one of the 30 instances, Table 3 also reports the absolute optimality gap attained upon termination.

In 15 out of 27 (56%) of treatment combinations, scip found an optimal solution for each instance of P4 within the instance's allotted 30-minute time limit. In the other 12 treatment combinations, the average absolute optimality gap achieved by scip was less than or equal to 0.001, indicating that scip identified very high quality

Table 27. Solver performance for a 3^{5-2}_{III} fractional factorial design with 30 random instances of P4 at each setting and 1800-second run-time limit

Run	Factors $\mathcal{D}/\mathcal{I}/\mathcal{B}$	$\mathcal{J}/\mathcal{K}/\mathcal{L}$	$\mathcal{P}$	$\mathcal{S}$	λ	Req'd Comp. Effort (sec)	Abs. Optimality Gap Attained*
1	8	15	35	4	15	0.294 ± 0.176	–
2	2	5	15	2	5	5.232 ± 3.527	–
3	2	10	35	2	15	8.685 ± 5.940	–
4	2	15	25	2	10	11.313 ± 7.736	–
5	5	5	35	2	10	27.728 ± 13.788	–
6	5	10	25	2	5	15.015 ± 5.113	–
7	5	15	15	2	15	12.676 ± 5.609	–
8	8	5	25	2	15	28.232 ± 13.096	–
9	8	10	15	2	10	1,238.189 ± 513.792	0.001 ± 0.003
10	8	15	35	2	5	0.814 ± 0.646	–
11	2	5	25	3	5	1.910 ± 1.583	0.000 ± 0.000
12	2	10	15	3	15	87.584 ± 112.629	–
13	2	15	35	3	10	3.149 ± 2.216	0.000 ± 0.000
14	5	5	15	3	10	348.490 ± 437.747	0.000 ± 0.000
15	5	10	35	3	5	819.954 ± 683.335	0.000 ± 0.000
16	5	15	25	3	15	258.860 ± 320.917	0.000 ± 0.000
17	8	5	35	3	15	960.703 ± 646.639	0.000 ± 0.000
18	8	10	25	3	10	838.435 ± 726.696	0.000 ± 0.000
19	8	15	15	3	5	0.526 ± 0.421	–
20	2	5	35	4	5	1.898 ± 2.068	–
21	2	10	25	4	15	2.124 ± 1.703	0.000 ± 0.000
22	2	15	15	4	10	1.130 ± 0.663	–
23	5	5	25	4	10	3.957 ± 3.513	–
24	5	10	15	4	5	276.019 ± 455.152	0.000 ± 0.000
25	5	15	35	4	15	0.822 ± 0.514	0.000 ± 0.000
26	8	5	15	4	15	97.116 ± 142.193	–
27	8	10	35	4	10	192.375 ± 333.794	0.000 ± 0.000

*An entry of '–' indicates scip identified a global optimal solution for all 30 instances of a factor-level run for problem P4

solutions even in the instances wherein a global optimal solution was not positively identified.

For the required computational effort response, a simple linear regression (SLR) model is computed to further observe the effects of each scenario feature level on the response. Table 28 presents the coefficient estimates for SLR model using the data presented in Table 27.

Table 28. Standard Least Squares Regression Coefficient Estimates for Required Computational Effort (seconds)

Term	**Estimate**	**Std Error**	t **Ratio**	**Prob**$> |t|$
$\mathcal{D}/\mathcal{I}/\mathcal{B}$	65.154	5.467	11.920	<.0001
$\mathcal{J}/\mathcal{K}/\mathcal{L}$	35.431	3.280	10.800	<.0001
$\mathcal{P}$	7.941	1.640	4.840	<.0001
$\mathcal{S}$	-42.855	16.402	-2.610	0.0091
λ	-13.982	3.280	-4.260	<.0001

For a significance level of $\alpha = 0.05$, the fifth column in Table 28 indicates that each of the scenario feature levels is significant for predicting the computational effort required by scip when solving instances of P4. The second column presents an interesting result: the number of types of assets, locations at which to place assets, and paths available for the intruder to employ correlate positively with the amount of time required by scip to obtain optimal solutions. This result comports with conventional wisdom; having more options for a problem instance requires greater computational effort to identify an optimal solution. The number of stages over which the attacker may be detected and interdicted by the defender's assets and the number of intruder missiles the intruder uses correlate negatively with the response, indicating that more stages and/or intruder missiles result in a lesser amount of time required for scip to obtain optimal solutions for instances of P4. Collectively, these results indicate that solving instances of P4 having more locations for assets and asset types may induce computational challenges, but those challenges may be offset with an artificial par-

titioning of paths into a greater number of stages, should it be acceptable from a modeling perspective for the application of interest.

4.4 Conclusions

Given three respective sets of detection, interdiction, and dual-purpose resources, each having different types of resources with heterogenous capabilities, this research develops a mathematical programming model to effectively defend a set of population centers against attack by a limited number of ICBMs by locating sets of BMD resources to detect and interdict ICBMs over a range of launch-to-target missile paths. These paths, and their respective, spatio-temporally defined flight stages, are employed under the assumption the adversary will observe the asset locations and respond with a ICBM targeting strategy that maximizes the expected damage of an attack. We set forth a mixed-integer nonlinear program (MINLP) and develop subsequent models to linearize and reformulate the model to a single-objective mixed-integer linear program (MILP), referred to as P4.

After defining P4, we examine a small, illustrative instance of P4 to ensure that the model provides the expected optimal solution. We subsequently examine a larger-sized instance of P4 to demonstrate the readily discernible insights from an optimal solution for an instance that is too large and complex to analyze by inspection. The larger-sized instance of P4 confirms that the attacker will attack paths with the highest expected damage inflicted upon defender targets, implying that P4 has potential for use in much larger-sized instances.

After defining scenario features, this work details a method to parameterize instances of P4 in a manner dependent on scenario feature levels. For low, medium, and high scenario feature levels, testing compares four leading commercial solvers on their performance in obtaining solutions for 30 instances having medium-level sce-

nario features. The solver scip prevailed as the most consistent solver at finding high quality solutions and also in a timely manner, and was used as the solver for testing afterwards.

A fractional factorial design tested scip's ability to obtain optimal solutions in a given time limit for various scenario sizes. Using this data, an SLS regression model informed an examination of which scenario features are most significant in predicting the amount of time required by scip to obtain an optimal solution. All five scenario features examined are significant; three of the features correlate positively with the computational effort required, and the other two correspond negatively. Results identify the problem features that will induce computational challenges, even when solving the MINLP with the superlatively performing commercial solver.

Future research should examine the underlying problem in a similar game-theoretic context, with the caveat that the defender has incomplete information about the attacker's capabilities. Such an extension may identify asset location solutions that are suboptimal to the model formulated herein, but robust to missing or incorrect information about an adversary. An additional extension of merit would be to consider an attacker having different missile types with heterogenous capabilities and susceptibilities to interdiction.

V. Conclusions and Recommendations

This dissertation considers the problem of an intruder attempting to traverse a defender's territory, divided into distinct stages either spatially, temporally, or both, and wherein the defender aims to locate and employ disparate sets of resources to lower the probability of a successful intrusion. Various optimization techniques are used to model the problem, including mixed-integer nonlinear programming, multi-objective optimization, bilevel programming, and mixed-integer linear programming. Additionally, multiple solution techniques are examined such as leading commercial solvers for global optimization and multiple-objective optimization genetic algorithms. In addition to the models developed, many differently-sized test instances of the underlying problem are generated and solved to test the efficacy and efficiency of the respective solution methodologies. The research presented in this dissertation is of interest to planners in scenarios like the ballistic missile defense enterprise, wherein multiple resources are located and employed to minimize the expected damage inflicted by intruder missiles.

5.1 Conclusions

Given two respective sets of detection resources and interdiction resources, each having different types of resources with heterogeneous capabilities, Chapter II addresses the problem of locating and allocating them over a sequence of spatially-defined stages to effectively detect and intercept an intruder. A mixed-integer nonlinear program, and several variants, are constructed to address the underlying problem using a leading commercial solver for global optimization. Analysis identifies which factors in the Resource Allocation for Intruder Detection and Interdiction (RAIDI) scenarios influence the solution quality found by the model variants. Testing results

identified that the number of types of detection and interdiction resources are the significant factors in determining the relative optimality gap obtained by the model variants, and that every feature level is a significant factor in determining the computational effort required to solve an instance of a RAIDI scenario. The superlative RAIDI model variant identified via two phases of empirical testing is the *default-b* model, which augments the default model with simple upper and lower bounds on each of the probability calculations to enhance the performance of the commercial solver's branch-and-bound procedure.

The research presented in Chapter II makes three contributions to the literature. First, it sets forth a baseline mathematical programming model – and seven alternative variants – to address the underlying problem of allocating limited resources for the detection and interdiction of an intruder. Second, it conducts empirical testing to evaluate and compare the effect of alternative model variants on the efficacy and efficiency of a leading commercial solver to identify optimal solutions. Third, it rigorously examines the impact of selected problem features on the ability of a leading commercial solver to address larger instances of the underlying problem, portending its utility for larger applications.

Given an intruder attempting to traverse a spatially-decomposed region via multiple possible paths, Chapter III aims to effectively and cost-efficiently identify a defensive strategy that locates sets of detection resources and interdiction resources, each of which has different types of resources that vary by cost and capability. In comparing different solution methodologies, limitations for identifying a global optimal solution via a leading commercial solver (BARON) were identified during computational testing, which motivated the exploration of metaheuristics as a valid solution method. For the two multi-objective GAs chosen, RWGA and NSGA-II, each of which was selected based on different conceptual performance characteristics, empir-

ical testing demonstrated their superior performance in comparison with BARON, with respect to the both the quantity of non-dominated solutions identified (individually or relative to all methods tested) and the required computational effort to do so.

Within the context of the related literature, this research presented in Chapter III makes two contributions. In its first contribution to address the underlying problem, this research sets forth a mathematical programming model having several collectively complicating aspects that differentiate it from other research in the literature, as reviewed in Section 3.1.1. The model addresses the location of assets across an enterprise comprised of different asset types (i.e., detection and interdiction assets) and capabilities, including dual-purpose assets representing actual assets for certain motivating scenarios (e.g., AEGIS class destroyers in a BMD scenario). The enterprise approach of the model considers the location of these assets in a defender's territory organized into multiple stages, better representing the geographic boundaries often used to organize defenses for related applications (e.g., border patrol). Finally, the model employs a multi-objective approach to enable the examination of the tradeoffs between the effectiveness and cost of defensive asset configurations. In its second contribution, this research identifies and empirically tests alternative, conceptually sound solution methodologies for instances of the underlying problem. Empirical testing first identifies the instance size-specific limitations of a leading commercial, global optimization solver, motivating the examination of metaheuristics. Subsequent testing compares the relative efficacy of two metaheuristics for solving larger-sized instances, identifying the superlative technique that provides practical utility to the relevant mathematical programming model presented in the first contribution.

Given three respective sets of detection, interdiction, and dual-purpose resources, each having different types of resources with heterogenous capabilities, Chapter IV

develops a mathematical programming model to effectively defend a set of population centers against attack by a limited number of ICBMs by locating sets of BMD resources to detect and interdict ICBMs over a range of launch-to-target missile paths. These paths and their respective, spatio-temporally defined flight stages, are employed under the assumption the adversary will observe the asset locations and respond with a ICBM targeting strategy that maximizes the expected damage of an attack. We set forth a mixed-integer nonlinear program (MINLP), and develop subsequent models to linearize and reformulate the model to a single-objective mixed-integer linear program (MILP), referred to as P4. P4 is validated using a small, illustrative instance and subsequently tested on an instance too large and complex to analyze by inspection which confirms that the attacker will act as expected by attacking paths with the highest expected damage inflicted upon defender targets. Subsequent analysis uses a fractional factorial design, which determined that every scenario feature for instances of P4 are significant in predicting the amount of time the solver (scip) needs to obtain an optimal solution and that certain scenario features induce significant computational challenges.

Within the context of the related literature, the research in Chapter IV makes three contributions. First, IV sets forth a game theoretic, bilevel program modeling framework for the problem of allocating missile defense resources to detect and interdict intruder ballistic missiles attempting to destroy valuable targets. Second, it applies a series of transformations that reformulate the model as a single-level mathematical program that is shown to be convex and, hence, readily solvable to optimality by any of a number of commercial optimization solvers. Third, the research conducts testing to both illustrate its efficacy and empirically examine its practical tractability, both of which are sound for application on large-scale instances of the underlying problem.

5.2 Recommendations

Future research may examine a similar problem to those studied in Chapters II-IV, albeit having certain differences not considered herein due to anticipated tractability issues. Although these issues could be insurmountable, only a deliberate effort will identify the challenges in detail and, perhaps, overcome them.

One possible avenue to explore is the removal of the assumed independence between probability of detection and interdiction within a stage. Although revisiting this assumption is not helpful to the defender in terms of efficiently identifying a solution, it may lend more fidelity to the underlying problem. However, removing this assumption leads to various modeling issues that are not simple to remedy, and it will almost certainly require the design and use of a metaheuristic to find solutions in a reasonable amount of time.

Another problem worthy of exploration is the situation wherein an intruder has incomplete information. For example, if the intruder cannot observe the defender's location decisions with 100% accuracy (and the defender is aware of this shortcoming), how might the resulting solutions change? A similar problem may arise if the defender is given incomplete information about the intruder's options regarding an attack (e.g., unknown number of paths or attacker target values).

Finally, future research may also examine adding more objectives to a problem similar to the one observed in Chapter IV. Multi-objective optimization in combination with a bilevel programming formulation may yield an interesting study, particularly when examining potential solution methods. Chapter III portends the use of multi-objective genetic algorithms as a potential solution method for instances of the underlying problem, and its contributions may prove useful in this research extension.

Bibliography

Arms Control Association (2019), 'Chronology of U.S.-North Korean nuclear and missile diplomacy', `https://www.armscontrol.org/factsheets/dprkchron#2019`. Accessed on 4 November 2019.

Basciftci, B., Ahmed, S. and Shen, S. (2021), 'Distributionally robust facility location problem under decision-dependent stochastic demand', *European Journal of Operational Research* **292**(2), 548–561.

Bazaraa, M. S., Sherali, H. D. and Shetty, C. M. (2013), *Nonlinear programming: theory and algorithms*, John Wiley & Sons.

Bell, J. E., Griffis, S. E., Cunningham III, W. A. and Eberlan, J. A. (2011), 'Location optimization of strategic alert sites for homeland defense', *Omega* **39**(2), 151–158.

Berman, O. and Krass, D. (2002), 'The generalized maximal covering location problem', *Computers & Operations Research* **29**(6), 563–581.

Bethe, H. A., Garwin, R. L., Gottfried, K. and Kendall, H. W. (1984), 'Space-based ballistic-missile defense', *Scientific American* **251**(4), 39–49.

Boardman, N. T., Lunday, B. J. and Robbins, M. J. (2017), 'Heterogeneous surface-to-air missile defense battery location: a game theoretic approach', *Journal of Heuristics* **23**(6), 417–447.

Borrero, J. S., Prokopyev, O. A. and Sauré, D. (2016), 'Sequential shortest path interdiction with incomplete information', *Decision Analysis* **13**(1), 68–98.

Brown, G., Carlyle, M., Diehl, D., Kline, J. and Wood, K. (2005), 'A two-sided optimization for theater ballistic missile defense', *Operations Research* **53**(5), 745–763.

Brown, G., Carlyle, M., Salmerón, J. and Wood, K. (2006), 'Defending critical infrastructure', *Interfaces* **36**(6), 530–544.

Capar, I., Kuby, M., Leon, V. J. and Tsai, Y.-J. (2013), 'An arc cover–path-cover formulation and strategic analysis of alternative-fuel station locations', *European Journal of Operational Research* **227**(1), 142–151.

Church, R. L. and Murray, A. (2018), *Location Covering Models*, Springer, Cham, Switzerland.

Church, R. L. and ReVelle, C. S. (1976), 'Theoretical and computational links between the p-median, location set-covering, and the maximal covering location problem', *Geographical Analysis* **8**(4), 406–415.

Church, R. and ReVelle, C. (1974), The maximal covering location problem, *in* 'Papers of the Regional Science Association', Vol. 32, Springer-Verlag, pp. 101–118.

Cormican, K. J., Morton, D. P. and Wood, R. K. (1998), 'Stochastic network interdiction', *Operations Research* **46**(2), 184–197.

Czyzyk, J., Mesnier, M. P. and Moré, J. J. (1998), 'The NEOS server', *"IEEE Journal on Computational Science and Engineering"* **5**(3), 68–75.

Daskin, M. S. (1983), 'A maximum expected covering location model: formulation, properties and heuristic solution', *Transportation Science* **17**(1), 48–70.

Daskin, M. S. (2011), *Network and discrete location: models, algorithms, and applications*, John Wiley & Sons, Hoboken, New Jersey.

de Grey, A. (2005), 'Whole-body interdiction of lengthening of telomeres: a proposal for cancer prevention', *Front Biosci* **10**, 2420–2429.

Deb, K. (2001), *Multi-objective optimization using evolutionary algorithms*, Vol. 16, John Wiley & Sons.

Deb, K. (2014), Multi-objective optimization, *in* 'Search methodologies', Springer, pp. 403–449.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197.

Dolan, E. D. (2001), The NEOS server 4.0 administrative guide, Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory.

Drake, J. H., Starkey, A., Owusu, G. and Burke, E. K. (2020), 'Multiobjective evolutionary algorithms for strategic deployment of resources in operational units', *European Journal of Operational Research* **282**(2), 729–740.

Drezner, Z. and Hamacher, H. W. (2001), *Facility location: applications and theory*, Springer Science & Business Media, Berlin, Germany.

Ehrgott, M. (2005), *Multicriteria Optimization*, Vol. 491, Springer Science & Business Media, Berlin, Germany.

Eliş, H., Tansel, B., Oğuz, O., Güney, M. and Kian, R. (2021), 'On guarding real terrains: The terrain guarding and the blocking path problems', *Omega* **102**, 102303.

Garwin, R. L. and Bethe, H. A. (1968), 'Anti-ballistic-missile systems', *Scientific American* **218**(3), 21–31.

González-Díaz, J., González-Rodríguez, B., Leal, M. and Puerto, J. (2021), 'Global optimization for bilevel portfolio design: Economic insights from the Dow Jones Index', *Omega* **102**, 102353.

Gonzalez, S. R., Jalali, H. and Van Nieuwenhuyse, I. (2020), 'A multiobjective stochastic simulation optimization algorithm', *European Journal of Operational Research* **284**(1), 212–226.

Gourley, S. R. (2011), 'Soldier Armed: PAC-3 MSE update', *Army Magazine* **61**(7), 65–66.

Gropp, W. and Moré, J. J. (1997), Optimization environments and the NEOS server, *in* M. D. Buhman and A. Iserles, eds, 'Approximation Theory and Optimization', Cambridge University Press, Cambridge, United Kingdom, pp. 167 – 182.

Hakimi, S. L. (1964), 'Optimum locations of switching centers and the absolute centers and medians of a graph', *Operations Research* **12**(3), 450–459.

Hakimi, S. L. (1965), 'Optimum distribution of switching centers in a communication network and some related graph theoretic problems', *Operations Research* **13**(3), 462–475.

Han, C. Y., Lunday, B. J. and Robbins, M. J. (2016), 'A game theoretic model for the optimal location of integrated air defense system missile batteries', *INFORMS Journal on Computing* **28**(3), 405–416.

Hausken, K. (2010), 'Tactical identification of wide intruders', *Military Operations Research* **15**, 51–60.

Haywood, A. B., Lunday, B. J., Robbins, M. J. and Pachter, M. N. (2021), 'The weighted intruder path covering problem', *European Journal of Operational Research* .

Haywood, A., Lunday, B., Robbins, M. and Pachter, M. (2020), Enterprise resource location-allocation for intruder detection and interdiction, Technical report, Air Force Institute of Technology, Department of Operational Sciences.

Heinrichs, R. L. (2020), 'Biden must prioritize missile defense - defense one', `https://www.defenseone.com/ideas/2020/12/biden-must-prioritize-missile-defense/170944/`. (Accessed on 01/17/2021).

Holland, J. H., Holland, J. H. et al. (1975), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, MI.

Hu, X., Zhu, W., Ma, H., An, B., Zhi, Y. and Wu, Y. (2021), 'Orientational variable-length strip covering problem: A branch-and-price-based algorithm', *European Journal of Operational Research* **289**(1), 254–269.

Huang, S.-J. (2000), 'An immune-based optimization method to capacitor placement in a radial distribution system', *IEEE Transactions on Power Delivery* **15**(2), 744–749.

Israeli, E. and Wood, R. K. (2002), 'Shortest-path network interdiction', *Networks: An International Journal* **40**(2), 97–111.

Johnson, C. M. (1970), 'Ballistic-missile defense radars', *IEEE Spectrum* **7**(3), 32–41.

Karabulut, E., Aras, N. and Altınel, İ. K. (2017), 'Optimal sensor deployment to increase the security of the maximal breach path in border surveillance', *European Journal of Operational Research* **259**(1), 19–36.

Karasakal, O. and Karasakal, E. K. (2004), 'A maximal covering location model in the presence of partial coverage', *Computers & Operations Research* **31**(9), 1515–1526.

Konak, A., Coit, D. W. and Smith, A. E. (2006), 'Multi-objective optimization using genetic algorithms: A tutorial', *Reliability Engineering & System Safety* **91**(9), 992–1007.

Laporte, G., Nickel, S. and da Gama, F. S. (2015), *Location science*, Vol. 528, Springer, Cham, Switzerland.

Larter, David B. (2018), 'The US Navy is fed up with ballistic missile defense patrols', `https://www.defensenews.com/naval/2018/06/16/the-us-navy-is-fed-up-with-ballistic-missile-defense-patrols/`. Accessed on 4 November 2019.

Lessin, A. M., Lunday, B. J. and Hill, R. R. (2018), 'A bilevel exposure-oriented sensor location problem for border security', *Computers & Operations Research* **98**, 56–68.

Lessin, A. M., Lunday, B. J. and Hill, R. R. (2019), 'A multi-objective, bilevel sensor relocation problem for border security', *IISE Transactions* **51**(10), 1091–1109.

Lunday, B. J., Sherali, H. D. and Glickman, T. S. (2010), 'The nested event tree model with application to combating terrorism', *INFORMS Journal on Computing* **22**(4), 620–634.

Mahecic, A. (2020), 'UNHCR seeks support for refugees and hosts in Ethiopia', `https://www.unhcr.org/news/briefing/2020/1/5e2ab8ec4/unhcr-seeks-support-refugees-hosts-ethiopia.html`. Accessed on 11 February 2020.

Marler, R. T. and Arora, J. S. (2004), 'Survey of multi-objective optimization methods for engineering', *Structural and multidisciplinary optimization* **26**(6), 369–395.

Martin, T. W. (2021), 'North Korea's missiles and nuclear weapons: Everything you need to know - WSJ', https://www.wsj.com/articles/north-koreas-missiles-and-nuclear-weapons-everything-you-need-to-know-11610712018. (Accessed on 01/17/2021).

Matlin, S. (1970), 'A review of the literature on the missile-allocation problem', *Operations Research* **18**(2), 334–373.

Missile Defense Agency (2013), *Missile Defense - The First Seventy Years*, Washington, DC.

Missile Defense Agency (2016*a*), 'Cobra Dane fact sheet', https://www.mda.mil/global/documents/pdf/cobradane.pdf. Accessed on 11 November 2019.

Missile Defense Agency (2016*b*), 'Upgraded Early Warning Radar fact sheet', https://www.mda.mil/global/documents/pdf/uewr1.pdf. Accessed on 11 November 2019.

Missile Defense Agency (2017*a*), 'MDA historical funding', https://www.mda.mil/global/documents/pdf/FY17_histfunds.pdf. Retrieved on 15 December 2020.

Missile Defense Agency (2017*b*), 'Space Tracking and Surveillance System fact sheet', https://www.mda.mil/global/documents/pdf/stss.pdf. Accessed on 11 November 2019.

Missile Defense Agency (2018*a*), 'Sea-Based X-Band Radar fact sheet', https://www.mda.mil/global/documents/pdf/sbx.pdf. Accessed on 11 November 2019.

Missile Defense Agency (2018*b*), 'THAAD fact sheet', https://www.mda.mil/global/documents/pdf/thaad.pdf. Accessed on 4 November 2019.

Missile Defense Agency (2019), 'Congress backs off push for space-based missile intercept layer', https://www.defensenews.com/congress/2019/12/11/congress-backs-off-push-for-space-based-missile-intercept-layer/. (Accessed on 12/16/2020).

Missile Defense Agency (2020), 'Budget estimates overview: FY2021', https://www.mda.mil/global/documents/pdf/budgetfy21.pdf. Retrieved on 15 December 2020.

Moghaddam, M. and Nof, S. Y. (2014), Location-allocation decisions in collaborative networks of service enterprises, *in* 'IIE Annual Conference. Proceedings', Institute of Industrial and Systems Engineers (IISE), p. 4141.

Monk, E. and Wagner, B. (2012), *Concepts in enterprise resource planning*, Cengage Learning, Boston, Massachusetts.

Morecroft, J. D. (1983), 'A systems perspective on material requirements planning', *Decision Sciences* **14**(1), 1–18.

Morton, D. P., Pan, F. and Saeger, K. J. (2007), 'Models for nuclear smuggling interdiction', *IIE Transactions* **39**(1), 3–14.

Murata, T. and Ishibuchi, H. (1995), MOGA: Multi-objective Genetic Algorithms, *in* 'IEEE International Conference on Evolutionary Computation', Vol. 1, pp. 289–294.

Musman, S., Lehner, P. and Elsaesser, C. (1997), 'Sensor planning for elusive targets', *Mathematical and Computer Modelling* **25**(3), 103–115.

Nandi, A. K. and Medal, H. R. (2016), 'Methods for removing links in a network to minimize the spread of infections', *Computers & Operations Research* **69**, 10–24.

Nandi, A. K., Medal, H. R. and Vadlamani, S. (2016), 'Interdicting attack graphs to protect organizations from cyber attacks: A bi-level defender–attacker model', *Computers & Operations Research* **75**, 118–131.

National Research Council (2008), *U.S. Conventional Prompt Global Strike: Issues for 2008 and Beyond*, The National Academies Press, Washington, DC. Accessed on 4 November 2019.

O'Connell, K. and Cafasso, J. (2018), 'Septicemia: Causes, symptoms, and complications', `https://www.healthline.com/health/septicemia`. Accessed on 11 February 2020.

Paul, N. R., Lunday, B. J. and Nurre, S. G. (2017), 'A multiobjective, maximal conditional covering location problem applied to the relocation of hierarchical emergency response facilities', *Omega* **66**, 147–158.

Rabbani, M., Heidari, R. and Yazdanparast, R. (2019), 'A stochastic multi-period industrial hazardous waste location-routing problem: Integrating NSGA-II and Monte Carlo simulation', *European Journal of Operational Research* **272**(3), 945–961.

Ryoo, H. S. and Sahinidis, N. V. (1995), 'Global optimization of nonconvex nlps and minlps with applications in process design', *Computers & Chemical Engineering* **19**(5), 551–566.

Ryoo, H. S. and Sahinidis, N. V. (1996), 'A branch-and-reduce approach to global optimization', *Journal of Global Optimization* **8**(2), 107–138.

Sahinidis, N. and Tawarmalani, M. (2004), 'BARON: The GAMS solver manual', *GAMS Development Corporation* pp. 9–20.

Sahinidis, N. V. (1996), 'BARON: A general purpose global optimization software package', *Journal of Global Optimization* **8**(2), 201–205.

Scheiper, B., Schiffer, M. and Walther, G. (2019), 'The flow refueling location problem with load flow control', *Omega* **83**, 50–69.

Schlesinger, J. and Solomon, R. (2020), 'E-skimming cyberattack is growing along with online shopping', `https://www.cnbc.com/2020/01/31/e-skimming-cyberattack-is-growing-along-with-online-shopping.html`. Accessed on 11 February 2020.

Serafino, P. and Ventre, C. (2016), 'Heterogeneous facility location without money', *Theoretical Computer Science* **636**, 27–46.

Shehab, E., Sharp, M., Supramaniam, L. and Spedding, T. A. (2004), 'Enterprise resource planning', *Business Process Management Journal* **10**(4), 359–386.

Shoham, Y. and Leyton-Brown, K. (2008), *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*, Cambridge University Press, Cambridge, United Kingdom.

Speier, R. H., Nacouzi, G., Lee, C. and Moore, R. M. (2017), *Hypersonic missile nonproliferation: hindering the spread of a new class of weapons*, Rand Corporation, Santa Monica, CA.

Srinivas, N. and Deb, K. (1994), 'Muiltiobjective optimization using nondominated sorting in genetic algorithms', *Evolutionary Computation* **2**(3), 221–248.

Talbi, E.-G., Basseur, M., Nebro, A. J. and Alba, E. (2012), 'Multi-objective optimization using metaheuristics: non-standard algorithms', *International Transactions in Operational Research* **19**(1-2), 283–305.

Tawarmalani, M. and Sahinidis, N. V. (2004), 'Global optimization of mixed-integer nonlinear programs: A theoretical and computational study', *Mathematical Programming* **99**(3), 563–591.

Tawarmalani, M. and Sahinidis, N. V. (2005), 'A polyhedral branch-and-cut approach to global optimization', *Mathematical Programming* **103**(2), 225–249.

Thompson, L. (2020), 'Why the Pentagon's THAAD missile defense system is becoming critical to protection of the U.S. homeland', `https://www.forbes.com/sites/lorenthompson/2020/03/23/why-the-pentagons-thaad-missile-defense-system-is-becoming-critical-to-protection-of-the-us-homeland/#27e47e7d55a6`. Accessed on 3 April 2020.

Umble, E. J., Haft, R. R. and Umble, M. M. (2003), 'Enterprise resource planning: Implementation procedures and critical success factors', *European Journal of Operational Research* **146**(2), 241–257.

United States Department of Defense (2019), *Missile Defense Review*, Washington, DC.

United States Joint Chiefs of Staff (2017), *Joint Publication 3-01: Countering Air and Missile Threats*, Washington, DC.

Upchurch, C., Kuby, M. and Lim, S. (2009), 'A model for location of capacitated alternative-fuel stations', *Geographical Analysis* **41**(1), 85–106.

US Navy (2019), 'Aegis Weapon System', `https://www.navy.mil/navydata/fact_display.asp?cid=2100&tid=200&ct=2`. Accessed on 11 November 2019.

Wilkening, D. A. (2000), 'A simple model for calculating ballistic missile defense effectiveness', *Science & Global Security* **8**(2), 183–215.

Wood, R. K. (1993), 'Deterministic network interdiction', *Mathematical and Computer Modelling* **17**(2), 1–18.

Wood, R. K. (2010), 'Bilevel network interdiction models: Formulations and solutions', *Network* **174**, 175.

Yates, J. (2013), Network interdiction methods and approximations in a HAZMAT transportation setting, *in* 'Handbook of OR/MS Models in Hazardous Materials Transportation', Springer, Cham, Switzerland, pp. 187–243.

Zhang, C. and Ramirez-Marquez, J. E. (2013), 'Protecting critical infrastructures against intentional attacks: A two-stage game with incomplete information', *IIE Transactions* **45**(3), 244–258.

Zheng, J. and Castañón, D. A. (2012), Dynamic network interdiction games with imperfect information and deception, *in* '2012 IEEE 51st IEEE Conference on Decision and Control (CDC)', pp. 7758–7763.

Zitzler, E., Deb, K. and Thiele, L. (2000), 'Comparison of multiobjective evolutionary algorithms: Empirical results', *Evolutionary Computation* **8**(2), 173–195.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
7–06–2021	Doctoral Dissertation	October 2018 — July 2021

4. TITLE AND SUBTITLE

Enterprise Resource Allocation for Intruder Detection and Interception

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Haywood, Adam B.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-DS-21-S-043

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Strategic Development Planning and Experimentation (SDPE)
Mr. David Panson
Air Force Research Laboratory
WPAFB, OH 45433
david.panson@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION A
APPROVAL FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This dissertation considers the problem of an intruder attempting to traverse a defender's territory in which the defender locates and employs disparate sets of resources to lower the probability of a successful intrusion. This research is conducted in the form of three related research components. The first component examines the problem in which the defender subdivides their territory into spatial stages and knows the plan of intrusion. The second component studies a similar problem but is unaware of the intrusion plan, introduces more defensive assets capable of lowering the probability of a successful intrusion, and examines alternative solution methods for instances of the problem. The third component further studies the underlying problem by using a game-theoretic framework in which the attacker observes defender location decisions prior to formulating an appropriate intrusion plan.

15. SUBJECT TERMS

resource location and allocation, missile defense, nonlinear programming, multi-objective genetic algorithm, game theory

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Brian J. Lunday, AFIT/ENS
U	U	U	UU	137	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636, x4624; brian.lunday@afit.edu