

Risk-Aware Decision Support for Critical Infrastructure Protection using Multi-Objective Optimization

By: **Nicolas Primeau**

Thesis submitted in partial fulfillment
of the requirements for the degree of

**Master of Applied Science in
Electrical and Computer Engineering**

The Ottawa-Carleton Institute for
Electrical and Computer Engineering
School of Electrical Engineering and Computer Science
University of Ottawa

October 2017

© Nicolas Primeau, Ottawa, Canada, 2017

Abstract

The world of today is increasingly dependant on a functional, globalized economy. The defence and security establishments' reliance on supplies and logistics is not new. First responders rely on many tools and systems that are critical to their endeavours. Somewhat disjoint at first glance, these domains share a common need for complex physical or logistical infrastructures such as power plants, ports, supply chains, to name a few examples. All of these are potentially vulnerable to attacks, disruptions, breakdowns, or other activities that disable the infrastructure and consequently cause important physical or economic damage. An obligation exists to protect these critical infrastructures and a decision support system that is able to detect, identify, and mitigate the risk of unwanted events would be invaluable in preventing the disastrous consequences of compromised infrastructure. This thesis explores the design and application of such a system. It starts with a pre-existing, actively researched risk management framework and proposes a methodology to apply it in new contexts, as well as contributions to provide the framework with the ability to solve new problems. Relevant case studies in critical infrastructure protection are presented, as well as applications of the developed methodology with the proposed modifications when suitable. Simulations, results, and insightful discussions are provided for each of the case studies. Finally, research trends, future work, and a conclusion are given, completing this thesis.

Acknowledgments

This thesis would not have been possible if not for the efforts of

- Dr. Petriu, for funding this research and the many expensive conferences
- Dr. Rami Abielmona and Dr. Rafael Falcon, for providing consistent guidance and valuable reviews throughout the last two years
- To all the good folk at Larus, who are always ready to lend a hand
- To my family, who supported me unconditionally,
- To Emmanuelle Goulet, for always listening and for enduring me,
- And to my friends, who must be wondering where I was these past few years.

Contents

Abstract	ii
Acknowledgments	iii
1 Introduction	1
1.1 Motivations	3
1.2 Contributions	4
1.3 Thesis Organization	6
2 Background and Related Works	7
2.1 WSAWs and RSNs	7
2.2 Computational Intelligence	11
2.2.1 Evolutionary Algorithms	11
2.2.2 Swarm Algorithms	13
2.2.3 Fuzzy Logic	15
2.2.4 Learning Systems	16
2.2.5 Hybrid Systems	18
2.3 Multi-Objective Optimization	18
2.4 CI in WSAW and RSNs	19
2.5 Risk Management Framework	22
2.6 Agent-based Simulations	27
2.7 Chapter Summary	28

3	RMF and Modifications	30
3.1	Reactive Risk Management Framework	30
3.2	Auction Protocols for Multi-Robot Task Allocation	31
3.3	RMF For Hybrid Robotic Sensor Networks	35
3.4	Continuous Risk-Aware Response Generation	39
3.5	Multi-Objective Ant Colony Optimization	42
3.6	Chapter Summary	46
4	Case Studies in Critical Infrastructure Protection	47
4.1	Critical Infrastructure Protection	47
4.2	Maritime Smuggling	49
4.3	Maritime Supply Chains: Disruption Mitigation	52
4.4	Maritime Supply Chains: Disruption Prevention	56
4.5	Chapter Summary	57
5	Applications and Experimental Results	59
5.1	Application for CIP Perimeter Coverage	60
5.1.1	Risk Model and Risk Features	60
5.1.2	Risk Mitigation	60
5.1.3	Result of Auction Protocols for RSNs in CIP	64
5.2	Maritime Smuggling Detection and Mitigation	70
5.2.1	Risk Model for Suspicious Activity	70
5.2.2	Maritime Smuggling Behaviour Detection	72
5.2.3	Risk Mitigation	74
5.2.4	Refinement and Validation	76
5.2.5	Result of Hybrid RSNs for Maritime Smuggling Mitigation	80
5.3	Supply Chain Disruption Mitigation	83
5.3.1	Disruption Risk Model and Features	84
5.3.2	Disruption Mitigation Optimization	86
5.3.3	Refinement and Validation	89

5.3.4	Result of rRMF for Supply Chain Disruption Mitigation . . .	90
5.4	Supply Chain Disruption Prevention	96
5.4.1	Disruption Mitigation Prevention	96
5.4.2	Refinement and Validation	98
5.4.3	Results of MO-ACO for Risk-Aware Ship Routing	99
5.5	Chapter Summary	104
6	Trends, Future Work, and Conclusion	106
6.1	Research Trends	106
6.2	Future Work	107
6.3	Conclusion	109
	Bibliography	111

List of Algorithms

3.1	First-Price Sealed Bid Protocol for Auctioneer	33
3.2	Dutch-Japanese Protocol for Auctioneer	35
3.3	English Protocol for Auctioneer	36
3.4	Resource Fitness Function	38
3.5	Connectivity Fitness Function	39
3.6	Relevancy Fitness Function	39
3.7	MO-ACO Solution Construction	44
3.8	Pheromone Update Procedure	45
3.9	MO-ACO Algorithm	46
5.1	Cost Fitness Function	75
5.2	Latency Fitness Function	76
5.3	Risk Mitigation Fitness Function	76
5.4	Cost Fitness Function	89

List of Figures

2.1	Conceptual Diagram of Wireless Sensor Network	8
2.2	Conceptual Diagram of Wireless Sensor and Actuator Network	8
2.3	Commonly found problems in WSNs	9
2.4	Computational Intelligence Families and Techniques	12
2.5	Mamdani-type Fuzzy Inference System	16
2.6	Simple Artificial Neural Network	17
2.7	Pareto-front for convex optimization problem	19
2.8	Endsley's model of situation awareness [1]	22
2.9	Decision Support Systems	23
2.10	Conceptual Diagram of Risk Management Framework	23
2.11	Methodology for RMF	25
3.1	Reactive Risk Management Framework	31
3.2	Illustration of FPSB, English, Dutch, and Japanese Auction Protocols	33
3.3	Encoding of MRTA Solution for UAV Monitoring	37
3.4	Example of the foraging behaviour of ants ¹	42
3.5	Visitation Rule Algorithm	45
4.1	United States Department of Defense CIP Event Lifecycle ²	48
4.2	Example of RSN for CIP	49
4.3	Environment near Port of Barcelona	51
4.4	Maritime Smuggling Scenario	52
4.5	Example Supply Chain	53

4.6	Consequences of a Disruption	54
4.7	Example Disruption Scenario with Parameters	55
4.8	Example Scenario for Disruption Prevention	57
5.1	Chromosome for Repositioning	61
5.2	E-D-HO Configuration	67
5.3	E-D-HO Results for First-Price Sealed Bid	68
5.4	E-D-HO Results for Dutch-Japanese Protocol	68
5.5	E-D-HO Results for English Protocol	69
5.6	Maritime Smuggling Detection and Mitigation Overview	70
5.7	Mitigation Task Chromosome	75
5.8	Geographic area for maritime smuggling case study	77
5.9	Merchant Ship Behaviour	78
5.10	Ferry Behaviour	78
5.11	Small Recreational Ship Behaviour	78
5.12	Smuggling Merchant Vessel Behaviour	79
5.13	Small Smuggling Vessel Behaviour	79
5.14	Initial Environment	80
5.15	Monitoring Task Allocation	81
5.16	Smuggling Ship Risk Over Time	82
5.17	Successful Risk Mitigation	83
5.18	Example Gene	87
5.19	Example Chromosome	88
5.20	Disruption Risk for Vessel 1 over Time	91
5.21	Disruption Risk for Vessel 2 over Time	91
5.22	Disruption Risk for Vessel 3 over Time	92
5.23	MOEA Optimal Solutions Sample Risk over Time	92
5.24	MOEA Optimal Solutions Sample Cost over Time	92
5.25	MOEA Optimal Solutions Sample Resiliency over Time	93
5.26	Heuristic for Next Node Visitation	98

5.27	Number of Solutions per Algorithm	100
5.28	Solution Spread Distance to Best	100
5.29	Average Crowding Distance	101
5.30	Hypervolume Distance to Best Solution Found	102
5.31	Algorithm Pareto-Fronts for Experiment 33	102
5.32	MO-ACO Solutions for Experiment 10	103
5.33	NSGA-II Solutions for Experiment 10	103
5.34	Mixed Algorithm Solutions for Experiment 10	103
5.35	Sequential Algorithm Solutions for Experiment 10	104

List of Tables

1	List of Abbreviations	xii
5.1	Risk Mitigation Value Mapping	62
5.2	Worst-Case Auction Protocol Complexity	63
5.3	Scenarios	65
5.4	Sparse Scenario Metrics	66
5.5	Dense Scenario Metrics	67
5.6	Mitigating Assets	75
5.7	Monitoring Solution Fitness	81
5.8	Mitigation Solution Fitness	82
5.9	Fitness Values of Select Solutions in Sample 16	94
5.10	Fitness Values of Select Solutions in Sample 151	94
5.11	Actions in Solutions in Sample 16	95
5.12	Actions in Select Solutions in Sample 151	95

List of Abbreviations

AIS	Automatic Identification System
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ANN	Artificial Neural Network
ASN	Aerial Sensor Network
AoI	Area of Interest
CI	Computational Intelligence
CIP	Critical Infrastructure Protection
CoA	Course of Action
DSS	Decision Support System
EA	Evolutionary Algorithm
FIS	Fuzzy Inference System
GA	Genetic Algorithm
HMM	Hidden Markov Models
MCDA	Multi-Criterion Decision Analysis
ML	Machine Learning
MOEA	Multi-Objective Evolutionary Algorithm
MOO	Multi-Objective Optimization
MRTA	Multi-Robot Task Allocation
NSGA-2	Non-dominated Sorting Genetic Algorithm II
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
RMF	Risk Management Framework
RMU	Risk Mitigation Unit
RSN	Robotic Sensor Network
TSP	Traveling Salesman Problem
UAV	Unmanned Aerial Vehicle
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network

Table 1: List of Abbreviations

1 Introduction

Situation awareness is a complex process so fundamental to any task that it is often overlooked, yet it imposes constant constraints and limits the scope of our activities. How efficient would supply chains be if operators could be omniscient and omnipresent, predicting events before they happen and solving them, when they do, with optimal course of actions (COAs)? Could air traffic accidents be prevented, piracy avoided, or maritime disasters averted if the proper decision maker would have total knowledge of the state of their environment? Yet, no operator would have the ability to monitor the overwhelming amount of information that would be generated in order to accomplish this, nor would they be able to predict the possible stream of events that could unfold from the current state of the environment, while thinking of different possible solutions and optimizing them. All of these tasks would be unthinkable and even impossible, without even considering the amount of time that these critical processes need. However, this problem does not mean that such a system is impossible, only that certain tools are needed in order to reduce the amount of information, predict possible events, and optimize proper course of actions.

The described tool has two key terms descriptors: omniscient and omnipresent. The former requires a knowledge of everything, and the latter requires the ability to be present and act from anywhere. In order to know everything in a certain region, one must be able to perceive the environment in the area of interest (AOI). Perception is different than knowing and, consequently, there must also be the ability to reason through all the information to determine what is relevant and what is not. Finally, total knowledge includes information about the future. Prediction becomes a third

critical element.

An omnipresent system would have the ability to act anywhere, anyhow, while an omniscient system would act in the most optimal way possible. Of course, neither a truly omniscient nor an omnipresent system is feasible, but it is an objective that will and does drive the development of such decision support systems (DSSs), much as computer memory should be infinite, or that all algorithms should have constant complexity. Designing such a system is a daunting task but a few research streams have proven quite valuable.

The first field of immediate relevance is that of wireless sensor networks (WSNs). WSNs are networks of wirelessly connected sensors deployed in an AoI that allow for a wider perception of the environment [2]. The ability to act upon the environment can be given to these networks by adding actuator nodes transforming WSNs into wireless sensor and actuator networks (WSANs) [3]. Recently, the Internet of Things (IoT) has gained momentum as a new concept that shares many ideas with WSANs, the latter being networks that are more dedicated and that usually operate on a smaller scale.

A second research stream is Computational Intelligence (CI) and machine learning (ML) [4]. CI is a vast field composed of generally nature-inspired techniques, often meta-heuristic in nature, that attempt to solve problems that often have no feasible (or deterministic) solutions. ML is a more mathematically inspired field that attempts to learn relationships from existing data. Both of these fields are actively being researched and the boundaries often get blurry. These fields allow us to reason on data by providing techniques that can learn what is relevant, infer new data, and predict future states. Additionally, they also allow for relatively fast optimizations that can find feasible solutions, even when considering multi-objective situations.

Finally, the research on event modeling [5], knowledge representation [6], and risk management, is highly important. In order to make sense from the incoming data, it has to be represented in a certain way. These representations must be understood as environment states. A sequence of these states can be considered to be more

important than others, and can be modeled as events. Ultimately, these can all be combined at a high level with a risk-based perspective. From this perspective, undesirable events must be mitigated, the state of the environment must be controlled and maintained in desirable ones.

This thesis does not claim to have built a perfect system, seamlessly integrating all research streams. It starts with an actively researched risk management framework (RMF) [7, 8, 9, 10] that combines many of the described techniques to solve a vast array of problems. There remains many unsolved problems and many more to be uncovered. This thesis attempts to solve some of them and push the framework towards new possibilities.

The following sections will first discuss the motivations behind this research, followed by the objectives and contributions, and conclude with a section detailing the organization of this thesis.

1.1 Motivations

The motivation behind this research is plainly to research problems and solutions related to DSSs, or a system that enhances the situational awareness [1] of decision makers. A DSS can be applied anywhere that an operator requires more control over an environment, as vague and wide as that definition may be. This thesis will put emphasis on a DSS system for the problem of critical infrastructure protection.

Particularly, problems related to COAs optimized for efficiency and other metrics, and a general application methodology are of interest. Generating COAs requires the representation of knowledge by representing actions in such a way as they can be combined to create greater actions. It also requires the optimization of these combination of actions in order to determine the best solution per the defined objective functions. Real problems rarely have a single goal, and more often have more than one conflicting objective. These multi-objective optimization (MOO) problems require

multiple solutions that cover the tradeoff between these objectives, referred to as the front of Pareto-optimal solutions [11], or the Pareto-front.

Developing a general DSS is also compelling to be able to apply the system to many situations. This is a daunting task since different scenarios have often drastically different action possibilities, active actors, environmental conditions, and constraints. Consequently, steps that are few in number and easy to apply, while being not too restrictive in their assumptions, have to be determined. Environment models are crucial to both developing the methodology, and determining COAs. Hence, environment modeling, action modeling, and knowledge representation will often appear in this thesis.

The methods to solve the described problems are equally interesting. CI and ML are hot research fields, with many new algorithms that solve problems of increasing complexity. CI and ML algorithms allow for more complex event modeling and knowledge representation, and for new DSS systems to be explored, bringing the research output to the cutting edge of many different fields.

The ambitious project of a general DSS would be invaluable in many applications, and this remains a constant motivation for pouring efforts into this research. It could help save lives when applied for first responders, help to mitigate environmental disasters like oil spills, create more efficient economies via automation, or help defend sovereignty in military applications, to give a few examples of relevant domains of applications

1.2 Contributions

There are three objectives to this thesis: (1) Derive a methodology for applying the RMF in a general context; (2) Explore more abstract critical infrastructure protection scenarios that require risk identification, detection, and mitigation; (3) Propose modifications pertaining to different modules of the RMF. To this end, many contributions are presented in this thesis. The following is a list of the major contributions

of this thesis.

1. A reactive, distributed version of the RMF was designed and implemented. The IoT provides a different perspective of the world, and consequently, a different approach is needed when developing applications for it. This work was published in a conference paper [12];
2. Two new auction protocols are developed for the Fuzzy-Auction Multi-Robot Task Allocation (MRTA) technique proposed in [8]. All three protocols are analysed in networks of varying density and heterogeneity. This work was published in a conference paper [13], and was also presented at [14];
3. Unmanned Aerial Vehicles (UAVs) are added into WSANs by formulating their integration as part of the RMF [15]. The increased information gathering capabilities of UAVs allows for better risk mitigation. This work was published in a conference paper [15];
4. A CI technique to generate optimized solutions in dynamic environments. The proposed technique adapts solutions to changing constraints and objective functions. This work as also published in [12]; and
5. The generation of risk-aware routes for maritime vessels using an original CI technique. This contribution adds the ability to propose Pareto-optimal routes in terms of safety and risk for vessels. This work was not yet published at the time when this thesis was written, but was presented at [16].

Additionally, four case studies and simulations of the RMF [13, 15, 12, 16] are given. The domains include Critical Infrastructure Protection (CIP), maritime smuggling mitigation, maritime supply chain disruption mitigation, and vessel navigation. As the underlying means of perception changes from WSANs to the IoT, so does the framework and its requirements. These contributions were added with the intention to build a smarter, more efficient DSS for increasingly complex perceptions of the environment provided by the IoT. Ultimately, this leads to an increased ability to detect, identify, and mitigate risk by giving better situational awareness to decision

makers.

1.3 Thesis Organization

This thesis contains six chapters, with this section ending the first. Chapter 2 presents an in-depth literature review and relevant background information on many domains that are explored in this thesis. Chapter 3 details the existing architecture of the RMF and how to apply it, present a reactive and distributed version of it, and then propose modifications for risk detection, identification, and mitigation in various environments. Chapter 4 describes various case studies for the application of the RMF. Chapter 5 describes the application of the RMF following the methodology previously presented, develops agent-based simulations, and presents the results of these simulations. Finally, current research trends, future research avenues, and a conclusion is given in Chapter 6.

2 Background and Related Works

Chapter 2 will present works and necessary background information that will serve as context for the rest of the thesis. The topics that are presented in this section are: wireless sensor and actuator networks (WSANs) and robotic sensor networks (RSNs), computational intelligence (CI), multi-objective optimization (MOO), CI in WSANs/RSNs, a previously developed risk management framework (RMF), and finally, a quick review of agent-based simulations.

2.1 WSANs and RSNs

Research on WSANs and RSNs served as the basis for the beginning of the research and later on served as a springboard into other types of networks with similar problems. Wireless sensor networks are composed of wirelessly interconnected, usually self-powered sensor nodes used to monitor an area of interest (AoI). These nodes have limited computational power and usually send sensed data about their environment to special nodes known as sink nodes. Sink nodes are used as an interface to the network. Such networks have applications in many domains such as the military, healthcare, environmental, and for industrial and home monitoring [17]. Hardware failures, power exhaustion, and insufficient computational power are common problems in these networks [2]. Figure 2.1 presents a conceptual view of a WSN.

WSANs are WSNs with nodes that are capable of performing distributed computations and perform actuation tasks [18]. Any node within a WSAN can usually perform as a sensor, an actuator, a sink node, or a combination of these roles per its available

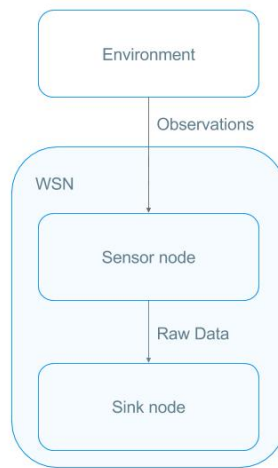


Figure 2.1: Conceptual Diagram of Wireless Sensor Network

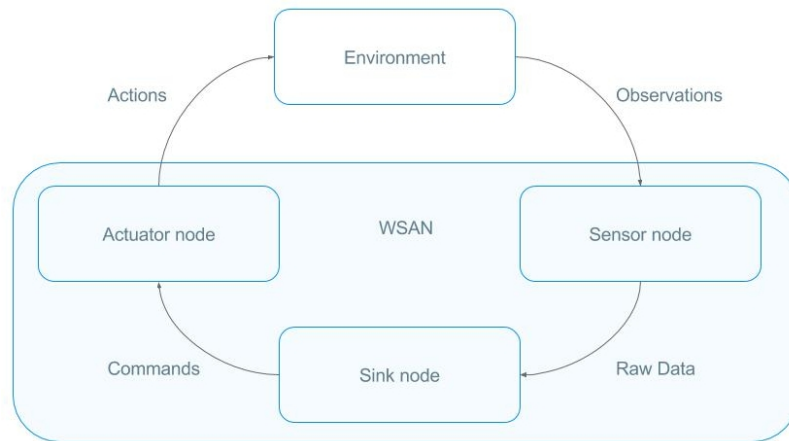


Figure 2.2: Conceptual Diagram of Wireless Sensor and Actuator Network

hardware. Actuator nodes typically have more resources than sensor nodes. Actuators can be mobile, allowing them to change the topology of the network. Figure 2.2 presents a conceptual diagram of WSANs, where the sensors send their perception of the environment to the sink node that processes the data, and perhaps asks assistance from the operator. The WSAN then sends commands to actuators that finally change the state of the environment. Robotic Sensor Networks (RSNs) are a type of a WSAN where some or all of the nodes have enough computational power, energy, and hardware to perform all three roles of sensing, actuation, and sink node.

WSANs have the same problems as WSNs such as communication issues and resource depletion, in addition to new problems such as localization, sink mobility, actuation,

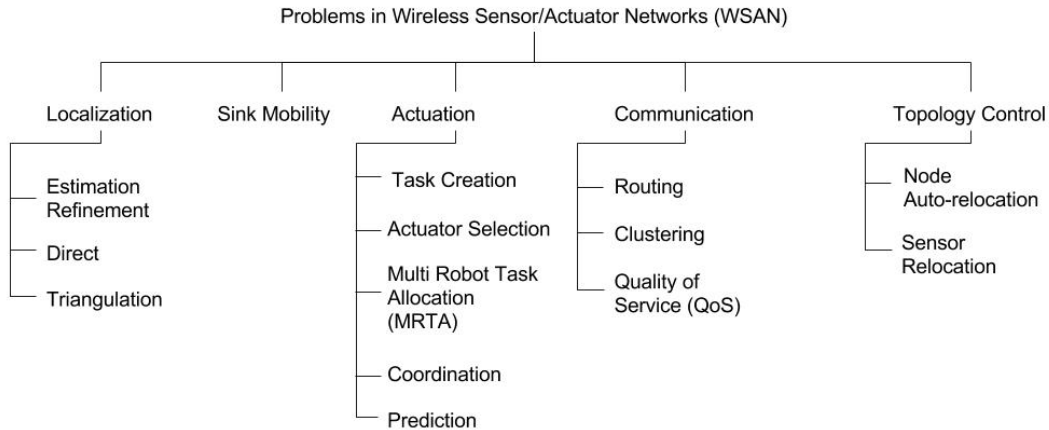


Figure 2.3: Commonly found problems in WSNs

communication issues related to the possible dynamic topology of the network, and topology control. These problems are illustrated in Figure 2.3. While this thesis mostly concentrates on the problems of actuation and topology control, it's important to understand that other concerns that may be present in WSNs when developing new systems, such as environmental factors.

Localization has typically been solved through three different methods, either by estimating a location through messages from anchor nodes that know their position [19], by using meta-information from the message such as angle and time of arrival [20], or by triangulation [21]. Methods that use meta-information are good in that they require less messages to compute positions and can be used independently, but lose precision when applied in closed environments, i.e. indoor environments, due to complex paths signals may take. Localization by triangulation suffers from the issue of complex signal paths, to a lesser degree, but is more simple and can return very good approximations for positions. The problem with triangulation is that it requires at the very least three well positioned nodes, and many more for adequate accuracy.

Sink mobility is one of the oldest problems found in WSNs. In static topology networks, nodes closest to the sink node are constantly broadcasting messages causing them to exhaust their resources quickly, something that is called the sinkhole problem. The sink mobility problem attempts to find an optimal route for the sink node in order to visit all required nodes. Such works as [22] attempt to solve this problem by

formulating it as a form of the Traveling Salesman Problem (TSP). Formulating the sink mobility problem as a TSP can be useful but is not always realistic in WSANs because nodes can communicate in a certain area, so a sink node visiting each node is not an optimal solution.

Actuation is a bigger problem that can be subdivided into multiple components such as task creation, actuator selection, multi-robot task allocation (MRTA) [23], actuator coordination, and task prediction. MRTA is the abstract problem of optimizing the assigning of tasks to a set of capable agents, such as WSAN nodes. The work presented in [24] contains many of these tasks, where a networks of robots must detect abnormal behaviour and send probes to collect new information. This work is specific to WSANs for intruder detection, and lacks generality. A more general approach to the actuation problem is detailed in this thesis.

Communication issues, like finding efficient routes to the sink node, are present in WSNs but are accentuated in WSANs due to its dynamic topology and heterogeneous nodes. Commonly found problems are routing, clustering, and quality of service. The work presented in [25] presents a routing algorithm to discover routes to a mobile sink node. It uses a complex algorithm that may not be suitable in highly mobile networks.

Finally, the problem of topology control can be subdivided into two problems: node auto-relocation, and sensor deployment. The first concerns mobile sensor nodes that reposition themselves to restore sensor coverage over the AoI, while the second involves mobile actuators positioning immobile sensor nodes. The work presented in [26] coordinates multiple robotic nodes to reposition sensor nodes in order to proactively prevent coverage loss. This last work explores many different MOEAs but doesn't explore other computational techniques that can be helpful in solving this combinatorial problem.

2.2 Computational Intelligence

The Institute of Electrical and Electronics Engineers (IEEE) Computational Intelligence Society (CIS) defines computational intelligence (CI) in their constitution ¹, Article I, Section 5 as "the theory, design, application, and development of biologically and linguistically motivated computational paradigms emphasizing neural networks, connectionist systems, genetic algorithms, evolutionary programming, fuzzy systems, and hybrid intelligent systems in which these paradigms are contained."

CI is the application of soft-computing techniques to provide solutions to often NP-Hard problems. The definition of CI is still not well agreed upon but some concepts are understood to be fundamental to CI. The methods are nearly always heuristic in nature meaning they provide solutions that may be inexact, hence, not always optimal or precise. CI algorithms are often, if not always, inspired by nature taking some ideas but not replicating precisely the mechanism by which they are performed in nature.

Three categories of CI techniques are considered as fundamental: evolutionary computation, artificial neural networks (ANNs), and fuzzy logic [4]. The following subsections will present four categories, making the distinction between evolutionary computation and swarm algorithms: evolutionary algorithms (EA), swarm algorithms, fuzzy logic, learning systems, and hybrid systems. Figure 2.4 presents some of the most popular CI techniques and their sub-categories. A fifth category for hybrid algorithms is also present, though it is not presented in Figure 2.4 due to the often unique algorithmic hybridizations.

2.2.1 Evolutionary Algorithms

Evolutionary algorithms attempt to mimic evolution to discover appropriate solutions to a problem [27]. A typical EA has a population composed of individuals, where each

¹IEEE Computational Intelligence Society, Constitution. (2017, July 13). Retrieved from <http://cis.ieee.org/constitution.html>

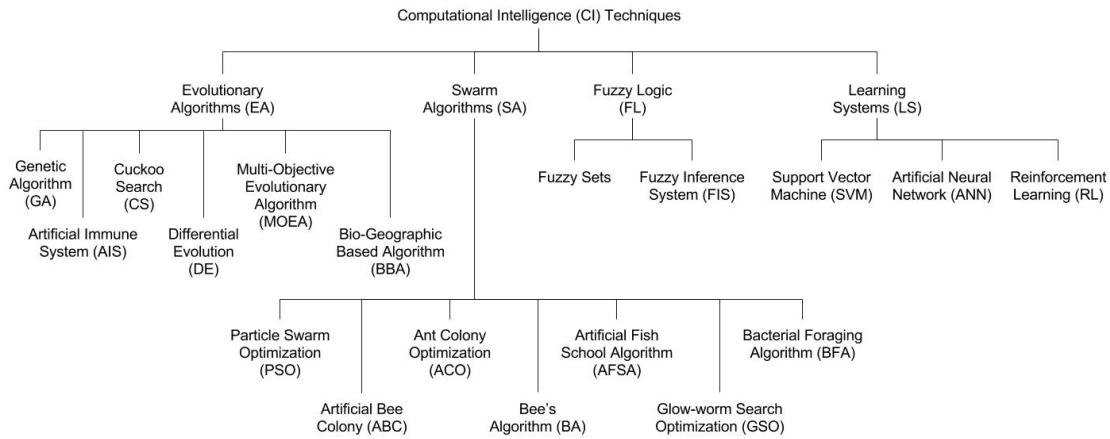


Figure 2.4: Computational Intelligence Families and Techniques

individual is represented by a chromosome that is a collection of genes. This population is mutable, with operators operating at the intra-individual or inter-individual level to create new individuals or modify existing ones.

One iteration of such an algorithm usually has the following actions: First, the population is modified by the genetic operators, typically a mutation and a crossover operator. The former mutates existing chromosomes into new ones and the latter creates new chromosomes from at least two existing ones. Then, the individuals in the current population are evaluated by the fitness function(s) through some process with only the most fit individuals selected to form the population of the next iteration, referred to as the next generation of the population.

The fitness function can be optimized by either maximizing or minimizing it, leading to a population of best solutions. This is repeated until some defined stopping criteria is reached which is usually a maximum number of generation, a threshold fitness function value, or execution time. The best solution in the population, as defined by the fitness function, is then defined as the solution for the problem. There can be more components to this such as constraints on individuals, excluding them from the population, or modifications of the steps in the iteration. Techniques in this CI category are good at solving optimization problems or combination problems.

Genetic Algorithms (GAs) are one of the first, and also the closest to the definition of EAs. A typical GA will follow the iteration process given above, starting with a

randomly generated population. There are usually two operators, a mutation operator that mutates an existing solution into a new one, and a crossover operator that generates offspring that may be part of the next generation. There are many variants of these operators that are tailored to produce better individuals for particular situations. Finally, a single fitness function that must be minimized or maximized is used to evaluate the individuals. The best individual is returned from such an algorithm as the best solution found. A GA's performance is hugely influenced by the stopping criteria. There may be constraints on the solutions.

Multi-Objective Evolutionary Algorithms (MOEAs) are an extension of GAs, used to find optimized solutions for multiple fitness functions and multiple objectives. The immediate difference between these techniques and GAs is the fact that a single solution can rarely be given since an individual that optimizes a fitness function will rarely optimize the others. Consequently, trade-offs must be made between optimizing one function versus another. The solutions that do not strictly dominate each other form the Pareto-optimal front [28]. MOEAs return a set of solutions that are part of this front, with another selection mechanism in place afterwards to make the final choice. One currently commonly used MOEA is the Non-dominated Sorting Genetic Algorithm (NSGA-II) [29].

Other less prominent EAs were also found which are mentioned here for completeness, such as Cuckoo search [30], differential evolution [31], artificial immune systems [32], and bio-geographic based algorithms [33].

2.2.2 Swarm Algorithms

Similarly to EAs, swarm algorithms, or swarm intelligence, is a branch of naturally inspired algorithms based on the interaction between organisms [4]. If EAs are inspired by the biological phenomenon of evolution, swarm algorithms are inspired by the flocking behaviour of large groups, often referred to as swarms. There are many common features shared between swarm algorithms and EAs, for example populations

of individuals, iterations, and stopping criteria, to name a few. However, generally, there is no concept of generations or evolutions within SAs.

Swarm algorithms work with the hypothesis that many individuals acting separately, but cooperatively, may be able to find a better solutions faster than on their own. Similarly to EAs, swarm algorithms are generally gradient-free. Instead they use other methods to explore the solution space, attempting to optimize a fitness function. These algorithms are mainly used to discover combinations of values that maximize a certain function. Hence, they are best suited for combinatorial and optimization problems.

Particle swarm optimization (PSO) is a swarm algorithm that is based on the flocking behaviour of birds. First, a population of particles is created. Then, for each iteration, a particle's position and speed are updated relative to the performance of their neighbours. PSO attempts to explore the solution space by having groups of particles "flock" to high value positions, hence discovering better combinations of values for the fitness function. This iteration is repeated until some stopping criterion holds true, stopping criterion that are generally the same as EAs. Similarly to GAs, PSOs have been adapted to generate multi-objective solutions.

Ant Colony Optimization (ACO) is another well known swarm algorithm. This algorithm is based on the food finding behaviour of ant swarms. As the ants explore they deposit pheromones (a chemical substance). They proceed to evaluate it upon encountering a food source then head back to the nest, depositing pheromones in proportion to the food source quality. ACO works similarly by first defining a set of solution components and a set of pheromone values, the pheromone model. The set of pheromone values corresponds directly to the set of solution components. ACO works by probabilistically and iteratively assigning higher pheromone values to good solution components. These higher pheromone values help the algorithm focus on relevant areas. ACO method is relevant to combinatorial problems like routing, and is useful for optimization.

Other swarm algorithms include artificial bee colony [34], bacterial foraging algorithm

[35], artificial fish swarm algorithm [36], bee's algorithm [37], and glow-worm swarm optimization [38]. Bacterial foraging algorithm is based on the foraging behaviour of swarms of bacteria through simulated chemotaxis, reproduction, and elimination and dispersal.

2.2.3 Fuzzy Logic

Fuzzy logic is a very different method than EA and swarm algorithms. While the latter techniques are based on solving combinatorial problems or optimization problems, fuzzy logic adds the ability to reason in non-crisp ways [39]. Fuzzy logic adds a degree of membership of an element to a set instead of reasoning in absolutes, where an element is either part of a set or not. The membership is determined by a membership function that maps elements to a membership degree between 0 and 1. Fuzzy logic is useful when reasoning about environments that are imprecise, which is more common to human reasoning.

Fuzzy Inference Systems (FIS) are systems that are able to infer a crisp value from a set of inputs, fuzzy sets and their membership functions, and inference rules [40]. The two most commonly found FIS models are the Mamdani [41] and the Takagi-Sugeno FIS model [40]. An inference rule is one that follows a defined grammar, syntax of precedents, antecedents and a weight. The rules of an FIS are usually hard to determine manually, requiring expert knowledge or much experimental validation. Such a system is usually constructed as such: First, the inputs are fuzzified into their corresponding fuzzy sets via the membership functions. The fuzzy sets are labelled by linguistic terms. The inference rules use these linguistic terms as precedents to add weight to the antecedents, which are either linguistic terms corresponding to linguistic terms for output fuzzy sets in the Mamdani FIS model, or functions over the inputs that give values in the case of the Sugeno FIS Model. The impact of each rule is weighted such that some rules could be more important than others. Finally, this output is defuzzified through appropriate methods like the centroid method for

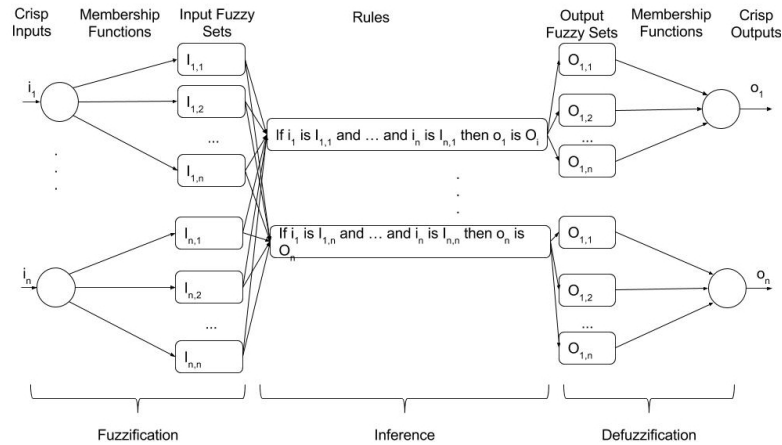


Figure 2.5: Mamdani-type Fuzzy Inference System

the Mamdani FIS, or the weighted average of the output values from the rules for the Sugeno model. A Mamdani FIS model is shown in Figure 2.5.

2.2.4 Learning Systems

A learning system uses the concept of learning relations between a set of quantitative values, referred to as features, and states or objects [42], similarly as living organisms do. This relationship is not known by the system and must be "learned" from empirical data. There are three classes of learning systems that are generally recognized: Unsupervised, Supervised, and Reinforced. Each of these methods attempt to learn the correct model in order to infer the correct state of a set of features.

In supervised learning, a set of input features are mapped to one of two or more discrete states. A supervised learning system has a learning phase in which the system is fed a training set of features, and class tuples. The system attempts to optimize its internal model to have the highest number of correct inferences were it not given the class. When a satisfactory level of precision and recall is attained, the system is used to infer the state of yet unseen features. These systems are widely used for classification purposes.

In unsupervised learning, the system does not have the state corresponding to a certain set of features, however, attempts to determine the relationship between certain

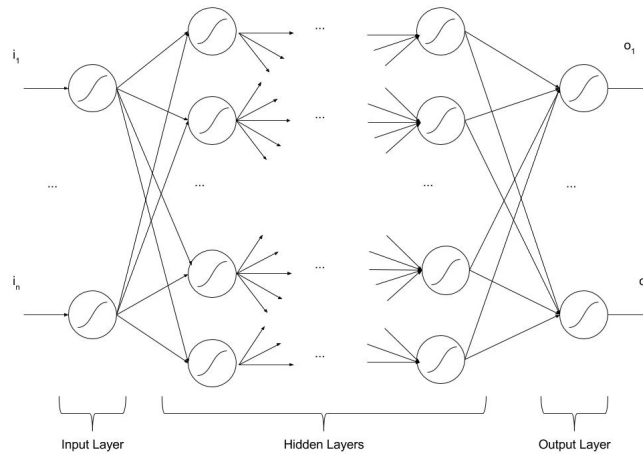


Figure 2.6: Simple Artificial Neural Network

features, and create states itself. Finally, in Reinforced Learning, there is no training phase. Instead, the system is fed features, an inference is made, then the system is given a value proportional to the error in the inference. The system then attempts to minimize this error.

Artificial Neural Networks (ANNs) are loosely inspired by the brain. A simple neural network is typically composed of interconnected neurons in layers [43]. It has an input layer, the output layer, and one or more hidden layers. Each neuron is composed of an activation function. All edges between layers are weighted. A typical example of a neural network is given in Figure 2.6. ANNs can be used in classification with the output being the class for the features, in reinforcement learning as the policy function, in adaptive controllers where the outputs are the control signals, and many more applications. Research in ANN is very active, with architectures of increasing complexity being used [44].

Reinforcement learning is a machine learning method to determine an optimal policy that dictates which actions to take in certain states. The problem is often formulated as a Markov Decision Process (MDP), where there is a set of states, and a set of actions. The problem is then to determine the optimal action/state associations, referred to as the policy function. Common RL algorithms are Q-learning and State-Action-Reward-State-Action, commonly referred to as SARSA, with intensive research ongoing in new methods based on ANNs and deep learning [45].

2.2.5 Hybrid Systems

Hybrid Systems (HS) are a CI category that captures any system that combines two or more CI techniques that are usually combined to make up for the shortcomings of any one of the methods. Fuzzy adaptive resonance theory is one popular example of a hybrid system [46] that combines fuzzy logic and ANNs. SAs and EAs are commonly hybridized due to their similarities [47].

2.3 Multi-Objective Optimization

Multi-objective optimization (MOO) is the problem of finding solutions to optimization problems that have multi-dimensional, often constrained solution spaces, with objective functions that are often contradictory [11]. MOO problems usually do not have one solution, but a set of solutions that are Pareto-optimal that optimize the objective functions, or in other words, solutions that dominate other solutions in one objective but not in all.

In order to solve such problems, a multi-objective problem has to be modeled, the objective functions have to be explicitly defined, and any constraints have to be given. Then, methods can be used to solve the problem. There exists many multi-objective variants of the CI methods covered in the previous section that exist, such as NSGA-II [29], multi-objective ant colony optimization (MO-ACO) [48], and many others. It is important to note that CI techniques have no proof of convergence, and consequently the solutions returned might not be close to the true Pareto front.

Figure 2.7 presents an example of a Pareto-front for a two-dimensional MOO problem. The goal of a MOO algorithm is to estimate, as much as possible, the true Pareto-front with a set of discrete solutions that can be represented as points. There are multiple metrics that can be used to evaluate a set of solutions returned from an algorithm, such as hyper-volume, distance from a reference set, spread, spacing, etc. [11].

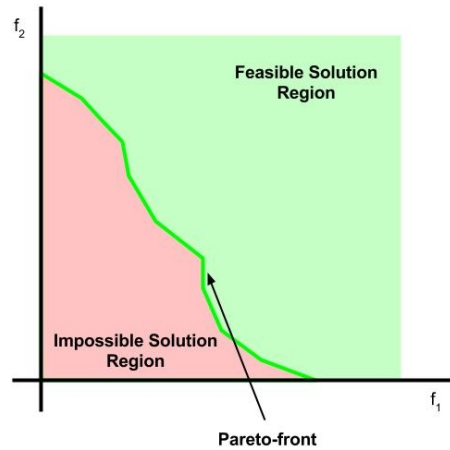


Figure 2.7: Pareto-front for convex optimization problem

2.4 CI in WSN and RSNs

CI has seen extensive use in WSNs for a variety of problems. The following section will present relevant examples of works that use CI to solve actuation tasks, communication, sink mobility, topology control, and localization.

The problem of actuation has many examples of CI use. An interesting application of WSNs is presented in [49], where the WSN serves as a DSS to guide occupants of a building towards exits. In this network, motion sensors are used as sensors and light switches as actuators. The network uses an ANN to allocate light on or off tasks to the actuators, and learns to predict these tasks during its training phase. The authors of [24] use Fuzzy Adaptive Resonance Theory to detect abnormalities and send a node to probe the area of the abnormality, while the authors of [50] uses a GA to select appropriate sensor positions to track a moving object. Finally, the work in [51] uses an FIS in each node in order to avoid collisions between moving nodes.

The ANN in [49] and the work in [24] require training sets to learn the appropriate actions. Training data is not always available, but a bigger problem is that these methods require more defined topologies and configurations. For example, if a sensor were to be added to the network, new training data would be required for the first application. The third system [50] uses a GA to select appropriate nodes for task allocation, and consequently is more flexible to changes in topology. However, it con-

siders only one objective and this considerably simplifies the reality in which WSAANs operate. Finally, the last application [51] has an interesting use of an FIS to give the nodes the ability to coordinate among themselves.

However, this method is rather simplistic and not suitable for more complex tasks, and more importantly does not optimize in any way the behaviour of the WSAAN. An actuation coordination method is used in this thesis combining a multi-objective GA and fuzzy systems for more efficient task allocations. The agents are able to detect whether actions are required via an FIS. Then, the MRTA problem is solved by giving the ability to self-include themselves for tasks using another FIS, and resolving efficient MRTA allocations with a multi-objective GA.

The problem of communication in WSAANs has been the focus of many researchers. Some attempt to solve the routing problem with CI, such as in [25] where an Artificial Immune System-Artificial Bee Colony algorithm is used to generate routes based on signal strength, latency, and energy use. Others try to find routes to mobile sinks, such as in [52, 53], which both use modified PSO algorithms. Groups of nodes known as clusters can also be used for more efficient communication. [54, 55] are two examples of algorithms that use GAs to form clusters. In the former, nodes first use an FIS to determine their ability of being cluster heads.

Sink mobility is perhaps the oldest problem of WSAANs, and one of the original motivating problems. Consequently, there are many approaches to solving this problem. [56] uses an artificial immune system algorithm to determine which nodes to visit. These nodes become cluster heads. [57] uses an ACO algorithm to determine a mobile sink route that reduces the number of communication hops needed, goes through regions with higher residual energy, and the distance that has to be travelled. The authors of [58] consider a case with multiple mobile sinks, then formulate the problem as a TSP problem with multiple agents and use a GA to solve it.

Topology control is another commonly researched problem in WSAANs in which CI can be found. The authors of [59] propose a scheme where mobile nodes are randomly deployed, then optimized for network coverage via an Artificial Fish School Algorithm

(AFSA). A similar approach based on Glowworm Swarm Optimization (GSO) that optimizes distance traveled by the nodes, energy use, and redundant coverage is used in [60]. A variety of CI techniques are used for this problem, including MOEAs [61], PSO [62], multi-objective AIS [63], ACO [64], FIS [65], and Biogeography-based optimization [66], to name a few. Another tactic to control the topology of a network is to have actuators relocate static sensor nodes. This is used in [67] with a MOEA, in [68] for UAVs with a GA and PSO, and for UAVs again in [69] using PSO and Bacterial Foraging Algorithm.

Among the variety of methods used to solve topology control are MOEAs, FIS, and ACO. The use of MOEAs and FIS has been used previously in works concerning the RMF, with promising results [8]. However, the use of fuzzy systems in this last work was not fully explored, something that this thesis will pick up. The use of ACO in topology control and other problems in general was also noticed. Interestingly, multi-objective EAs were common in many works, but not a single use of a multi-objective ACO algorithm was surveyed. A MO-ACO is developed in this thesis, along with an application and an analysis.

Finally, research for the problem of localization in WSAANs seems to have made the most diverse use of CI techniques, with all categories present. The authors of [70] use a PSO algorithm to refine estimates of a node's position based on mobile anchor nodes. A similar method using a GA is found in [71]. [20, 72] use a trained ANN to determine the Cartesian coordinates of a node based on communication metadata. The authors of [73] use FL to estimate the probability of the position of a node based on received signal strength indicator in indoor environments. Finally, [21] uses an ABC/GA algorithm to plan an optimal route for a mobile anchor node to determine the location of nodes.

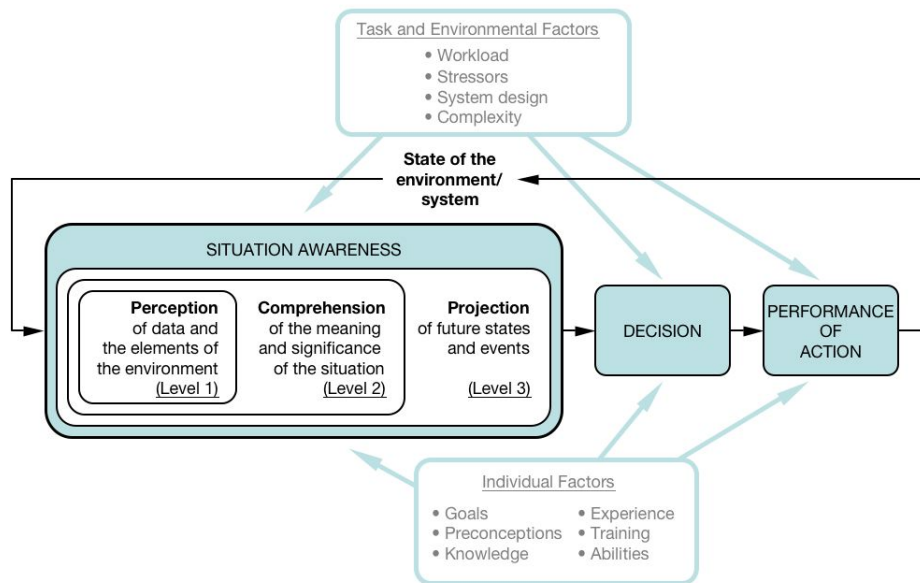


Figure 2.8: Endsley's model of situation awareness [1]

2.5 Risk Management Framework

This section will cover relevant background information on the risk management framework (RMF), a major component of this thesis. The RMF is a major component of a DSS intended to help operators gain and maintain an accurate and refined situational awareness. Situational awareness is the task of comprehending important aspects in the perception of the environment, predicting possible future states, and using this information in order to determine an appropriate COA. This is shown in Endsley's situational awareness diagram [1], shown in Figure 2.8.

In a completely manual mode of operation, an operator is confronted with a raw, unfiltered perception of the environment. As the quantity of information received from the environment increases, an operator may become overwhelmed. Consequently, a framework to manage the perception of the environment as well as filter out irrelevant information could be used by an operator in order to re-establish a relevant SA. Additionally, optimized COAs could be proposed to the operator to present a more refined control over the situation. Finally, the process could be improved and tweaked by the operator in order to specify the degree and type of information needed. This is represented in Figure 2.9.

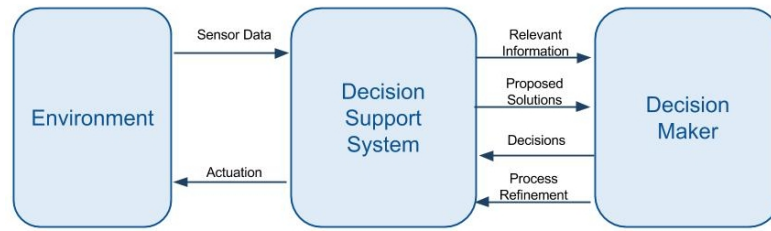


Figure 2.9: Decision Support Systems

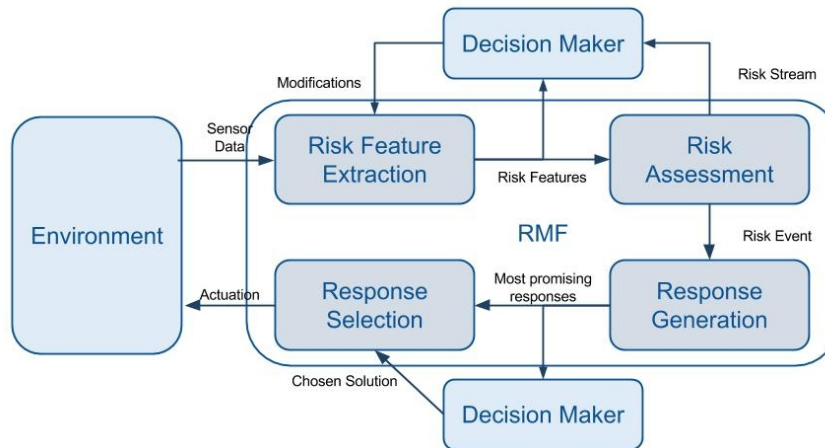


Figure 2.10: Conceptual Diagram of Risk Management Framework

The RMF was first presented in [7] to identify the risk of a node relative to a certain event by using information from WSNs and an FIS. Risk features are defined from raw data streams, fuzzified, and evaluated through a risk model for a specific event encoded as FIS rules. The output of this FIS corresponds to the risk of the modeled event. The RMF was applied for critical infrastructure protection (CIP), where the operator would be presented with the battery level risk, and the risk of intrusion based on distance. The operator would then review the values and alter the risk models.

The RMF was extended and applied for search-and-rescue operations in [74]. A response selection module was added, in which a set of multi-objective risk mitigating solutions are proposed and picked by an operator. This requires solutions to be encoded and modeled for an MOEA, NSGA-II in this case. The work was used in a simulated search-and-rescue operation. The risk management framework is depicted in Figure 2.10.

However, the RMF was first used in a robotic sensor network in [75] for critical infrastructure protection (CIP). The nodes in this work are capable of sensing their

environment and relocating themselves. Each node perceives its environment and determines its ability to maintain coverage over its portion of the perimeter. Once the risk of a loss in perimeter coverage increases past a threshold, a set of non-dominated, multi-objective risk mitigating solutions are created via NSGA-II. These solutions propose a range of solutions, from relocating nodes in order to maintain adequate coverage at the cost of energy, or maintaining current positions even as coverage is weakened.

This work was extended in [8] by allowing nodes to either bid to be part of the coalition that would be able to optimize to restore coverage, or to bid after on optimized positions. These bids are numeric values that indicate the suitability of a node based on multiple factors such as the remaining energy of a node, the redundancy in its current position, and the distance to the event. The bid was processed via an FIS.

This framework was augmented in [9] to allow the ingestion and fusion of soft data sources such as historical records of worldwide maritime incidents with hard data sources such as automatic identification system (AIS) data. The AIS is an identification system that is mandated by the International Maritime Organization's (IMO) International Convention for the Safety of Life at Sea (SOLAS) ² for international voyaging ships with gross tonnage of 300 or more, and all passenger ships. The work in [76] augments the RMF to be able to take behavioral intents into consideration as inferred from anomalies. The ability to evaluate different risk models that can be activated via contextual information was added more recently in [10].

A general application methodology developed through all the works can be repeated to apply the RMF for any context that requires risk management, and for any implementation. This methodology can be summed up as a series of steps, as shown in Figure 2.11.

The first step revolves around risk modelling. In order to manage the risk inherent to

² International Convention for the Safety of Life at Sea (SOLAS). (1974, November 1). Retrieved July 18, 2017, from [http://www.imo.org/en/About/conventions/listofconventions/pages/international-convention-for-the-safety-of-life-at-sea-\(solas\),-1974.aspx](http://www.imo.org/en/About/conventions/listofconventions/pages/international-convention-for-the-safety-of-life-at-sea-(solas),-1974.aspx)

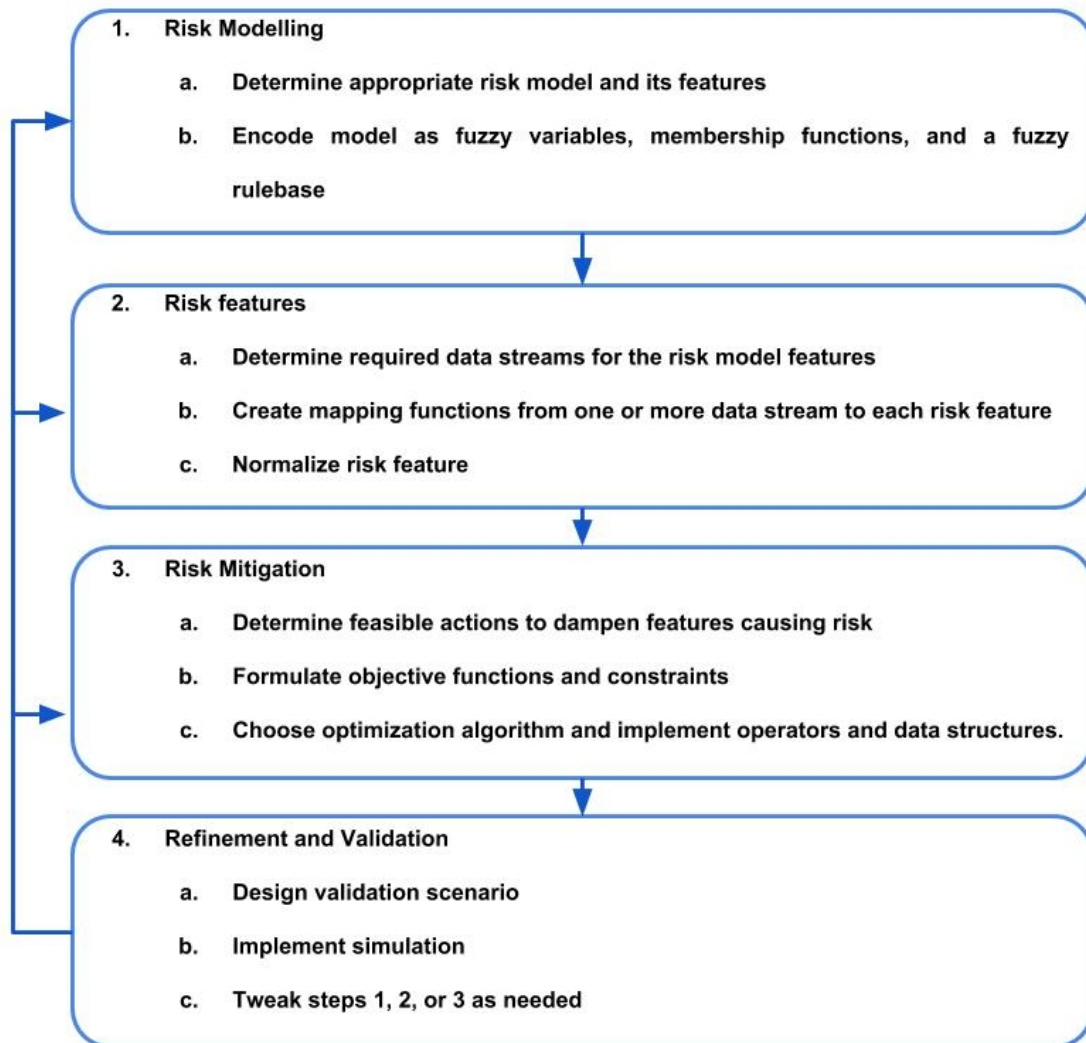


Figure 2.11: Methodology for RMF

certain environmental states, risky states must first be detected. Consequently, this step is the core of the risk management process. First, a risk model has to be specified in terms of features. For example, a forest fire risk is more likely in arid situations, historically fire-prone, highly-wooded areas. The relationship between features and the event can be hard to grasp with crisp values, while fuzzy variables and rules can be used to create a more natural model based more closely to human reasoning. Consequently, the model can be encoded as a fuzzy rulebase. The input features are crisp since they originate from digital computers so membership functions have to be created for the fuzzy variables. This fuzzy model can then be used in an FIS.

The second step of this methodology requires the creation of risk features. The objective of this step is to define the input sensors and the perception of the environment that is needed. To do this, three subsets are needed. The first is to determine the data streams needed in order to create the features that are needed for the fuzzy variables as defined in the first step of the methodology. These data streams are not always available, so the risk model itself may have to be revisited and modified accordingly. Following the identification of these data streams, functions mapping or reducing the one or multiple data streams have to be defined. The last step is to normalize these risk features into the range of 0 to 1.

The third and final step of the methodology is required if the risk management framework has the ability to take action upon the environment. The goal of this step is to define the actuation capabilities of the RMF. First, actions that are feasible by the actuators of the network in which the RMF operates have to be identified. These actions should result in the mitigation of the risk features that lead to a higher risk. Then, the objectives of the risk mitigation step have to be formulated. One of this objectives is usually the risk mitigation and another would be the cost, though there are usually more than one ways to evaluate cost. Once these objectives functions are defined, an optimization algorithm has to be chosen. Depending on the optimization algorithm, the proper data structures and operators have to be implemented. This entails the encoding of actions into a the proper data structure, and the implementation

of cost functions that can evaluate this encoding.

Following these steps, the RMF is ready to be tested. A scenario is envisioned for the RMF per its application. This scenario is then simulated, providing the needed data streams for the RMF and reacting to the actions of the RMF. Small tweaks to the results of the three previous steps are usually needed, refining the application. The application of the RMF is complete after this step.

2.6 Agent-based Simulations

This section presents a literature review of some of the research in agent-based simulation. An agent-based simulation is done by defining appropriate agents [77] and defining how they can interact with each other as well as with the environment. Agents in agent-based simulations range from simple reactive agents to complex, cognitive agents [78]. This type of simulation is adequate for a wide-range of areas and is interesting for its ability to generate emergent behaviour and complex relationships [77].

As RSNs are composed of multiple nodes, agents in agent-based simulations naturally become candidates for agents. This allows for the behaviour of each agent to be defined, and allows complex simulations to happen by only simulating the data for any given node's sensors. Such a simulation has been effective at simulating anomalies such as maritime pirates in order to detect them in [79], simulating events such as evacuations of multi-story buildings [80], or simulating financial markets [81]. Agent-based simulations have also seen previous use in WSNs; the authors of [82] present a formally defined agent-based simulation framework for WSNs for complex adaptive environments.

2.7 Chapter Summary

This chapter reviewed some of the fundamental concepts that is visited in this thesis, namely WSANs/RSNs, CI, MOO, the use of CI in WSANs/RSNs, the RMF, and agent-based simulations. These can grouped into the three major research streams that are crucial to this thesis, WSANs/RSNs, CI, and the RMF. Agent-based simulations are presented since this type of simulation is used extensively to simulate scenarios and validate the proposed additions to the RMF.

The first stream is that of WSANs/RSNs. This section presents concepts and research for WSNs, WSANs, and RSNs that demonstrates the multitude of problems in these networks, as well as their applications. WSANs serve as the backbone for many of the applications in this thesis.

The second stream comprises CI and MOO. The section on CI presented an in-depth literature review of CI definitions, concepts, and existing techniques that detailed the five major components of CI: evolutionary algorithms, swarm algorithms, fuzzy logic, learning systems, and hybrid systems. CI serves as the core of the developments in the thesis. A smaller section is dedicated to MOO, which is another recurrent theme.

The third stream is that of the CI in WSANs/RSNs and the RMF. The section on CI in WSANs/RSNs presents many example applications of CI to resolve the problems in WSANs. Prior research on the RMF is then presented, with some of this research being applications of CI in WSANs. This research stream bridges over the CI and WSANs. The RMF is central to risk identification, detection, and mitigation, the major theme of this thesis.

While the RMF's application methodology may be unspecific to any application, it still lacks certain abilities that would augment many aspects of it. The RMF's design itself is not suitable for networks such as the IoT, something that will be described in the next chapter. The work on using the RMF for CIP with RSNs is incomplete in the sense that the response generation is not as optimized as it could be. The RMF does not fully leverage specialized robotic nodes to increase its own monitoring

abilities. Other research possibilities exist in the response generation module, such as new algorithms that are more suited to the problems of situational awareness and DSSs. The next chapter will present a novel system architecture for the RMF along with four additions each designed, simulated and tested to solve some of these problems.

3 RMF and Modifications

Chapter 3 explains the five major contributions as presented in the first chapter. There have been multiple implementations of the RMF, yet none have been designed for the asynchronous and volatile nature of the data generated by the IoT, or designed to scale to the amount of the possibly tremendous amount of data emanating from such networks. Consequently, a reactive and distributed version of the RMF is explained. Furthermore, modifications of the RMF for hybrid RSNs are presented, as well as three modifications to the CoA module of the RMF: (1), alternative auction protocols for better MRTA allocations; (2), continuous response generation, to allow for solutions that adapt as the environment changes; and (3) the use of an alternative, novel CI technique for MOO to solve a routing problem.

3.1 Reactive Risk Management Framework

This section will introduce the modifications to the RMF's basic architecture in order to leverage the IoT's asynchronicity and unreliability. The RMF as presented in previous works [74, 75, 13] executes as follows; Risk features are extracted from sensor-provided raw data and assessed by an FIS in order to give a numeric risk value. Whenever this risk crosses a threshold, an operator is presented with Pareto-optimal risk mitigating CoAs. The operator may then choose one to reduce the risk.

The RMF architecture was designed for robotic sensor nodes where any given node might have up to a dozen sensors, and it consequently does not scale properly. Furthermore, the user is not informed of problems until the mitigating actions are pre-

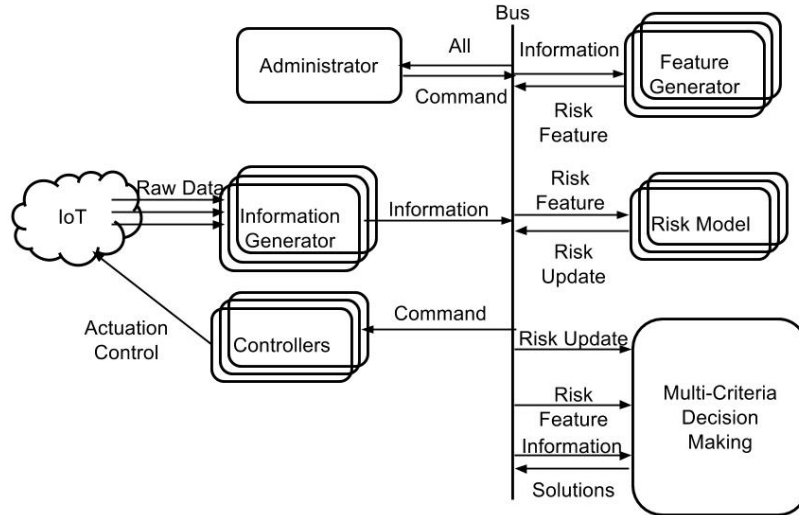


Figure 3.1: Reactive Risk Management Framework

sented. The MRTA process is not always quick due to the solution space complexity, so the operator may want to be aware of the current state of the system. The sequential design of the architecture leaves all components fundamentally reliant on previous components to function correctly. Without all risk features, the risk model may not determine appropriate risk values.

A modular, distributed design for the RMF is presented in Figure 3.1. The design separates existing components into individual parts that can be added or removed to accomplish goals and increase functionality. The bus-based architecture allows information to be reused and foster greater transparency. Bus-based communication is an abstraction of a communication system that provides many-to-many communication through a publish/subscribe model [83]. This decouples the communication medium from the component's function, avoiding the necessity to plan for shortcomings in the communication system. As communication in distributed systems is an active research field, this leads to simpler system designs.

3.2 Auction Protocols for Multi-Robot Task Allocation

This section first presents two novel auction protocols for the selection of viable candidates for multi-robot task allocation (MRTA), as well as proposes a task valuation

system that is common to both protocols. MRTA is the optimization process by which multiple capable agents are appropriately allocated tasks [84] where an efficient allocation minimizes the resource utilization while maximizing the impact of the tasks. The MRTA process is a core component of RSNs as it serves as the fundamental mechanism that enables coordinated actuation. When utilized in RSNs, MRTA problems often operate with only partial understanding of the environment and requires swift resolution.

A previous work in risk-aware RSNs for CIP [75] has used a single-task multi-robot instantaneous assignment (STMRIA) [23] to allocate risk mitigating tasks to sets of robotic nodes as a mechanism to manage risky events. A set of robots considered as candidates for the task allocation process is referred to as the coalition. This coalition must first be identified in no trivial way since an appropriate coalition will help the MRTA process converge faster on an optimized task allocation.

Market-based techniques [85, 86] to determine coalitions through emergent behaviour have proven to be a most suitable choice to create these coalitions. Auctions for membership in the coalition in which agents bid based on their ability to accept new tasks can be used, resulting in only the most apt agents being selected. These bids can be computed from an agent's knowledge of its resource, position, or any other available information.

The RMF as applied for risk-aware RSNs in [75] has used the First Price Sealed Bid (FPSB) [87] auction, in which all participants submit a single bid with each one being unaware of other bids. However, other auction schemes such as the Dutch, Japanese, and English auction [87] exist in reality and can be used as inspiration for new auction algorithms. The real protocols are illustrated in Figure 3.2. The authors of [88] describe how a competitive mechanism such as auctions with selfish agents can result in cooperative behavior by carefully defining value. To leverage competitive auction protocols, a task valuation model must first be introduced to allow agents the ability to assign value to tasks.

The value of task valuation model is risk mitigation, that is, how much an action

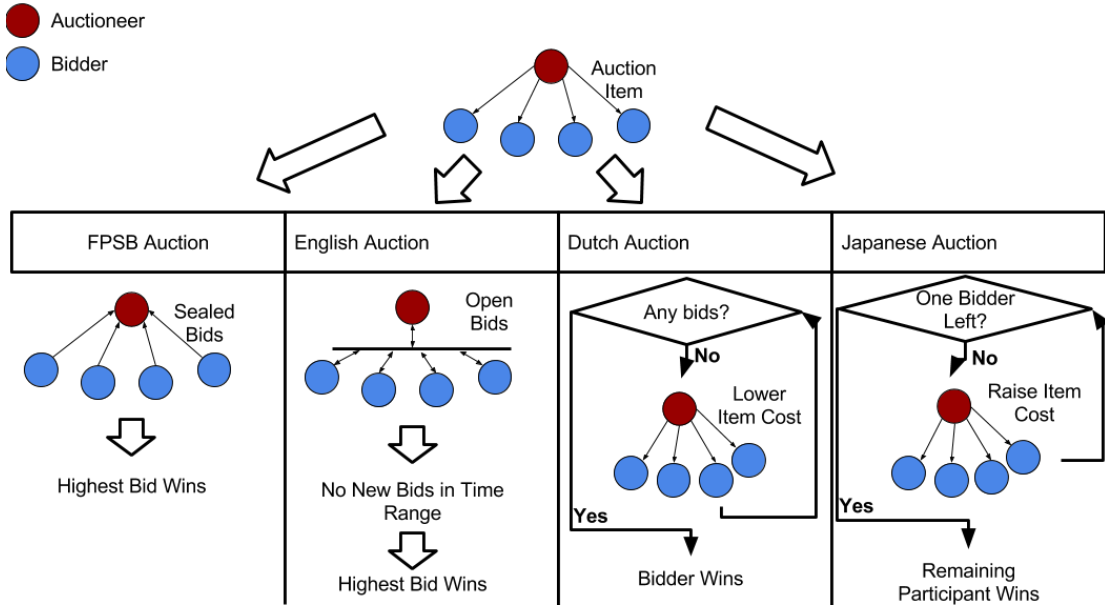


Figure 3.2: Illustration of FPSB, English, Dutch, and Japanese Auction Protocols

leads to a reduction in risk. This value is a representation of risk mitigation and is fundamentally unlimited in value. The gain is the difference between the value of the agent's state resulting from accepting a task and current value of the agent's state. The cost is computed as the energy needed to accomplish a task, and any other loss. The profit of accomplishing a task is then defined as $Profit = Gain - Cost$.

The first protocol is the FPSB protocol. This protocol attempts a simple optimization which does not allow agents to reason on their ability to join the coalition relative to others. It additionally does not take into consideration the concept of task value introduced earlier. In this protocol, each agent computes its own bid based on its ability and resources when receiving a new task announcement. These bids are sent to an auctioneer, a unique role played by one node that does not preclude a node from placing a bid. The winners of this protocol are those that placed the highest bid. This protocol is described in Algorithm 3.1.

Algorithm 3.1 First-Price Sealed Bid Protocol for Auctioneer

Broadcast Auction
 Bids \leftarrow collect Bids until timeout
 Coalition \leftarrow highest bids, $|Coalition| \leq \text{Maximum}$

The proposed Dutch-Japanese Auction Protocol takes inspiration from the Japanese

and the Dutch auction protocols [87], and is the first contributed auction protocol. These are both non-bidding auction protocols. Japanese auctions work by increasing the auction item value progressively. At each step, the participants indicate if they are willing to pay the new value. This process continues until there is only one remaining participant. In the Dutch protocol, the price is set relatively high, only to be gradually decreased until one participant claims the auction. It can be seen as the opposite of the Japanese auction.

To converge on an appropriate coalition size, the proposed auction protocol starts the bidding threshold at the estimated task value. Then, depending on the response, this threshold is raised or reduced by 50%. If the size is still unacceptable for the same reason as in the previous round, the threshold is further modified by 50%. If the size has grown too big or too small after this modification, then the threshold is augmented or diminished by 50%, i.e. modified by a factor of 2. This will continue until a coalition size within the minimum and maximum size is found.

Agents can dynamically determine the gain in value resulting from joining the coalition on each message round, which contain the set of currently winning agents. For example, in RSNs for CIP, if neighbouring agents are participating in the coalition, it is possible that some of these agents will get relocated, augmenting the value of participation for the remaining agents since they can get relocated to a close position with less redundant coverage. This allows the MRTA optimizer to fix any hole resulting from moving agents. If an agent determines that the task value weighted with its ability to complete the task, plus the additional value resulting from neighbour participation is greater than the value of their current state, they will indicate they are willing to participate, else they send their ability only. With this behavior, the auction defaults to a FPSB auction if it cannot find a better coalition. The protocol for the auctioneer is presented in Algorithm 3.2.

The proposed English Auction Protocol, the second proposed auction protocol, uses the widely known English auction which works in the following way: A starting value is first announced, often relatively low compared to the actual value of the auction

Algorithm 3.2 Dutch-Japanese Protocol for Auctioneer

```

Modifier  $\leftarrow$  0.5
Threshold  $\leftarrow$  Task Value
Previous  $\leftarrow$  Small
while  $|Bids|$  not in range and not Auction Timeout do
  Broadcast Auction
  Bids  $\leftarrow$  collect Bids until timeout
  if  $|Bids| < Minimum$  then
    if Previous is Big then
      Modifier  $\leftarrow$  Modifier / 2
    end if
    Threshold  $\leftarrow$  Threshold * (1 - Modifier)
    Previous  $\leftarrow$  Small
  end if
  if  $|Bids| > Maximum$  then
    if Previous is Small then
      Modifier  $\leftarrow$  Modifier / 2
    end if
    Threshold  $\leftarrow$  Threshold * (1 + Modifier)
    Previous  $\leftarrow$  Big
  end if
end while
Coalition  $\leftarrow$  highest bids,  $|Coalition| \leq$  Maximum
  
```

item. Then, bids are made by different participants in a monotonically increasing way. When no other participant is willing to outbid the highest bid, the auction ends, and the highest bid wins. The proposed English protocol is the following: The minimum bid is announced at 0, then the auctioneer listens for bids and periodically broadcasts an auction status update, which includes this minimum bid. Agents then adjust their own bids depending on this information and send it to the auctioneer. Agents may only bid higher than the minimum bid. If no new bids are received within a certain time, the auctioneer ends the auction and the bidders above the minimum bid form the coalition. The protocol is presented in Algorithm 3.3.

3.3 RMF For Hybrid Robotic Sensor Networks

This section will propose a methodology to integrate UAVs into RSNs by formulating the problem in the context of risk management. The RMF of [13] uses ground-based

Algorithm 3.3 English Protocol for Auctioneer

```

Threshold  $\leftarrow$  0
Round Timer  $\leftarrow$  0
Bid List  $\leftarrow$   $\emptyset$ 
Broadcast Auction
while Round Timer not done and not Auction Timeout do
  if bid received then
    Remove  $B \in$  Bid List |  $B.sender$  is  $bid.sender$ 
    Add bid to Bid List
    Round Timer  $\leftarrow$  0
    if  $|Bids\ above\ threshold| > Maximum$  then
      Threshold  $\leftarrow$  value of cut-off bid
    end if
  end if
  if Round Timer /2 elapsed then Broadcast Auction
  end if
end while
Coalition  $\leftarrow$  highest bids,  $|Coalition| \leq Maximum$ 

```

RSNs. These nodes usually have longer lifespans and are capable of more robust mitigation plans, however, lack the ability to monitor expansive areas, and the opposite is typically true for their aerial counterparts. There is synergy by combining both. The ability to cue the ground network of any developing risky situations is advantageous since it results in a more efficient MRTA in terms of resource utilization and risk mitigation. Hence, a need exists to properly integrate UAVs into RSNs.

The aerial nodes can be redirected to monitor specific areas with two steps. The first step is to define a set of risk features and an FIS that can detect situations that warrants greater attention. These features work at a coarse level so they should try to be characteristic of situations that might be risky but not strive to be overly accurate. The purpose of this step is to guide the aerial sensor network (ASN) to focus on certain areas, since it is inefficient to keep track of all of the AoI of the RSN.

The second step is to define the MRTA process that of monitoring tasks to ASN nodes for certain regions of the AoI when an event requires extra attention. The AoI can be divided into a grid where each cell can be monitored by a UAV. A monitoring task is defined as the allocation of cells to UAVS for them to monitor. The mobility of UAVs can be leveraged by allocating a sequence of cells to a UAV in order for it to monitor

	Gene 1	Gene 2	Gene 3	...
Layer 1	Enabled?	Enabled?	Enabled?	...
Layer 2	Target Cell	Target Cell	Target Cell	...
Layer 3	Target Cell	Target Cell	Target Cell	...
Layer 4	Target Cell	Target Cell	Target Cell	...

Figure 3.3: Encoding of MRTA Solution for UAV Monitoring

multiply-connected regions. This MRTA process will utilize the same multi-objective genetic algorithm approach as used in other MRTA processes [8]. As such, proper data structures, operators, and fitness functions must be defined.

Figure 3.3 presents the chromosome that is used. This chromosome is an encoding of a task allocation solution. Each gene, or element of the chromosome, corresponds to a UAV and can either be enabled or disabled, determining if the UAV is assigned tasks or not. It has a sequence of cells to monitor for the UAV. A mutation operator for this chromosome is defined as follows.

$$C = \{Cell_{0,0}, Cell_{0,1}, \dots, Cell_{n-1,n-1}\} \quad (3.1)$$

$$A = \{0, 1\}, W = f_P(C) \quad (3.2)$$

Where A is either 0 or 1, C is the set of all cells in the AoI, W is the set of cells that require monitoring, and f_P is a function yielding the cells that require monitoring. A gene can be described as follows.

$$Gene = \begin{bmatrix} a \in A \\ w_1 \in C \\ w_2 \in C \\ w_3 \in C \end{bmatrix} \quad (3.3)$$

Where a waypoint, w_i , constitutes the center of a cell and a deadline in time to reach this location. Too many planned waypoints constrain the ASN for too long, while few layers result in repeated costly MRTA processes. A limit of three waypoints

has empirically proved to strike a fair balance between these two constraints. The mutation operation can be defined as follows, where Gene' is the mutated gene.

$$Gene' = \begin{bmatrix} P_A \\ P_W \\ P_W \\ P_W \end{bmatrix} \quad (3.4)$$

$$P_A = \begin{cases} 0.5 & \text{if } a = 0 \\ 0.5 & \text{else} \end{cases} \quad (3.5)$$

$$P_W = \begin{cases} 1/|W| & w \mid w \in W \\ 0 & \text{else} \end{cases} \quad (3.6)$$

The one-point crossover operator is used [89]. Three fitness functions are minimized. The first evaluates the energy needs of the chromosome and is presented in Algorithm 3.4.

Algorithm 3.4 Resource Fitness Function

```

Resources ← 0
for all Gene | Gene.a = 1, Gene ∈ Chromosome do
  Segment1 ← Distance (UAVi, Gene.w1)
  Segment2 ← Distance (Gene.w1, Gene.w2)
  Segment3 ← Distance (Gene.w2, Gene.w3)
  Path ← Segment1 + Segment2 + Segment3
  Resources ← Resources + UAVi.efficiency * Path
end for
Return Resources

```

The second fitness function measures network connectivity and is presented in Algorithm 3.5. It uses the k-connectivity metric, defined in Algorithm 3 of [90], that measures the number of alternative communication paths around a certain node.

The third and final fitness function, Algorithm 3.6, measures the relevancy of the paths over the projected area of the tracked object. The function f_p returns the cells that are intersected by traveling in a straight line between two points.

Algorithm 3.5 Connectivity Fitness Function

```

Redundancy  $\leftarrow$  0
for all Cell Layer in Gene do
  for all  $Gene | Gene \in Chromosome$  do
    Metric  $\leftarrow$  k-redundancy (UAVi, Cell Layer)
    Redundancy  $\leftarrow$  Redundancy + Metric
  end for
end for
Return -1 * Redundancy
  
```

Algorithm 3.6 Relevancy Fitness Function

```

Path  $\leftarrow$   $\emptyset$ 
for all  $Gene | Gene.a = 1, Gene \in Chromosome$  do
   $S_1 \leftarrow \{ Cell | Cell \in f_P(C, Gene.uav.position, Gene_i.w_1) \}$ 
   $S_2 \leftarrow \{ Cell | Cell \in f_P(C, Gene_i.w_1, Gene_i.w_2) \}$ 
   $S_3 \leftarrow \{ Cell | Cell \in f_P(C, Gene_i.w_2, Gene_i.w_3) \}$ 
  Path  $\leftarrow Path \cup S_1 \cup S_2 \cup S_3$ 
end for
Covered Cells  $\leftarrow \{ Cell | Cell \in Path, Cell \in W \}$ 
Return  $-1 * |Covered| / |W|$ 
  
```

3.4 Continuous Risk-Aware Response Generation

This section presents a modified MOEA algorithm for the generation of CoAs through multi-criterion decision analysis (MCDA) which will enable more relevant and adapted solutions pertaining to the solution space as bounded by the state of the most up to date perception of the environment. The use of continuous here refers to an algorithm that does not terminate, meaning that it runs uninterrupted, or continuously. It will also present a fitness function to improve results.

Traditional MOEAs are applied to static solution spaces, where the fitness functions are well defined. If appropriate termination conditions are chosen, solutions generally become better and are often near optimal. Due to the inherent stochastic nature of EAs, precise termination conditions that would always result in globally optimal solutions are impossible to find. It is not uncommon and is appropriate to determine the termination condition experimentally.

As the environment in which the RMF works is fundamentally dynamic, the CoA generation module should not make the assumption that the optimization problem it

is trying to solve is static. This leads to a conflict between the solution generation technique, the MOEA, and the requirements of the system. However, it can be seen that the MOEA does not necessarily need to have a static optimization problem and already has the tools to accommodate a non-terminating execution. Pareto-optimal solutions can be viewed as the result of a stochastic function over the solution space resulting from the environment's state. As this state changes, so does the solution space and the potential solutions.

Executing a new MOEA whenever there is a change in the solution space is costly, inefficient, and even ultimately unnecessary. The fitness of solutions which mitigate the risk in certain environmental states are similar to the fitness of solutions derived from nearby environmental states in time if those two states are correlated. The assumption that two nearby environmental states are correlated is fair due to causality; a storm does not instantaneously appear, weather sensors will detect it as it is incoming and indicators will change over time. This requires that the environment changes at a slower pace than the speed at which solutions can adapt.

The convergence of solutions in EAs is highly susceptible to their initial population: if the initial population is close to the Pareto-optimal front, then the algorithm will converge quickly. These two observations suggest that if there would be a new execution for every environment state change, the previous execution solutions should be used as the starting population. Additionally, as soon as the environment's sensed state changes, any previous MOEA executions are likely now obsolete. In situations where the state changes are completely uncontrolled, this might result in a situation where no termination conditions are ever reached. This requires the MOEA to be able to adapt to the state changes as they happen without ever terminating. In other words, this requires the removal of all termination conditions.

The preceding is shown in Equations 3.7 to 3.10. f_{Space} is a function which bounds the solution space per the state of the environment, f_{EA} is the MOEA, providing solutions for a particular optimization problem derived from the state of the environment. f_{fitness} is a vector of fitness functions, while R_{StateEnv} and R_{Fitness} are the autocorrelation

functions for the state of the environment and the fitness values returned from the fitness functions, respectively.

$$SolutionSpace(t_i) = f_{Space}(State_{Env}(t_i)) \quad (3.7)$$

$$Solutions(t_i) = f_{EA}(SolutionSpace(t_i)) \quad (3.8)$$

$$Fitness(t_i) = f_{fitness}(Solutions(t_i)) \quad (3.9)$$

$$R_{State_{Env}}(t_i, t_{i+1}) > 0 \Rightarrow R_{Fitness}(t_i, t_{i+1}) > 0 \quad (3.10)$$

MOEAs evaluate solutions via fitness functions. If the fitness functions use the most recent data available, then they may evaluate solutions as they pertain to the current solution space and environment. This puts continuous pressure on the population to adapt as the algorithm attempts to converge to a slowly changing Pareto-optimal front. Since the algorithm does not terminate, sampling the population becomes appropriate.

Since solutions that are more suited for multiple environment states might be able to adapt easily to other specific cases, a special fitness function that measures the resiliency of a solution in the population can be used. The *Resiliency* fitness function attempts to determine the generality of a solution over time in light of the dynamic fitness functions. It works by keeping track of the number of generations a solution survives and normalizing over a Max parameter that can be seen as a weight for the fitness function: Lowering it will result in heavy bias towards solutions surviving a single iteration, while increasing it reduces the importance of resiliency. This fitness function should be minimized.

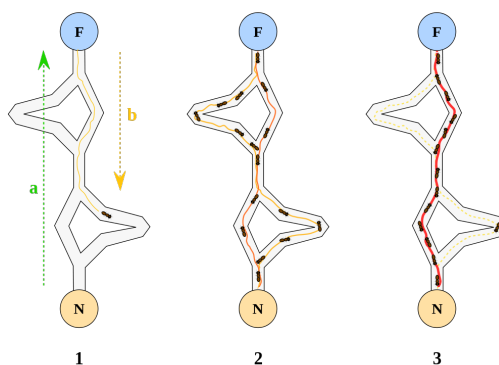


Figure 3.4: Example of the foraging behaviour of ants ¹

3.5 Multi-Objective Ant Colony Optimization

This section will present a novel multi-objective ant colony optimization (ACO) algorithm that can be used for combinatorial optimizations, such as the MRTA problem that is often but not always found in the CoA module of RMF. It combines many of the parts of other MO-ACO as described in [91], but presents a novel visitation rule that does not use one of the categories defined in the taxonomy. An application for a problem other than MRTA is found later in this thesis. The ACO algorithm is part of the meta-heuristic branch of algorithms, a class of algorithms that are generally suitable to soft-solve a wide variety of problems. The algorithm works by simulating the distributed foraging behaviour of ants. This is exemplified in Figure 3.4.

The problem is decomposed into a graph, where a sequence of nodes and edges represents a solution. Weighted edges between nodes indicate links between components in a solution. The ACO algorithm can then be applied. The goal of the algorithm is to discover the less costly solution that satisfies the given problem by dropping pheromones on edges indicating their suitability and overcoming the limitations of greedy approaches, primarily that of short-term optimal decisions that do not result in global optimality.

An iteration in this algorithm starts with the simulation of virtual ants. Practically speaking, this means a number of solutions are created by creating sequences of

¹Dréo, Johann . Shortest path find by an ant colony. Digital image. N.p., 27 May 2006. Web. 27 July 2017.

nodes/edges by using a visitation rule. When at a given node, the next node to visit is decided on the weight of the edge, and on the pheromones that were dropped on the edge. This action of checking the weight of an edge is called visibility. Nodes are visited until a certain stop criterion is met; criteria can include having reached a target node such as in the shortest path problem, or having visited all nodes, such as in the Travelling Salesman Problem (TSP).

The fitness of the generated solutions is computed. Generally, the fitness is simply the sum of all weights associated with the edges, though other fitness measures can be used. Pheromone values are computed from the fitness of the solutions and dropped on the edges that form the solution. A certain evaporation rate is set so that the level of pheromones is reduced uniformly over time so they do not build up to infinity. With this update, ants of the next iteration can decide to travel more promising edges with lots of pheromones, thus overcoming the limitations of the greedy algorithm.

The Ant System [92] is the original algorithm while the Ant Colony System [93] and Max-Min Ant System [94] are two famous variations. The visitation rule of the Ant System algorithm and the pheromone update rules are given in the following two equations.

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)^\alpha \times \eta_{ij}^\beta}{\sum_{\ell \in \nu N} \tau_{ij}(t)^\alpha \times \eta_{ij}^\beta} \quad (3.11)$$

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (3.12)$$

The variable $\tau_{i,j}$ is the pheromone associated with an edge, $\eta_{i,j}$ is the visibility of the ant, or the cost associated with the edge leading to the other node, α and β are weights for the visibility and pheromones. This value is normalized over the values for all other nodes so that the probabilities sum to one. The parameter ρ is the pheromone evaporation rate and is in the range of 0 to 1. Essentially, the current pheromones on the given path are reduced by a certain proportion. The second part is the sum of all pheromones dropped on the edge, computed as a function of the

fitness of a solution.

Many different multi-objective versions of the ACO algorithms exist [95] but many recurrent components are used, a fact that was used to create the taxonomy of [91]. The following novel algorithm can be characterized as a multi-pheromone structure, with Pareto evaluation of solutions, global pheromone updates, and an online archive. The solution construction method compares solutions using Pareto-based method, something that is novel as it is not present in the taxonomy.

The node visitation rule is a combination of the method in [96] for multi-objective Pareto Q-learning, the SPEA-II [97] selection operator, and the visitation rule of the ACS. Similarly to ACS, edges are sorted by Pareto rank per their visible costs with a probability of q_0 and then sorted per their probabilities with a probability of $1-q_0$. The nodes are sorted into their Pareto fronts, with no clear best edge between nodes within a front. The same solution selection operator that selects better solutions from the population as found in SPEA-II is used to select the next node to visit. This is illustrated in Figure 3.5 and the full solution construction algorithm is given in Algorithm 3.7, where the function gP returns pheromones given a pheromone matrix ρ , and an edge, and the function eW returns the weights for a given edge between node n and j . The function sP sets the pheromones for a given edge, given the pheromone matrix ρ , an edge e , and new pheromone values.

Algorithm 3.7 MO-ACO Solution Construction

Require: Node n , Graph g , Pheromone Matrix $\vec{\rho}$, α , β , q_0
 path $\leftarrow \{n\}$
while Termination conditions not met **do**
 $\vec{w} \leftarrow \{\}$
 if $X \sim U(0,1) \leq q_0$ **then**
 $\vec{w}_j \leftarrow e\vec{W}(n, j) \forall j \in neighbours(g, n)$
 else
 $\vec{w}_j \leftarrow g\vec{P}(\vec{\rho}, edge(n, j))^\alpha \cdot e\vec{W}(n, j)^\beta \forall j \in neighbours(g, n)$
 end if
 next node $\leftarrow ParetoNodeSelectionOperator(\vec{w})$
 path \leftarrow path + next node
end while **return** path

This algorithm uses the pheromone matrix for each objective method. This means

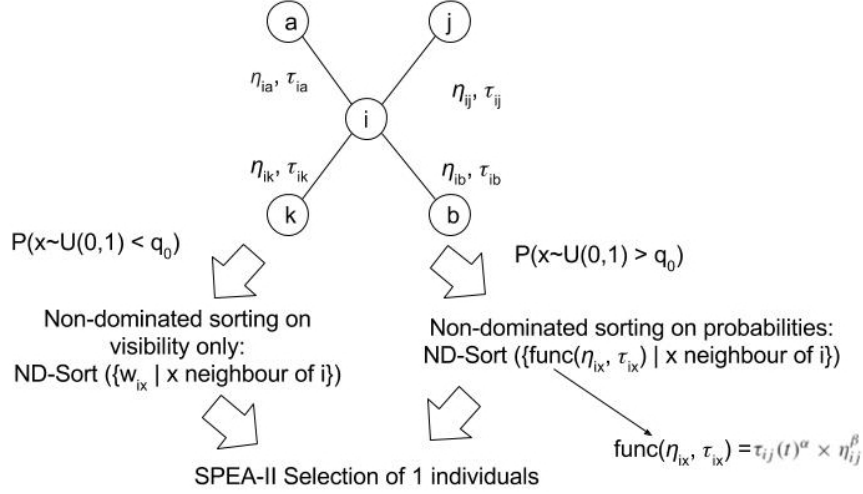


Figure 3.5: Visitation Rule Algorithm

that each edge has a corresponding vector of weights and pheromones. The pheromone update rule is the same as in the Max/Min Ant System, repeated with the individual fitness of each objectives. Only the first front of the archived solutions is used to update pheromones. The algorithm for the pheromone updates is given in Algorithm 3.8. Finally, the algorithm for the entire method is given in Algorithm 3.9.

Algorithm 3.8 Pheromone Update Procedure

Require: Graph g , Solutions $sols$, Pheromone Matrix $\vec{\rho}$, \vec{max} , \vec{min} , evaporation rate

```

updated edges  $\leftarrow \{\}$ 
for Solution  $s \in sols$  do
    for Edge  $e \in solution$  do
        updated edges  $\leftarrow updated\ edges + s$ 
         $\vec{cost} \leftarrow evaluateFitness(e)$ 
         $\vec{pheromones} \leftarrow max((1/cost + g\vec{P}(\vec{\rho}, e), \vec{max})$ 
         $sP(\vec{\rho}, e, \vec{pheromones})$ 
    end for
end for
for Edge  $e \in updated\ edges$  do
     $sP(\vec{\rho}, e, (1 - evaporationrate) * g\vec{P}(\vec{\rho}, e))$ 
end for
return  $\vec{\rho}$ 
    
```

Algorithm 3.9 MO-ACO Algorithm

Require: *Graph g , Starting Position s , Number of Ants $numAnts$, α , β , max , min , evaporation rate*
archive $\leftarrow \{\}$
 $\vec{\rho} \leftarrow \{\}$
 $sP(\vec{\rho}, edge, 0) \forall edge \in g$
while Termination conditions not met **do**
 generation $\leftarrow \{\}$
 while $|generation| < numAnts$ **do**
 generation $\leftarrow generation + MOACO\ Solution\ Construction(s, g, \vec{\rho}, \alpha, \beta)$
 end while
 archive \leftarrow archive + generation
 archive \leftarrow nonDominatedSorting(archive)
 archive \leftarrow truncate(archive)
 Pherome Update Procedure(g , nonDominatedSolutions(archive), $\vec{\rho}$, max , min , evaporation rate)
end while **return** archive

3.6 Chapter Summary

This chapter has presented a methodology for applying the RMF, a distributed design of the framework for the IoT, and multiple modifications to the framework. The methodology is used repeatedly when applying the RMF in the rest of this thesis and results in successful applications for the presented scenarios. The distributed version of the RMF provides a reactive implementation that will prove crucial for application in the IoT domain. Finally, multiple additions were proposed.

The first is the use of alternative auction protocols to determine appropriate coalitions for the CoA module of the RMF when solving the MRTA problem. The second details a methodology to integrate UAVs into the RMF from a risk-based angle. The third presents a modification to the CoA module of the RMF that allows for continuous optimization of risk mitigating actions for dynamic perspectives of the environments. Finally, the fourth proposed modification presents a novel, alternative CI technique to use in the CoA module for MOO based on ACO. The next chapter will present four application contexts and scenarios suitable for the RMF. The methodology is applied and the proposed contributions are demonstrated when the application is appropriate.

4 Case Studies in Critical Infrastructure Protection

This fourth chapter will describe a few case studies in critical infrastructure protection (CIP) for which the methodology presented in the previous chapter will be partially or completely applied with the modifications proposed in the last chapter. The first case study involves the monitoring of a perimeter around an important structure or resource that must be guarded and monitored against possible intruders. The second case study explains concepts relevant to maritime smuggling in ports and surrounding areas, which are important infrastructures which must be protected, then present a simulated scenario of maritime smuggling. The third section studies supply chains in the maritime domain, the effect of disruptions on them, as well as a synthetically generated situation where a storm threatens to disrupt the maritime segment of a supply chain. The fourth and final section again involves disruptions to the maritime supply chain, but at the micro level by trying to prevent in from the perspective of the vessel. These two last case studies concentrate on supply chains, which are a critical economic infrastructure.

4.1 Critical Infrastructure Protection

The Presidential Decision Directive 63 for the United States Department of Defense

¹ states that CIP is the protection of certain national infrastructures, e.g., energy,

¹The Department of Defense Critical Infrastructure Protection (CIP) Plan. (1998, November 18). Retrieved July 18, 2017, from <https://fas.org/irp/offdocs/pdd/DOD-CIP-Plan.htm>

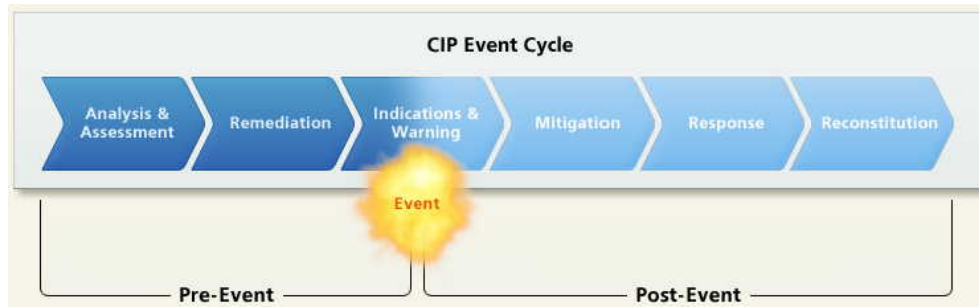


Figure 4.1: United States Department of Defense CIP Event Lifecycle ²

information and communications, and banking and finance, are critical to the national and economic security. It defines six steps of the CIP lifecycle. These are shown in Figure 4.1.

Analysis and assessment is first performed to determine which infrastructure is critical, its importance, and the effects of its loss. Remediation is done to fix weak points in the protection of the infrastructure by taking precautionary measures such as procedural changes, or system component changes. Then, at the indications and early warnings stage, the infrastructure is monitored for possible attacks or disruptions. Appropriate actions can be taken to mitigate the consequences of a compromised infrastructure if a damaging event is detected, an unauthorized intrusion of a secure perimeter for example. Other responses to the incident can then be enacted in order to prevent the breach of other system. The final stage is to repair the damaged infrastructure.

The monitoring of physical assets such as nuclear power plants, power dams, military installations, and other infrastructure is usually done with the help of networks of sensors, or WSNs. Consequently, the use of WSANs and RSNs here is more than suitable for the third stage of the CIP lifecycle, with the design of the monitoring network falling into the second stage of the CIP lifecycle. There are indeed many examples of applications of WSNs for CIP in literature, with some examples of applications in oil, gas, and water pipelines, telecommunication systems, and airports [98, 99, 100].

²CIP Event Cycle. (n.d.). Retrieved July 20, 2017, from <https://commons.wikimedia.org/wiki/File:CIP-chart1.jpg>

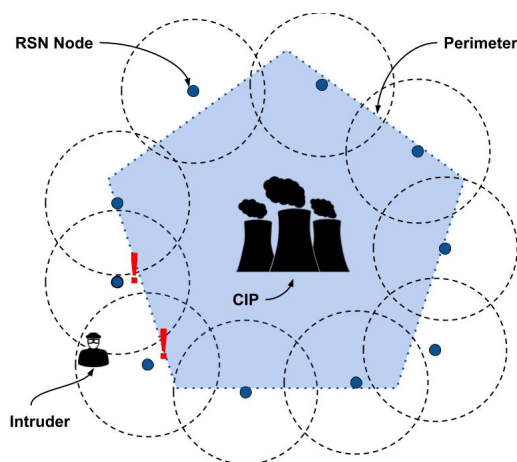


Figure 4.2: Example of RSN for CIP

A scenario similar to the one in [8] will be used to illustrate the application of the RMF. In this scenario, a perimeter must be protected around the infrastructure. Any intrusion on this perimeter must be detected, thus requiring constant monitoring over its entirety. An example of an RSN for CIP is shown in Figure 4.2. Consequently, an RSN with adequate sensors to detect intruders, as well as the capability to relocate themselves is deployed. The goal is to design a system based on the RSN that can maintain the perimeter coverage as the resources of the nodes start to run out, thus leading to node failures.

4.2 Maritime Smuggling

Maritime smuggling of weapons [101], illegal goods [102], or humans [103] is a real problem for ports around the world. Ports are gateways for the interior of countries and are major economic and sometimes military hubs. They often present the first line of defense against illegal activity that can cause risk for other inland resources. Consequently these critical infrastructures must be defended. A common smuggling tactic involves the use of a bigger vessel to transport the bulk of the smuggled goods, often titled a “Mother Ship”, that then meets with smaller vessels for distribution [104]. The meet-up event itself is often referred to as a rendez-vous or a cooperating event [104].

The inference of illicit activity or anomalous and possibly suspicious behaviour through gathered information has been researched in the past. For example, the authors of [79] use Hidden Markov Models (HMM) to identify maritime pirates, while [105] identifies five features, then use fused sensor data with a Bayesian Belief Network (BBN) to detect illicit activity.

The necessary information can be provided by monitoring a region for such behaviour with the use of assets that may need to move per the activity in the AoI. This problem is in fact well suited for the RSNs that are able to monitor portions of the AoI and relocate themselves to new areas. There has been a lot of research work in this area. The work in [106] optimizes the paths of surveillance assets to gather information on possible anomalous vessels, then use a BBN to identify anomalous behaviour such as piracy, illegal trafficking, terrorism, and other illicit activities.

The following details a scenario that involves maritime smuggling in an environment loosely based on the area near Barcelona, Spain, where a bigger vessel rendez-vous with three small boats to engage in smuggling activities. Many types of vessels exist in this scenario. Five mooring areas exist, though not every type of vessel can berth at each. The Port of Barcelona is presented as a mooring area for commercial and industrial vessels, as well as Port Vell, the civilian portion of the Port of Barcelona. Marina de Badalona, Port Olímpic, and Port Forùm are the last three mooring areas available, each corresponding to a local marina. The environment is shown in Figure 4.3.

The first vessel model is the merchant vessel (MV) that is equipped with an Automatic Identification System (AIS), and is used in commercial enterprises, such as moving cargo or commercial fishing. AIS is used to automatically send identification information and is often used to track ships or to coordinate their movement to avoid conflicts. MVs enter at certain points on the perimeter of the environment corresponding to high traffic lanes, head for the industrial port or Port Vell and leave after some time has elapsed. A MV may enter a state known as loitering, corresponding to real situations where vessels may loiter due to traffic, breakdowns, waiting for



Figure 4.3: Environment near Port of Barcelona

pilotage, etc.

The second vessel model is the ferry. Ferries enter at the same points as MVs and follow similar behaviour, except that Ferries always head to Port Vell instead.

The third vessel model is the large private vessel (LPV) that is required to broadcast AIS data, such as yachts. LPVs follow approximately the same behaviour as the two previous vessels, however they may enter the environment at any point, and may head to either Port Vell or Port Forùm, a nearby marina.

The small recreational vessel (SRV) is the fourth model and include vessels such as speed boats or sailing ships. It is not required to transmit AIS as it does not meet the IMO-mandated requirements. SRVs depart from either Port Vell, Port Olìmpic, Port Forùm or Marina de Badalona. SRVs depart with higher probability during the day towards one of many predesignated areas, and will navigate between these areas until they decide to return to their source port.

The last two models are the smugglers. The smuggling MV is based on the MV, and can participate in a rendez-vous with smuggling SRVs, but otherwise acts as a normal MV. It waits for the smuggling SRVs while in the rendez-vous, then continues towards the port. Smuggling SRVs act similarly to SRVs, but head directly for a Smuggling MV from any of the civilian ports, spend some time at the rendez-vous, then head

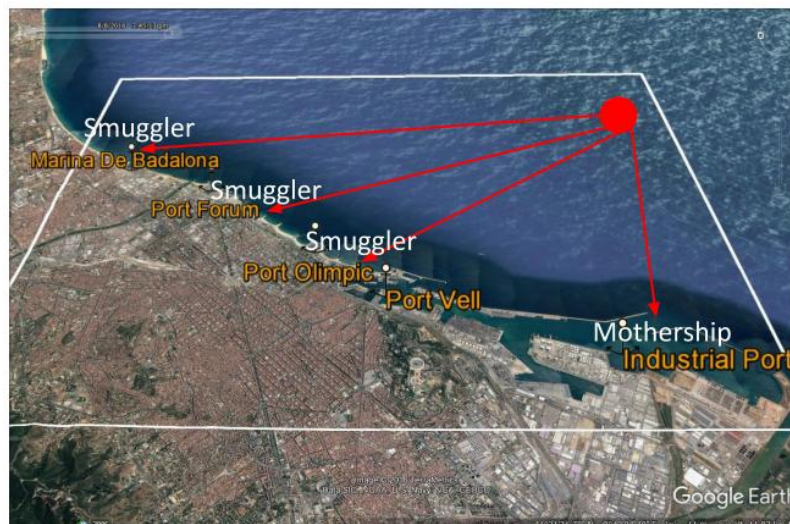


Figure 4.4: Maritime Smuggling Scenario

back to a civilian port.

A smuggling MV will arrive near the Port of Barcelona and loiter off the coast, only to rendez-vous with three smuggling SRVs. The smuggling MV will then continue towards the industrial port and the smuggling SRVs will head towards the Marina de Badalona, Port Olimpic, and Port Forùm. Many other types of vessels will be acting normally around the area. This scenario is illustrated in Figure 4.4.

An RSN can be designed to detect smuggling and take proper mitigating actions against it, thus reducing the consequences that result from smuggling activity. The system must be able to optimize its resources and avoid flagging non-smuggling vessels.

4.3 Maritime Supply Chains: Disruption Mitigation

Supply chains are a basic necessity of commercial entities in today's global markets. Be it for food, energy resources, commercial goods, or military supplies, supply chains are a crucial infrastructure that can wreak havoc on economic or military security. A supply chain is composed of segments and nodes, usually uniformly directed towards the endpoints at which goods arrive and leave the supply chain. Nodes correspond, for example, to factories or warehouses that may accept multiple resources and may

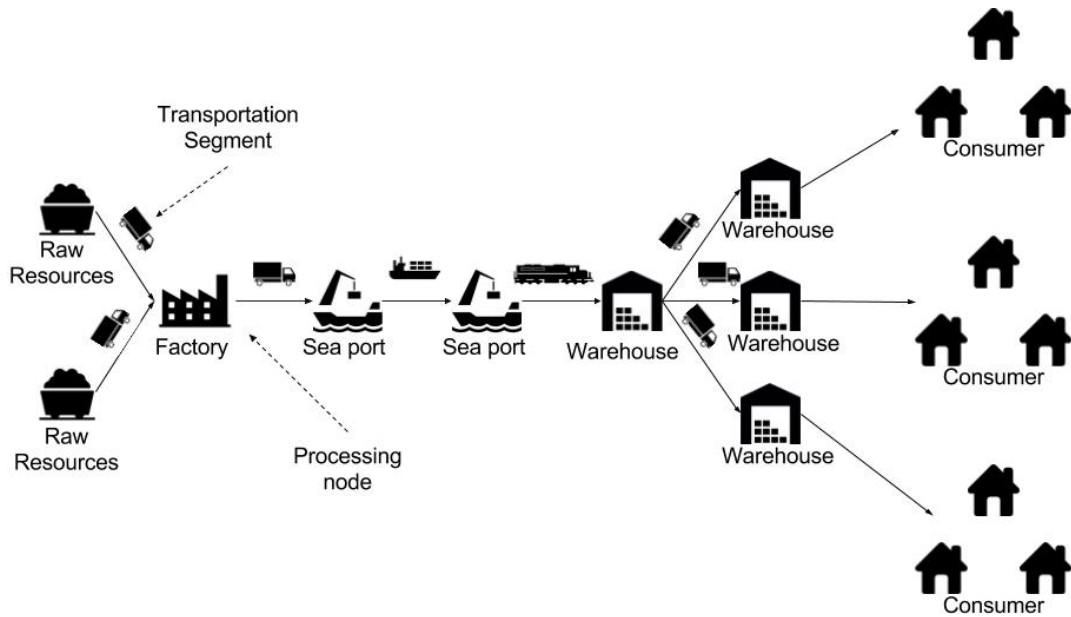


Figure 4.5: Example Supply Chain

output one or more products. Segments correspond to transportation links between these nodes, through one or many transportation modes. An example supply chain is given in Figure 4.5.

A disruption is any event that prevents a segment or node from completing its purpose. A factory node may face a labour dispute, breakdowns, or disasters such as fires [Chopra2004], to name a few. Segments are also susceptible to disruptions, such as natural disasters, border customs disputes, or quite simply breakdowns again. A disruption in the chain can have enormous impacts on subsequent downstream actors [107]. An example of a disruption is shown in Figure 4.6.

With recent globalization tendencies, supply chains have become increasingly extended to leverage regional advantages such as localized resources and labour [108], thus requiring extensive use of transportation modes. Due to the prevalence of oceans, freight ships are an ubiquitous and essential choice for transporting goods. However, this transportation mode has its own vulnerabilities [109]; transportation delays, breakdowns or traffic congestion should be predicted and their effects mitigated.

There has been some research into supply chain disruptions and methods to mitigate them, even suggesting that a system that is able to predict these disruptions

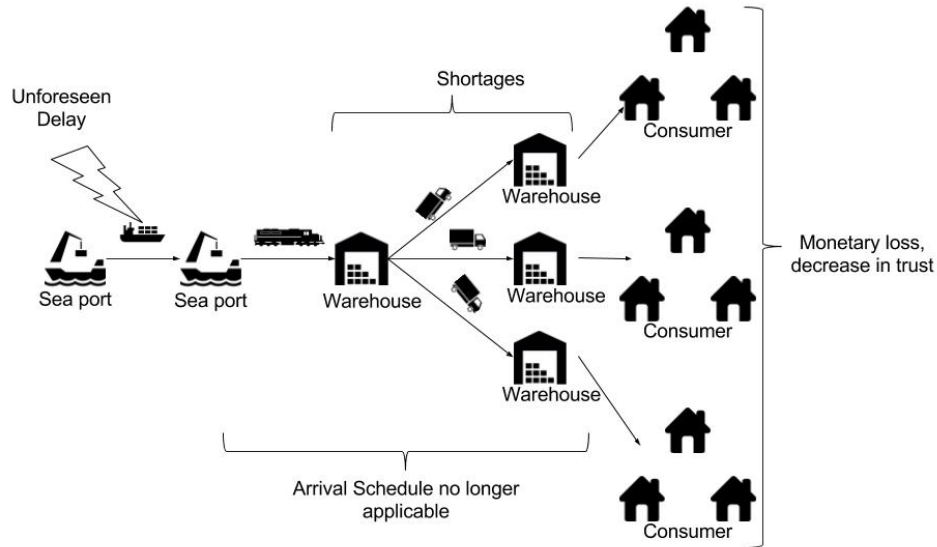


Figure 4.6: Consequences of a Disruption

and mitigate them would be invaluable. The authors of [110] present fundamental supply chain disruption concepts and mitigation strategies. They categorize sources of disruptions into natural disasters, human-related causes such as labour disputes and war, and manufacturing problems, among many other possibilities. They then present mitigation solutions and how they relate to each of those categories.

Similarly, the authors of [109] present effects of disruptions on supply chains and methods to create more resilient supply chains. They conclude by suggesting three avenues for future research, one of which is a recovery planning system that can mitigate supply chain disruptions as they occur. [111] presents a first attempt to develop such a system. The system aims to determine the probability of an event and its impact, then brings up some mitigating actions which can be selected. However, the system is entirely manual and relies on human labour.

The ability to predict disruptions requires enormous amount of information about the environment. The Internet of Things (IoT) is a new paradigm [112, 113] attempting to harness the wealth of information collected through any and all available sensors and remotely control any object that may have actuation capabilities. A system using the IoT must be able to handle the randomness inherent within the incoming information streams. The generated information is often unreliable or incomplete although there

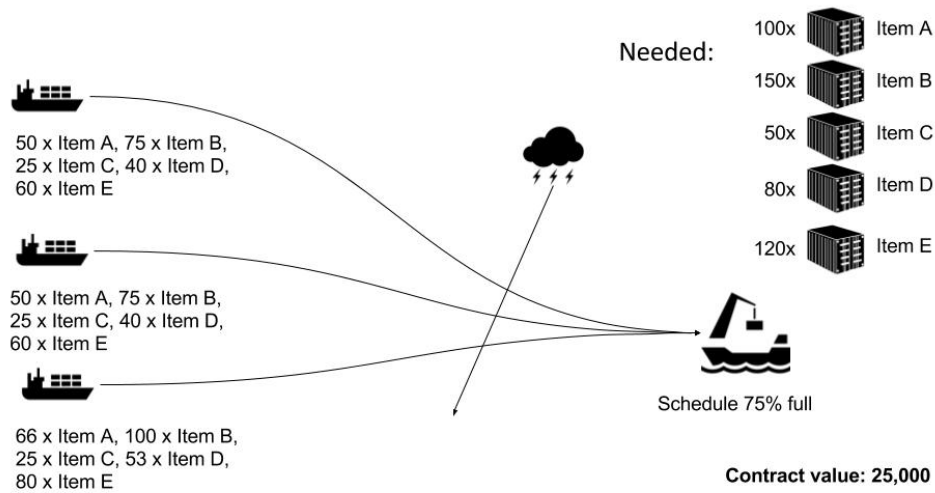


Figure 4.7: Example Disruption Scenario with Parameters

is typically a large volume of data being transmitted at a fast pace. The maritime world has not escaped this trend of ever increasing connectedness. By harnessing the information gathering capabilities of the so called maritime IoT (mIoT), the risk of a disruption could be predicted and/or tracked, while actions could be taken to mitigate it.

A scenario is synthetically generated with three cargo ships that are due at a port to deliver resources required for a contract. The first two generated ships are each carrying half the needed quantities of five commodities of varying costs for the contract, while the last one is transporting two thirds of the needed quantity for the contract. The ships are expected to dock two, three, and four days before the deadline. An example of this scenario with parameters is given in Figure 4.7.

The port where the ships are expected to arrive is fairly busy and so cannot fully promise to be able to accept the ships if they are delayed. The ships are at risk of crossing paths with a storm that would delay their arrival until after the time when the contract is due, resulting in an unfulfilled supply contract, and thus causing a disruption. There are additional providers available which can ship the needed resources on time for the contract at a certain cost, at varying degrees of risk. There are also other transportation modes available that are more costly, but also reduce risk.

The objective of this scenario is to design a system that is able to predict possible disruptions and determine solutions to mitigate the disruption. The system must be able to react swiftly and handle the flow of information generated by the mIoT.

4.4 Maritime Supply Chains: Disruption Prevention

With water covering roughly 71% of the earth's surface, maritime freighting is inevitable in today's globalized supply chains. Typical maritime freighting operations, transport cargo from one area of the world to another for further processing or for consumption. Current efforts are being made on streamlining freighting operations leaving them increasingly susceptible to disruptions, a major problem of supply chain management [114]. Weather events, breakdowns, and congestion are some of the many causes of supply chain disruptions.

As seen in the last section, the likelihood of a disruption and its effects should be respectively predicted, reduced, and mitigated. The solution to this problem has multiple facets, all of which fall within the domain of risk management. Situational awareness is paramount to all aspects of this problem, while MCDA is needed to choose context-appropriate actions.

Through the IoT and data collected by any means such as AIS, it is possible to determine the risk of certain events in some areas, such as weather events, collisions, piracy, and many others. With this information, a crowd-sourced risk heatmap can be built and utilized to allow vessels to avoid high-risk areas, thus reducing the risk of an event causing costly delays and possible disruptions in the supply chain.

This system approaches supply chain disruption mitigation the proactive perspective. Instead of accepting a disruption and then trying to react to them by finding appropriate mitigating CoAs, it tries to prevent the disruption by proactively changing the behaviour of the vessels. A combination of both mitigation and prevention would result in disruption prediction, prevention, and mitigation systems lying closer to what is proposed in [107, 109, 111].

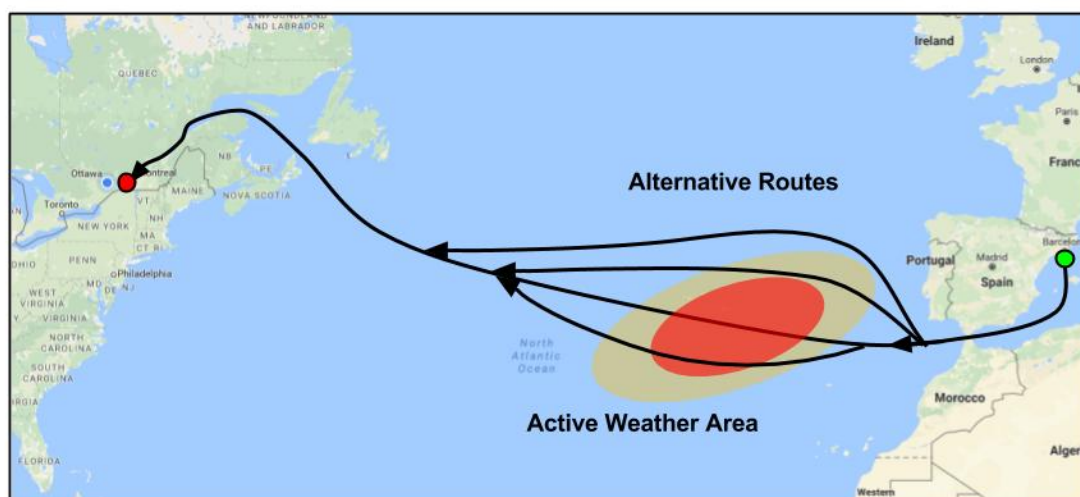


Figure 4.8: Example Scenario for Disruption Prevention

A scenario where a vessel travelling from one point to another is will travel through a region with a risk distribution and can follow one of many routes is envisioned. An example of this scenario providing possible alternative routes is shown in Figure 4.8. The goal is to design a system that can propose alternative routes for the vessel, ranging from lengthy, safer routes, to shorter yet riskier ones.

4.5 Chapter Summary

This chapter has presented four case studies warranting for the design and application of a system capable of fulfilling their objectives. The first scenario details the CIP problem and highlights RSNs as a viable solution. A system designed to be able to maintain perimeter coverage as nodes fail is needed. The second case study explores the world of maritime smuggling, and how to detect it via behaviour-based inferences. It requires that a system be designed to manage the assets that gather the data needed for the inferences, to detect maritime smuggling, and to mitigate it.

The third scenario explains supply chains, disruptions, and the efforts being made to predict and mitigate them. It depicts a scenario that demands a system be designed to predict and mitigate disruptions, with high-quality solutions and swift reactions. Finally, the last case-study explores supply chain disruptions from the proactive per-

spective, or how disruptions can be prevented by changing the behaviour of the vessel in a manner that may result in a controlled and smaller delay. A system that can propose different CoAs is needed in order to do this.

The next chapter will discuss the application of the methodology and additions proposed in the previous chapter for the scenarios seen in this one. The presented methodology for the RMF will be followed, and appropriate additions will be used to enhance the RMF and solve the problems that are found in the presented scenarios. Then, simulation study will be conducted and the results will be discussed.

5 Applications and Experimental Results

This chapter will follow the RMF methodology that was described in Chapter 3 and apply it for the case studies in critical infrastructure protection (CIP) that were described in Chapter 4. The first will be an application of the RMF to create a risk-aware robotic sensor network (RSN) to defend a perimeter around the CIP against possible intruders. The second section will describe an application of the methodology to detect and mitigate maritime smuggling. An application to predict and mitigate supply chain disruptions is presented in the third section, followed, in the last section, by an application that detects, and prevents disruptions before they impact the supply chain.

It will also present the results that were observed when running the simulations for each application. Section 5.1.3 will detail results for each auction protocol and then present relevant discussion and analysis of the observed metrics. Section 5.2.4 will present the simulation results and offer a step-by-step analysis of the behaviour of the hybrid RSN. The third section will present the observations for the Reactive Risk Management Framework (rRMF) with continuous optimization for supply chain disruption mitigation. Finally, the fourth and final section will present the results of the Multi-Objective Ant Colony Optimization (MO-ACO) algorithm for risk-aware ship routing to prevent supply chain disruptions, as well as the comparison with the three other algorithms under consideration.

5.1 Application for CIP Perimeter Coverage

The scenario described in Chapter 4.1 required a system capable of managing an RSN in order to maintain coverage of a perimeter surrounding a critical infrastructure. Such a network was designed in the work presented in [8]. The details of that application will be briefly summarised. In that work, market-based techniques [86] are introduced to either determine a proper coalition before the MRTA process, or afterwards when agents bid on the optimized tasks. Consequently, it is a great opportunity to apply, analyse, and compare the two auction protocols that were developed in Section 3.3.

5.1.1 Risk Model and Risk Features

The work in [8] describes an application of the RMF for CIP. This application uses a risk model for the *degree of distress* of a node, with three risk features. The first is the *battery level* of the node, the second is the *intrusion risk* or the distance at which a node detects intruders, and the third is the *terrain maneuverability* around a given node. These features are computed from self-reported sensor data or from the network's knowledge base, and then are fed into a fuzzy inference system (FIS). If the overall risk breaches a certain threshold, the node becomes a *node in distress* (NID).

5.1.2 Risk Mitigation

NIDs trigger an optimized MRTA process that generates and allocates risk mitigating tasks. During this MRTA process, the nodes are assigned relocation tasks in order to increase or restore perimeter coverage near a NID. This MRTA process optimizes the overall network allocation with respect to two conflicting objective functions. The first measures *perimeter coverage*, which must be maximized, while the second measures the *total energy expenses* of the nodes in a particular assignment solution,

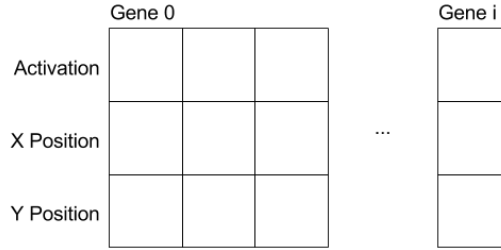


Figure 5.1: Chromosome for Repositioning

which must be minimized. An assignment solution is encoded via a chromosome, with each gene each corresponding to one of the nodes in the coalition. This is illustrated in Figure 5.1. Each gene has an activation layer and the destination of the node corresponding to a point on the perimeter. A one-point crossover operator and a uniform crossover operator are picked randomly, while a mutation operator can negate the activation layer or change the destination.

The coalition is computed via an auctioning mechanism that takes place before the MRTA process. In this coalition, agents compute their participation bid as a function of their battery level, their distance to the NID event, and the redundancy of a node's coverage at its current location. These numeric features are fed into an FIS that computes the bid of the nodes. The First-Price Sealed Bid auction technique as described in Section 3.3 is used. This presents a suitable application to use and compare the two novel auction protocols described in the same section.

Some changes are needed in order for the nodes to compute their bids in one of the two other auction protocols. First, a *task valuation system* has to be defined. The task valuation system used in this context is presented in Table 5.1. All functions and values in that table were determined experimentally. This valuation places higher value on perimeter points that have lower redundancy, on nodes that have fewer neighbours, and on nodes that have neighbours that are in distress. This means that nodes that cover portions of the perimeter with low coverage, have fewer neighbours, and have distressed nodes as their neighbours will ascribe higher value to their current position.

Table 5.1: Risk Mitigation Value Mapping

Item	RMU per item
Perimeter Point (P)	$\max(\text{TIF} - 0.25 \times (\text{RF}-1), 0)$
Neighbor (N)	$\max(10 - 2 \text{ per previous N}, 0)$
Distressed Neighbor (DN)	50

Agents compute their valuation on each round of the auction protocol, weight it with their measured ability computed as previously, and use this to compute their bid value. Since the auction protocol messages in each round contain the winners, or the members of the possible coalition, agents can recompute their value based on the knowledge that the agent might move. In other words, they can assume that the winning node will no longer be at its current location, thus influencing the value of the node at its current position.

The value of joining the coalition, the auction item value, fluctuates up and down when using the Dutch-Japanese protocol. As agents receive these messages from the auctioneer, they indicate if they are willing to participate based on the advertised value being higher or lower than their own computed worth in their current position per the valuation system values. If the value of the auction item is not high enough, then the agents send their ability to complete the task, as previously defined in [8]. In the case of the English auction protocol, agents start at a 10% of the value at their current position, then bid in increments of 10% of their value until their maximum value is attained. This allows for nodes to recompute their value, yielding coalitions with nodes that are from different parts of the network.

The worst-case complexity of each auction protocol for this application is presented in Table 5.2, where M denotes all possible combinations of agents in the network that can respond to the event, and N is the coalition size. M is an upper bound of the actual total number of bidding rounds required since the protocols need only find one valid combination of agents that can serve as a potential coalition. For each round, the bidders will need to iterate through the list of current winners and assess their possible gain. This is the same for the English auction, except with more bidding rounds. The DJ algorithm has such a complexity since it mimics a binary search

Table 5.2: Worst-Case Auction Protocol Complexity

Algorithm	Auctioneer Complexity	Bidder Complexity
FPSB	$O(1)$	$O(1)$
Dutch-Japanese	$O(\log M)$	$O(N \log M)$
English	$O(M)$	$O(NM)$

algorithm.

In order to test and compare the application of the RMF with one of the auction protocols, an agent-based simulation is implemented where each of the nodes in the RSN represents an individual agent that is capable of communicating with other nodes, has simulated sensor inputs, has the ability to relocate itself, and is running a version of the RMF. The following experimental procedure is used: (1) Start the simulation; (2) Wait until the network reaches steady state; (3) Trigger a single NID event; (4) Allow the task allocation to complete; (5) Terminate the experiment.

The simulated nodes in the network require some time to auto-configure themselves for ad-hoc communication networks. Following this, the network enters normal operations where the distress risk of every node is verified periodically. We refer to “steady state” as when the setup is complete, and before nodes attain a high risk of distress due to low battery levels. The following performance metrics will be used to evaluate the proposed protocols.

Network Utilization: The total number of packets created from task announcement to auction end. A low network utilization is preferable since communication is critical in battery-operated WSNs.

Coalition Size: The size of the coalition determined by the auction protocol. An auction protocol should always return a coalition size between the preset minimum and maximum sizes;

Coalition Diversity: The diversity of the locations of the perimeter where the agents are picked from. Always selecting the nearest neighbours of the node in distress that triggered the MRTA process results in a local thinning of the perimeter, which

ultimately increases the risk of a compromised critical infrastructure; and

Time: The execution time from when the task is announced to auction end. Auctions should not be lengthy in order for the MRTA process to effectively respond to the time-sensitive event(s). All nodes are equal in computational power.

The experiment will be conducted for each of the scenarios in Table 5.3 with the specified parameters. For the probabilistic scenarios, the same set of generated positions and sensor ranges will be used to ensure equal contexts. The following criteria are proposed when creating scenarios:

Equal distance (U) vs Exponential distance (E) The distance between any two robots at network start is either equal, or defined by an exponential distribution. An exponential distribution is used to simulate an experienced network, while equal distance corresponds to a newly deployed network;

Sparse (S) vs Dense (D): The density of robots per perimeter point, or the coverage redundancy of any given perimeter point. This can be changed by reducing the distance between robots. In a sparse scenario, there is often little that can be done to recover perimeter coverage and gives little to optimize in the MRTA process when compared to a dense scenario; and

Homogeneous (HO) vs Heterogeneous (HE) The sensor payload of the robots. In heterogeneous networks, some agents might be better suited for some tasks. In this case, the sensor range of the robot's sensors will vary per a normal distribution.

5.1.3 Result of Auction Protocols for RSNs in CIP

An auction timeout parameter was set to 180 seconds for all auction protocols in the experiments. The auction round timeout for the FPSB and the Dutch-Japanese protocol was set to 10 seconds. Finally, the minimum and maximum coalition sizes for all protocols were set to 5 and 15, respectively, proportional to the size of the network. These values were determined empirically. The coalition set size was bounded

Table 5.3: Scenarios

Scenario	Distance Between Robots (m)	Sensor Range (m)	Expected Robot Density (/m)
U,S,HO	40	35	0.025
E,S,HO	Exp(40)	35	0.025
U,S,HE	40	Norm(35,5)	0.025
E,S,HE	Exp(40)	Norm(35,5)	0.025
U,D,HO	20	35	0.05
E,D,HO	Exp(20)	35	0.05
U,D,HE	20	Norm(35,5)	0.05
E,D,HE	Exp(20)	Norm(35,5)	0.05

proportionally to the size of the network. The bounds were found experimentally. The agent with an identifier of 2 was selected as the event source in all scenarios.

The sparse network scenario results are presented in Table 5.4, with scenarios annotated as Distribution/Density/Agent Diversity, followed by the auction protocol. For example, *USHO* indicates a scenario where the network is uniformly (U) distributed over the perimeter, has a sparse (S) density, and has homogenous (HO) agents. The network has 20 agents for the equal distance scenarios and 19 in the exponentially distributed scenarios.

The FPSB protocol finds agents which are able to complete the task, while the two other protocols prompted other adjacent agents to participate in the coalition in subsequent rounds. Consequently, the proposed protocols perform better in terms of determining appropriate coalitions, but have a much higher network utilization. The Dutch-Japanese protocol takes more time to resolve than the two other protocols. The English protocol yields a better coalition than the FPSB protocol in as much time as the FPSB protocol, but at the cost of higher network utilization.

The dense network scenario results are presented in Table 5.5. The network has 40 agents for the equal distance scenarios and 38 in the exponentially distributed scenarios. The auction times are different, with the FPSB protocol ending in constant

Table 5.4: Sparse Scenario Metrics

Scenario	Time (s)	Coalition Size (Agents)	Network Utilization (Packets)
USHO-FPSB	10	6	39
USHO-DJ	20	9	78
USHO-EA	10	9	66
USHE-FPSB	10	6	39
USHE-DJ	20	7	78
USHE-EA	10	9	66
ESHO-FPSB	10	9	37
ESHO-DJ	20	10	74
ESHO-EA	10	10	56
ESHE-FPSB	10	8	37
ESHE-DJ	20	10	74
ESHE-EA	10	10	58

time, and the two proposed protocols often reaching timeout. Network utilization is much higher in both of the new protocols, due to the communication needed for each auction round

The red node (#2) is the distressed node that is at risk of failure in Figure 5.2. These are the same positions used by all E-D-HO scenarios. Note that there is a sizable gap in the lower portion of the network. Gaps are possible for any number of reasons, such as having a manned checkpoint in that area. RMF would not have permitted such a hole if a gap brings about a higher risk of a compromised CIP. It is also important to note that a failure in node 2 does not seem to bring about any important risk, since its sensor coverage is redundant with other sensor coverage. However, it is not the job of the auction protocol to determine this; the optimizer will decide the best ways to relocate the nodes, which includes the possibility of not moving them at all.

One of the problems with the FPSB protocol is its inability to select agents farther away from the event since agents only bid based on their ability, where distance to the event is an important factor. With the implementation of the EVS and the

Table 5.5: Dense Scenario Metrics

Scenario	Time (s)	Coalition Size (Agents)	Network Utilization (Packets)
UDHO-FPSB	10	13	79
UDHO-DJ	180	15	1422
UDHO-EA	180	15	1796
UDHE-FPSB	10	13	79
UDHE-DJ	180	15	1422
UDHE-EA	180	15	1798
EDHO-FPSB	10	13	75
EDHO-DJ	180	15	1350
EDHO-EA	180	15	1604
EDHE-FPSB	10	12	75
EDHE-DJ	20	15	150
EDHE-EA	140	15	1252

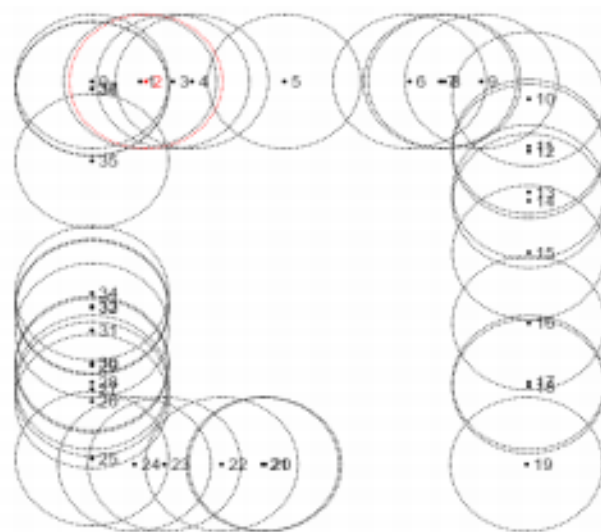


Figure 5.2: E-D-HO Configuration

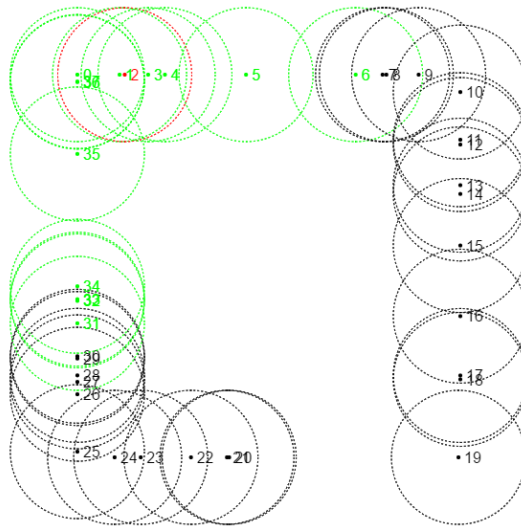


Figure 5.3: E-D-HO Results for First-Price Sealed Bid

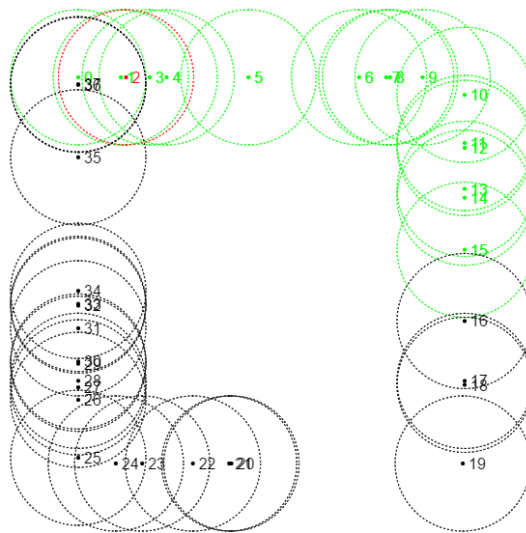


Figure 5.4: E-D-HO Results for Dutch-Japanese Protocol

multi-round protocols, it would be expected that smaller subgroups form from agents located in areas of higher density. Such subgroups are present and are illustrated in Figure 5.3 for the FPSB protocol, Figure 5.4 for the Dutch-Japanese protocol, and Figure 5.5 for the English protocol, where nodes in green form the coalition. This is a pattern that is repeated in other scenarios, and consequently it seems the proposed protocols are returning coalitions as predicted.

The Dutch-Japanese protocol returns a coalition which is usually the middle ground between the English and FPSB protocols. It favors sets of agents adjacent to the event, but will form subgroups and pick agents farther away from the event when

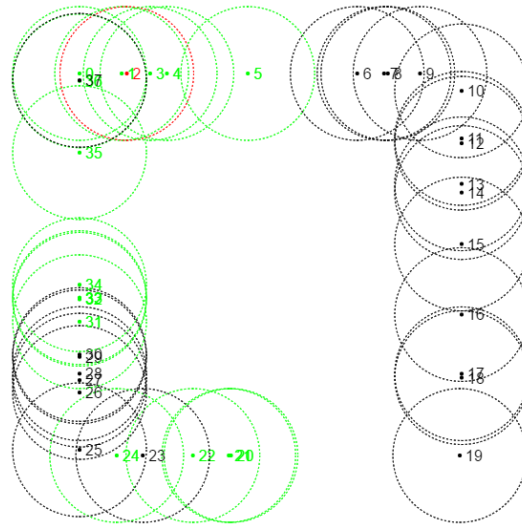


Figure 5.5: E-D-HO Results for English Protocol

warranted. In these dense scenario results, the two proposed protocols perform better in all scenarios when it came to coalition quality as they are able to select better nodes, and bigger coalitions.

However, the FPSB protocol performs better in terms of time and network utilization, since it guarantees constant time and only requires a fixed number of messages. The English protocol takes considerably more time in dense scenarios and heavily utilizes the network. Finally, the Dutch-Japanese protocol often lies in between these two protocols in all metrics, but can take more time in sparse scenarios. Since the protocols only need to identify one good combination of agents, they work better in highly diverse networks.

These protocols can be used in any MRTA situation by defining an appropriate valuation system. However, the end results are dependent on this definition. Hence, agents should be able to decide by themselves the risk mitigating value of accomplishing a task and learn the value of entities such as perimeter points or neighbors. Additionally, only the sensor range of the robots was modified in these scenarios. Scenarios involving more diverse and complex agents need to be evaluated. The proposed protocols often consumed their entire time budgets, while still yielding good coalitions. Thus, more work needs to be done to identify proper stopping criterion for the protocols other than time and coalition size.

This concludes the application of the methodology of the RMF, as all four steps have been completed in conjunction with the previous work done in [8]. The risk model was specified, followed by the required sensors and actuators. Appropriate risk mitigation actions were identified, as well as a method for an optimized MRTA. A pre-optimization step was added in which the task valuation system and the two novel auction protocols were applied. Then, a simulation was fleshed out in which this application can be tested to refine and validate the applied methodology. The results were presented, which showed the strengths and weakness of each protocol.

5.2 Maritime Smuggling Detection and Mitigation

This second section presents an application of the RMF for the case study in the detection and mitigation of maritime smuggling in ports as presented in Section 4.2. Each step of the methodology will be detailed. The AoI in this case study is rather large and acts of maritime smuggling must be detected and intercepted in real time, requiring adequate warning in order to compute the risk mitigation tasks. Consequently, the use of hybrid RSNs composed of air and ground assets is suitable, hence the contribution proposed in Section 3.4 should be used. An overview is given in Figure 5.6.

5.2.1 Risk Model for Suspicious Activity

This section will first present the risk model that is used to trigger monitoring requests for the ASN. First, risk features that can be used to direct the ASN towards possibly

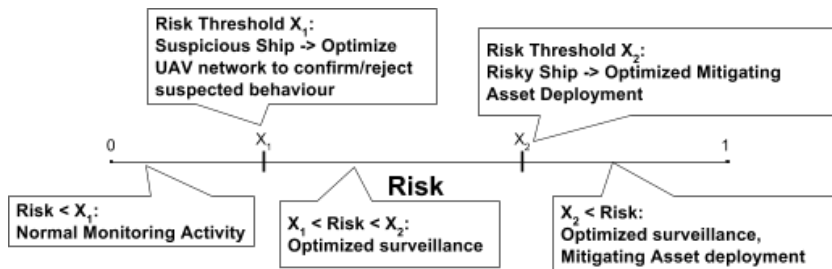


Figure 5.6: Maritime Smuggling Detection and Mitigation Overview

risky situations will be defined. The goal is to catch maritime smuggling events involving a mother ship that is AIS-enabled through cooperating detection. This rendez-vous would warrant additional monitoring but does not necessarily indicate smuggling activity. It could simply be a chance crossing, bunkering, or a tug-based operation, to name a few legitimate rendez-vous-based activities. The risk features to detect rendez-vous are defined as follows.

AIS Off Time (A): This risk feature is defined as $1 - e^{-10*t}$, where t is the percentage of time the AIS transceiver is perceived to be offline. This feature has the following linguistic terms and membership functions: Small (Trapezoidal: 0, 0, 0.05, 0.15), Moderate (Trapezoidal: 0.1, 0.15, 0.45, 0.6), and Large (Trapezoidal: 0.45, 0.6, 1, 1);

Risk of Departing Port (P): An indicator of the risk of the departing port between 0 and 1 that relies on expert knowledge. Some source ports are known to be less monitored than others [104]. This feature has the following linguistic definitions: Low (Trapezoidal: 0, 0, 0.25, 0.5), Medium (Triangle: 0.25, 0.5, 0.75), High (Trapezoidal: 0.5, 0.75, 1, 1);

Distance to Nearest Vessel (D): Normalized distance to the closest vessel measured in the vessel's width, with a maximum of 20 widths. Not all vessels transpond AIS, so the exact position of each vessel is not always known but detecting and tracking targets through data fusion [104] is possible. This feature has the following linguistic definitions: Close (Triangle: 0, 0, 0.5), Medium (Triangle: 0.25, 0.5, 0.75), Far (Triangle: 0.5, 1, 1); and

Time of Day (T): The current hour and minutes, normalized between 0 and 1, and deducted from 1. This feature has the following linguistic definitions: Pre-Dawn (Trapezoidal: 1, 1, 0.8, 0.7), Morning (Trapezoidal: 0.0, 0.0, 0.2, 0.3), Afternoon (Trapezoidal: 0.6, 0.5, 0.3, 0.2), Evening (Trapezoidal: 0.8, 0.7, 0.6, 0.5).

The output risk is defined as the following fuzzy variable:

Risk (R): The inferred suspicion of maritime smuggling risk. This feature has the following linguistic definitions: Low (Trapezoidal: 0, 0, 0.25, 0.5), Medium (Triangle:

0.25, 0.5, 0.75) High (Trapezoidal: 0.5, 0.75, 1, 1).

The inference rules are given below for a Mamdani-type FIS [115].

- If D is Far then Risk Low.
- If T is Morning or Afternoon and D is Close and P is Low and A is Small then R is Low.
- If D is Medium and P is not Low and A is not Small then R is Medium.
- If T is Pre-Dawn or Evening and D is Close and P is Low and A is Small then R is Medium.
- If T is Morning or Afternoon and D is Close and P is not Low and A is not Small then R is Medium.
- If Time of Day is Pre-Dawn or Evening and Distance to Nearest Vessel is Close and Risk of Departing Port is not Low and AIS Off Time is not Small then Risk is High.
- If D is Close and P is High and A is Large then R is High.

This concludes the extra step needed to include aerial nodes into the RSN for monitoring missions. When risk events are detected, the MRTA process defined in Section 3.4 will be executed, and the aerial nodes will provide additional information about the situation; this information can be used in order to detect maritime smuggling behaviour. The methodology of the RMF can now be applied.

5.2.2 Maritime Smuggling Behaviour Detection

The first step of the RMF methodology requires the definition of a risk model. In this context, the ASN is tracking the small recreational vehicles (SRVs) that were in rendez-vous with the merchant vessel (MV). Consequently, additional information is available. The following are the risk features of the model that will be used, as well as how they are computed.

Illumination (I): Normalized estimation of the extracted illumination emanating from a ship, gathered through light source identification analysis from the images of the tracked vessel gathered by the UAVs. This feature has the following linguistic definitions: Poor (Trapezoidal: 0, 0, 0.25, 0.75), Good (Trapezoidal: 0.25, 0.75, 1, 1);

Evasiveness (E): The absolute difference between two subsequent values defined as the average distance of the vessel to other vessels versus the average distance of ships to other vessels. This is used to capture the behaviour of vessels that are constantly attempting to keep a distance from other vessels. This feature has the following linguistic definitions: Social (Trapezoid: 0, 0, 0.25, 0.75), Evasive (Trapezoid: 0.25, 0.75, 1, 1); and

Time of Day (T): The current hour and minutes, normalized between 0 and 1, and deducted from 1. This feature has the following linguistic definitions: Pre-Dawn (Trapezoidal: 1, 1, 0.8, 0.7), Morning (Trapezoidal: 0.0, 0.0, 0.2, 0.3), Afternoon (Trapezoidal: 0.2, 0.3, 0.5, 0.6), Evening (Trapezoidal: 0.5, 0.6, 0.7, 0.8).

The output smuggling risk is defined as the following fuzzy variable:

Risk (R): The inferred smuggling risk. This feature has the following linguistic definitions: Low (Triangle: 0, 0.25, 0.5), Medium (Triangle: 0.1, 0.5, 0.9), High (Triangle: 0.75, 1, 1).

A Mamdani-type FIS [115] is used again. The inference rules are given below.

- E is Social and I is Good then R is Low
- E is Social and I is Poor and T is Morning then R is Low
- E is Social and I is Poor and T is Afternoon then R is Low
- E is Social and I is Poor and T is Pre-Dawn then R is Medium
- E is Social and I is Poor and T is Evening then R is Medium
- E is Evasive and I is Poor and T is Morning then R is Medium
- E is Evasive and I is Poor and T is Afternoon then R is Medium
- E is Evasive and I is Poor and T is Evening then R is High

- E is Evasive and I is Poor and T is Pre-Dawn then R is High
- E is Evasive and I is Good and T is Pre-Dawn then R is Medium
- E is Evasive and I is Good and T is Evening then R is Medium
- E is Evasive and I is Good and T is Morning then R is Low
- E is Evasive and I is Good and T is Afternoon then R is Low

5.2.3 Risk Mitigation

This step requires the definition of appropriate data structures, operators, and fitness functions for the risk-mitigating MRTA process. An MOEA is used for this MRTA so the data structures are genes and chromosomes, while the operators are mutation and crossover operators. Police vehicles and coast guard vessels will serve as mitigating assets. This step attempts to give the best combination of assets for a proper mitigation task allocation, so while these assets are not technically part of the RSNs, they could easily be exchanged in other scenarios. Tasks will first be defined, then genes and chromosomes will be designed, followed by mutation and crossover operators. Finally, the fitness functions will be presented.

A path can be predicted for the smuggling vessel with the data gathered by the ASN, and certain ports can thus be predicted as possible berthing points. A risk mitigation task is the relocation of a mitigation asset to one of the possible ports. Table 5.6 presents the assets considered and their parameters, where cost is the cost of the asset per meter travelled. A chromosome is encoded as shown in Figure 5.7. Each gene corresponds to one asset and has 2 layers; an activation layer and a target port. It can be described as follows, where L is the set of predicted ports as given by the function f_L .

$$Gene = \begin{bmatrix} a \in A \\ p \in L \end{bmatrix} \quad (5.1)$$

Table 5.6: Mitigating Assets

Type	Cost per meter	Risk Mitigation	Speed (m/s)
Police cruiser	1	0.1	20
Police helicopter	10	0.5	70
Coast guard vessel	5	0.25	15

	Gene 1	Gene 2	Gene 3	...
Layer 1	Enabled?	Enabled?	Enabled?	...
Layer 2	Target Port	Target Port	Target Port	...

Figure 5.7: Mitigation Task Chromosome

$$L = f_L(C) \quad (5.2)$$

The mutation operator can be described as follows, where P_A is as previously defined.

$$f_M = \begin{bmatrix} P_A \\ P_L \end{bmatrix} \quad (5.3)$$

$$P_L = \begin{cases} 1/|L| & l \mid l \in L \\ 0 & else \end{cases} \quad (5.4)$$

As solutions must aim to use as little resources as possible, a fitness function will thus evaluate the cost of a solution. The cost metric of a mitigating asset is simply an fiscal estimate of moving the asset by one meter multiplied by the distance (in meters) it is required to move. The fitness function is presented in Algorithm 5.1.

Algorithm 5.1 Cost Fitness Function

```

Cost ← 0
for all Gene | Gene.a = 1, Gene ∈ Chromosome do
    Distance ← Distance(Asseti, Gene.Port)
    Asset Cost ← Asseti.cost * Distance
    Cost ← Cost + Asset Cost
end for
Return Cost
    
```

The second fitness function evaluates the latency of a solution, with faster solutions preferable. The speed parameter of mitigating assets is measured in meters per second. The fitness function is defined in Algorithm 5.2.

Algorithm 5.2 Latency Fitness Function

```

Latency  $\leftarrow$  0
for all  $Gene \mid Gene.a = 1, Gene \in Chromosome$  do
    Distance  $\leftarrow$  Distance(Asseti, Gene.Port)
    Latency  $\leftarrow$  Distance / Asseti.speed
    Total Latency  $\leftarrow$  Total Latency + Latency
end for
Return Latency

```

The final fitness function measures the actual mitigating power of the solution encoded in the chromosome, with those that can adequately stop the smuggling behaviour preferred over ones that cannot. V refers to the set of tracked vessels. The fitness function is defined in Algorithm 5.3.

Algorithm 5.3 Risk Mitigation Fitness Function

```

Ports  $\leftarrow$  { $Gene.p \mid Gene.a = 1, Gene \in Chromosome$ }
P.risk  $\leftarrow$   $\sum_{i=0}^{|V|} Prob(V_i, P) \forall P \mid P \in Ports$ 
for all  $Gene \mid Gene.a = 1, Gene \in Chromosome$  do
    Gene.p.risk  $\leftarrow$  Gene.p.risk * Asseti.mitigation
end for
Return  $\sum_{i=0}^{|Ports|} Ports_i.risk$ 

```

5.2.4 Refinement and Validation

This section will present a simulation designed for the case study of Section 4.2 in which this application of the RMF can be used. The smuggling scenario will first be simulated then repeated with three non-smuggling, yet suspicious, SRVs. Figure 5.8 presents an aerial view of the port of Barcelona and surrounding area. The dotted lines represent real ferry routes. The AoI of the hybrid RSN is in red.

In this area, we find many types of ships, though only the following types of ships will be described: Merchant vessels, ferries, small recreational vessels (e.g., pleasure

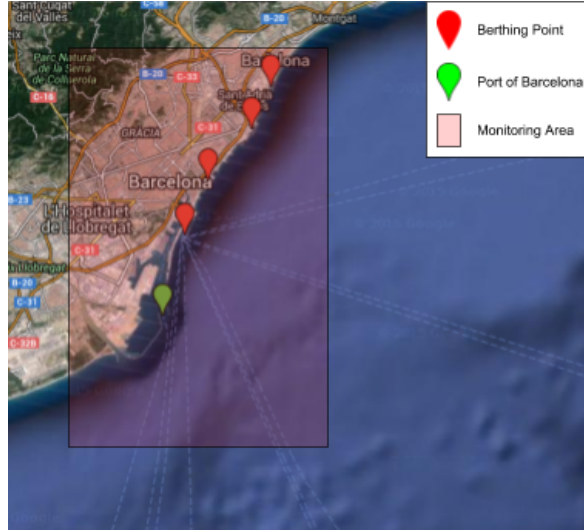


Figure 5.8: Geographic area for maritime smuggling case study

crafts), large private vessels (e.g., yachts). State machines with parameters will be used to define the behaviours of these ships, as was done in [116].

A merchant vessel (MV) is defined as a ship which carries cargo intended for the port of Barcelona, or any cargo ship leaving the port of Barcelona. A merchant ship first arrives in the port area, and is in the *arriving* state. In this state, the ship is heading towards the port. From this state, it may transition to the *loitering* state or the *In Port* state. In the *loitering* state, a merchant vessel might stop, move extremely slowly, or circle an area. This loitering state might be triggered by a myriad of reasons: the ship might have broken down, it might be waiting for traffic reasons, waiting for a pilot, etc. The vessel will exit the *loitering* state after a certain time period, then revert back to its previous state. Once the ship reaches the port, it goes into the *In Port* state for some time. Finally, it exits the port, with another optional *loitering* state. Once the ship exits the monitoring area, it is no longer relevant to record its state.

This is illustrated by the state machine in Figure 5.9. The cargo ship can travel at a speed of 0 to 15 m/s, has a docking time of 0 to 3 days, is 30 to 250 meters in length, has a source port, and its AIS can be off from 0 seconds or during the entire trip.

A ferry will behave in a similar way to a merchant vessel, but does not have the option to loiter. Figure 5.10 details the ferry behaviour. Ferries can travel from 0 to

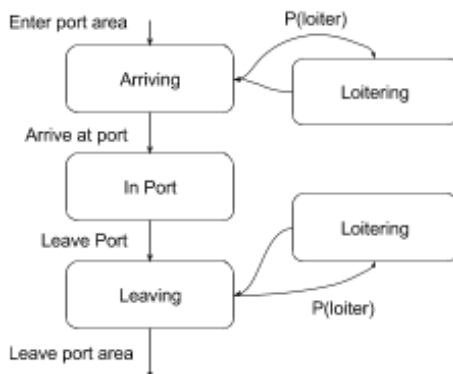


Figure 5.9: Merchant Ship Behaviour

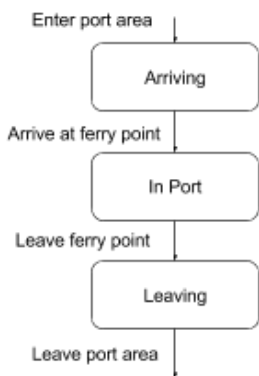


Figure 5.10: Ferry Behaviour

20 m/s, have a docking time of 0 to 1 hour, and have a length of 30 to 250 meters.

A small recreational vessel (SRV) is any vessel used for recreation, such as sailboats and small fishing vessels. These ships cannot operate in open sea, and are usually used during the day. Figure 5.11 details the behavioural model for these ships. SRVs have a speed of up to 50 m/s, are in operation from 5 to 20 hours, are 1 to 50 meters in length and are lit up to some degree defined between 0 and 1.

Large private vessels (LPVs), such as private yachts, will be modelled with the same behaviour as MVs. The difference is that large private vessels may head towards any



Figure 5.11: Small Recreational Ship Behaviour

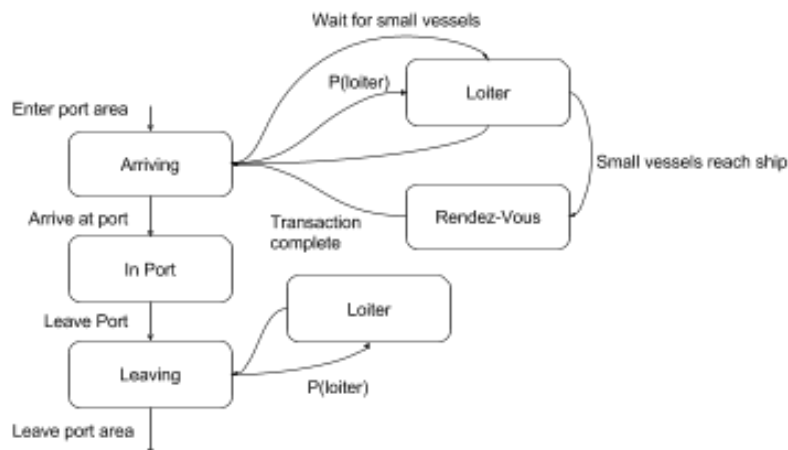


Figure 5.12: Smuggling Merchant Vessel Behaviour

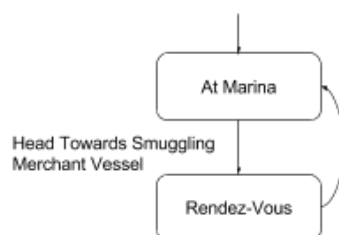


Figure 5.13: Small Smuggling Vessel Behaviour

of the marinas. LPVs can travel at a speed of up to 40 m/s, are 50 to 150 meters in length, and head towards one of the marinas.

A *smuggler* is a vessel which is conducting smuggling operations. In this document, a specific smuggling activity which includes a merchant vessel and smaller recreational vehicles will be described per the knowledge presented in [104]. A smuggling merchant vessel's behaviour is modelled as shown in Figure 5.12. This is essentially the same model followed by a merchant vessel, but with the added state of a rendez-vous, attainable through the *loitering* state. A smuggling merchant vessel has the same parameters as a merchant vessel.

A small smuggling vessel behaves similarly to its non-smuggling counterpart, except that it heads directly to the rendez-vous location, then heads back to the marina. This is shown in Figure 5.13. This ship has the same parameters as those of the small recreational vehicle.



Figure 5.14: Initial Environment

5.2.5 Result of Hybrid RSNs for Maritime Smuggling Mitigation

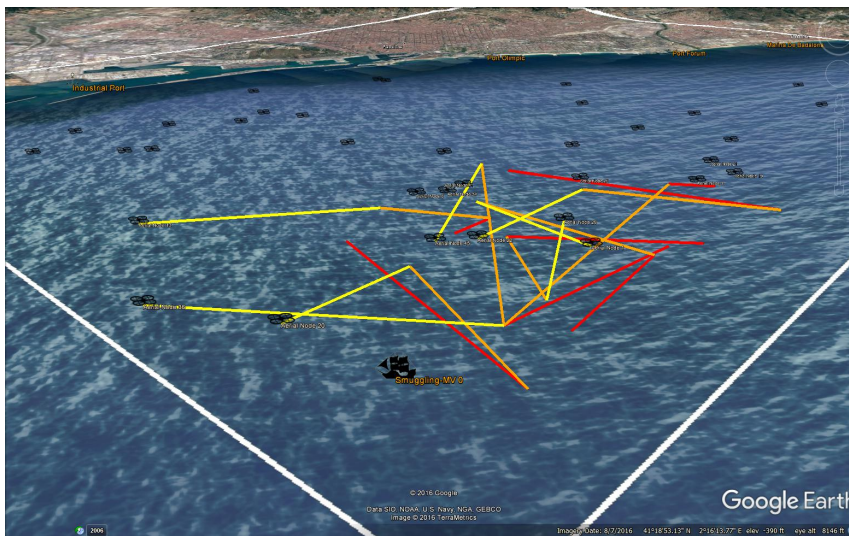
The initial environment is illustrated in Figure 5.14. The coast of Barcelona can be seen approximately 11 Km away. The environment's grid cells have a dimension of 250m x 250m for the purposes of ASN optimization. The risk threshold to trigger aerial monitoring was set at 0.5, corresponding to a medium risk, while the risk threshold for smuggling mitigation was set at 0.75 corresponding to a high risk. The coalition size was limited to 7 and the genetic algorithm had a stopping criterion of 100 generations, a population size of 100, a mutation rate of 0.2 and a crossover probability of 0.85.

The MV departed from a medium risk port (risk of 0.5). Its AIS transceiver was off for a period of 2 hours during a week-long trip yielding a value of 0.1122 for the "AIS off" risk feature. It is engaging in a rendez-vous with three SRVs, a value of 0.1 for the distance to nearest ship feature, and the time is about 00:30 giving the "Time of Day" feature a value of 0.979. These features evaluate to a suspicious risk of 0.63219, well above the preset suspicious risk threshold of 0.5.

An ASN optimization generates non-dominated solutions. Table 5.7 presents 3 of the solutions and their fitness values. A solution is shown in Figure 5.15, where yellow,

Table 5.7: Monitoring Solution Fitness

Solution ID	Resources	Connectivity	Relevancy
1	0.0	0.0	0.0
2	0.526	0	21
3	0.250	21	15
4	0.216	14	12
5	0.349	0	18

**Figure 5.15:** Monitoring Task Allocation

orange, and red correspond to the first, second, and third segments of the path, respectively. It has a relevancy of 21 cells, a fitness value of 0.5224 for resources, and 7 for connectivity. This solution is an expensive one, resource-wise, but offers a good connectivity for each segment and a high relevancy.

Additional data that was not previously available is collected with the ASN monitoring service. Images of the suspected vessels can be taken, as well as their positions that were previously sporadically reported from ship-based cameras, radar, and other third-party sources. Tracking the first vessel yields a value of 0.63219 for prior smuggling risk, 0.72173 for illumination, 0.62495 for evasiveness, and 0.958 for time of day evaluating to a smuggling risk of 0.5, well below the preset smuggling risk threshold of 0.75. However, this risk is raised on subsequent evaluations as a consequence of the prolonged monitoring, and breaches the 0.75 threshold about 10 minutes later, resulting in a risk mitigating MRTA process. The mean smuggling risk of one hundred

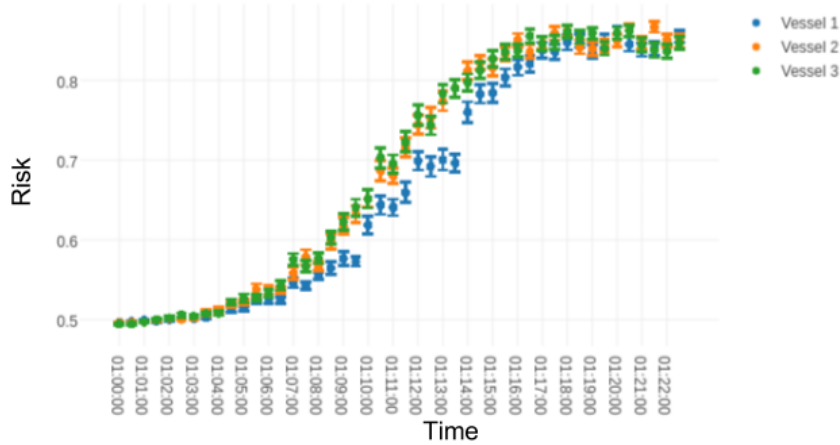


Figure 5.16: Smuggling Ship Risk Over Time

Table 5.8: Mitigation Solution Fitness

Solution ID	Cost	Latency (s)	Mitigation (%)
1	64,976.21	1,439.9	0.33215
2	1,301,129.33	13,085.4	1.3539E-4
3	68,104.74	1596.3	0.29893

repetitions of the experiment over time is shown for the three SRVs in Figure 5.16.

This risk mitigating MRTA process is started after the risk crosses the threshold of 0.75 and yields a set of nondominated solutions. The fitness values of three of these solutions are presented in Table 5.8. The end of the scenario is shown in Figure 5.17, where mitigating assets are present at the correctly predicted destinations of Port Olympic, Port Forum, and Marina de Badalona. Some assets are sent to Port Vell due to the chosen mitigation solution that valued mitigation over cost. This successfully concludes the scenario.

The second scenario starts with the detected rendez-vous triggering the aerial optimization. The second risk feature assessment for one of the vessels gives a value of 0.1935 for illumination, 0.11 for evasiveness and 0.958 for time of day, which evaluates to a risk of 0.25. The system tracks the ships until they reach their destination marinas, never having taken actions against the vessels. The experiments were repeated 100 times. The scenario successfully ends with no mitigating actions against the SRVs.

Briefly, the designed system was able to successfully detect and mitigate maritime



Figure 5.17: Successful Risk Mitigation

smuggling with the use of UAVs. Furthermore, it was able to differentiate between smuggling behaviour and normal civilian behaviour. This concludes the application and simulation of the RMF for maritime smuggling detection and mitigation. This completes the methodology for applying the RMF to maritime smuggling detection and mitigation for hybrid RSNs that have both UAVs and ground nodes.

5.3 Supply Chain Disruption Mitigation

The following section will present an application of the RMF methodology in order to predict and mitigate disruption in supply chains, like in the case study detailed in Section 4.3. This section also presents the first application of the RMF for the IoT instead of a RSN. Consequently, two of the contributions proposed in Chapter 3 are applicable. The first is the use of the rRMF presented in Section 3.2 and the second is the use of the Continuous Risk-Aware Response Generation method detailed in Section 3.5. The following subsections of this chapter will follow the methodology to apply the RSN for supply chain disruption mitigation. The application in this section is of a more proactive nature than the usual reactive behaviour of the RMF by predicting possible risk and attempting to resolve the risky situation before it happens.

5.3.1 Disruption Risk Model and Features

An application of the RMF requires appropriate vessel risk features from incoming data and the specification of a fuzzy risk model. The risk features are presented below. All risk features are normalized between 0 and 1, with 1 indicating a higher risk.

Storm Delay (W): This is the first risk feature. A three-dimensional linear regression as determined by the least-squares method to the ship's track, where a point corresponds to a timestamped two-dimensional position. The same is replicated for known storms based on meteorological reports from appropriate sources such as the United States National Oceanic and Atmospheric Administration (NOAA) ¹. If there is an intersection in the time range from the last positional update from the tracked entity to a set time in the future, the risk of delay is returned as 1. In the other case, a Gaussian function is used with mean on the proportion of the time dimension of the intersection point over the time range for the vessel and a parameterizable standard deviation. The risk becomes the integrated distance of the storm's time proportion in this function to the mean, normalized to 1. The closest points are used instead if there is no intersection, and a threshold is used as an intersection tolerance;

Importance to Contract (I): This risk feature takes into consideration the importance of a ship's cargo to its contract fulfillment. The proportion of the amount of a necessary item over the quantity demanded in the contract is computed. The maximum of these proportions becomes the risk feature value. A risk of 1 is yielded for any ship transporting a quantity of an item that is absolutely critical to fulfill the supply contract;

Deadline Proximity (D): This third risk feature analyses the risk of the proximity of the expected arrival time and date of a ship to the deadline. This feature is normalized for up to a configurable maximum of days. A ship that is closer to the deadline can not be delayed in any way or it might miss it; and

¹National Buoy Data Center. (n.d.). Retrieved July 19, 2017, from <http://www.ndbc.noaa.gov/>

Port Status at Arrival (P): This risk feature is the port's docking busyness at the expected time of arrival (ETA) of the ship. A port that is busier is less likely to be able to accommodate changes in a ship's ETA that may result in further delays. The feature is computed by taking a weighted average of a numerical indicator which can be data mined from AIS messages, sensors near the port, or through soft-data. This indicator represents the business of a port for a particular day. The weights follow a normal distribution.

After these features are specified, a risk model is used to incorporate them and provides a total risk value of a certain event. In this case, the event will be an unfulfilled contract. Consequently, the model is trying to evaluate the risk of an unfulfilled contract. The model will be based on a Takagi-Sugeno FIS [117]. All risk features are fuzzified into these linguistic terms and their corresponding membership functions.

L: The Low linguistic term has a triangle membership function with parameters 0, 0.25, and 0.5; and

M: The Medium linguistic term has a triangle membership function with parameters 0.25, 0.5, and 0.75; and

H: The High linguistic term has a triangle membership function with parameters 0.5, 0.75, and 1.0.

The output variable, the risk of a disruption (R), has the following linguistic terms and membership functions.

S: Ship-centric risk that evaluates to $S/2 + I/2$.

P: Port-centric risk that evaluates to $P/4 + D/4 + I/2$

A: Default risk evaluation that is $S/4 + I/4 + D/4 + P/4$

L: Low risk constant of 0.1.

M: Medium risk constant of 0.6.

H: High risk constant of 1.0.

The fuzzy rule base is composed of the following rules:

1. if W is H and I is H then R is A
2. if P is not H and D is H then R is A
3. if P is H and D is not H then R is A
4. if W is H and I is not H then R is A
5. if W is not H and I is H then R is A
6. if W is not H and I is not H then R is P
7. if P is H and D is H then R is S
8. if P is not H and D is not H then R is S
9. if W is L and I is not L and D is not L and P is not L then R is S
10. if W is M and I is M and D is M and P is M then R is M
11. if W is L and I is L and D is L and P is L then R is L
12. if W is H and I is H and D is H and P is H then R is H

Rules 1 to 5 are general rules which use all risk features. Rule 6 puts emphasis on risk features concerning the port, since risk features concerning the ships are fairly low. Rules 8 to 10 put more emphasis on the ship's risk features. Finally, the last three rules are used to bias the risk. The Apache Kafka ² framework was used for the underlying communication system and the jFuzzyLite library ³ was used for the FIS implementation.

5.3.2 Disruption Mitigation Optimization

The application of a MOEA to derive mitigation actions requires an appropriate solution encoding, mutation and crossover operators, as well as fitness functions to be defined. Solutions are composed of actions that yield one or more maritime supply chain (MSC) segments which are then evaluated by the fitness functions. A segment

²Apache Kafka. Computer software. Apache Kafka. Vers. 0.11.0.0. Apache, n.d. Web. 11 July 2017. <<https://kafka.apache.org/>>.

³jFuzzyLite. Computer software. FuzzyLite. Vers. 6.0. FuzzyLite, n.d. Web. 11 July 2017. <<https://www.fuzzylite.com/>>.

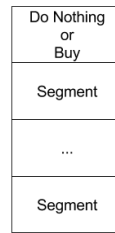


Figure 5.18: Example Gene

has the following properties: a failure risk associated with using the segment, items and their quantity that are transported on the segment, and the cost of transporting the quantity of item over the segment.

Three actions will be considered: (1) **Do Nothing** action, a special action which does not generate a segment but instead returns the transportation segments fulfilled by the ships already under way, (2) **Buy** action, which enables buying quantities of items from a particular supplier, and (3) **Segment** action, which splits existing segments into multiple components based on risk-cost tradeoffs. Each solution must have a minimum number of Do Nothing actions, one for each pre-existing segments. Some of these actions may be chained. The Do Nothing action and the Buy action are basic actions and form the first link of an action chain while Segment actions may extend the chain, segmenting the results of the previous action, including other Segment actions

A gene in the chromosome is an action chain. Genes are of varying length, meaning that some action chains are longer than others. A chromosome will be formed by multiple genes, and may be extended or reduced if possible through the genetic operators. Chromosomes ultimately represent combinations of action chains that can be enacted. Figure 5.18 presents an example of a gene while Figure 5.19 depicts the chromosome design.

A crossover operator and a mutation operator will be used. Since the length of the chromosomes differs, this operator first determines the length of the two offspring chromosomes, which is in the inclusive range of both of the parents' lengths. The genes of the parents corresponding to their Do Nothing action chain are divided between

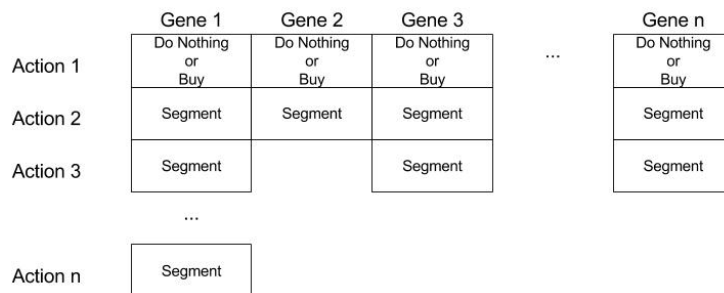


Figure 5.19: Example Chromosome

the two offspring. Subsequently, the offspring's genes are filled with duplicates from their parents by picking them randomly without replacement until they reach their required length.

The mutation operator does one of five randomly selected functions. The first function adds a Buy action that then forms the basis of a new action chain, i.e. a new gene. The second function is the opposite and removes a gene that has a Buy action at its base. The third function adds a Segment action to a random gene, if possible. The fourth function removes a Segment action from an applicable gene. The final function selects a gene and modifies its base Buy action. If there are no genes that can be used for the chosen function, then the mutation does the opposite effect. For example, if the mutation is to remove a Buy action and none are available, a Buy action will be added instead.

Two fitness functions that must be minimized will be used, along with the previously presented Resiliency Fitness Function. The first is the Cost Fitness Function which evaluates the average cost of segments resulting from a solution. Algorithm 5.4 presents this function. The f_s function returns the segments generated by an action chain, while f_c computes the total cost of a segment. The contract value is the value of the contract that is at risk of disruption.

The second fitness function computes the risk mitigation offered by a solution. The algorithm determines the risk of a trade item by sorting the segments which may have some of the trade items in order of increasing risk. The contract order for the trade item is then filled with lower risk items first. The risk for that trade item is then

Algorithm 5.4 Cost Fitness Function

```
count ← 0
cost ← 0
for Gene in Chromosome do
  for Segment in  $f_s(Gene)$  do
    cost ← cost +  $f_c(Segment) / Contract.Value$ 
    count ← count + 1
  end for
end for
Return  $cost/count$ 
```

the average of the risk of the components which were used to fill the order. This is repeated for each trade item. The average risk of all trade items is then computed and returned. The NSGA-II [29] implementation of the MOEA Framework ⁴ was used in this study.

5.3.3 Refinement and Validation

The simulation design here will be taken from the scenario presented in Section 4.3. Each vessel is designed as an agent that travels towards a destination location. A storm is set to intersect the path of certain ships, thus driving up the "Storm Delay" risk feature. A simulated stream for the "Storm Delay" (W) risk feature will be used due to the difficulty in generating quality synthetic data. Synthetic positions for ships are used.

The simulation has 100 suppliers that have a tradeoff between risk and cost so that no supplier is absolutely better than another, and the goods available at a supplier will be randomly selected. The same risk/cost tradeoff is applied to the 100 segmentation options. The risks of suppliers and segmentation options will be sampled from a Gaussian function with a mean of 0.6 and a standard deviation picked from a similar function of mean 0.1 and standard deviation 0.033. This is to simulate dynamic risk calculations for these suppliers as well. All data generated will be fed asynchronously to the rRMF, thus simulating data generated from the IoT.

⁴Hadka, David. MOEA Framework. Computer software. MOEA Framework. Vers. 2.1. N.p., n.d. Web. 11 July 2017. <<http://moeaframework.org/>>.

The first two generated trajectories are carrying half of the needed quantities of five commodities of varying costs for a contract each, while the last one is transporting two thirds of the needed quantity for the contract. A fairly busy port docking schedule will be used where the port is busy approximately 75% of the time. The ships are expected to dock two, three, and four days before the deadline, normalized over 10 days.

The "Storm Delay" risk value will be simulated for all three ships to stay at 0 for 3 minutes, increase to 1 over 2 minutes, then stabilize at 1 for another 3 minutes. This risk would realistically change in a timespan closer to many hours. The prediction length can be adjusted to have more accurate predictions or for extrapolating farther in the future. Finally, the intersection tolerance between two closest points is proportional to the size of the storm.

This concludes the application of the RMF methodology for maritime supply chain disruption mitigation with the rRMF for the maritime IoT and with continuous risk mitigation optimization.

5.3.4 Result of rRMF for Supply Chain Disruption Mitigation

A population size of 25 was used for the MOEA and a decreasing mutation probability that starts at 1 and ends at 0.5 after a minute. Since the algorithm runs continuously and starts before there is a need for mitigation, using a time-based decay is more suitable. A minute was experimentally found to be suitable in creating early diversity. A high mutation rate is needed to quickly explore and diverge from the original solution created from the three Do Nothing actions.

The overall disruption risk for the three vessels is reported in Figure 5.20, Figure 5.21, and Figure 5.22. The risk features for the first vessel had fairly constant values of 0.7 for feature D, 0.592 for feature P, and 0.758 for I. The second vessel had similar values, while the third had a risk value of 1 for feature I. The main driver behind the increase in Disruption risk is the feature D. In Figure 5.21, recurrent spikes can

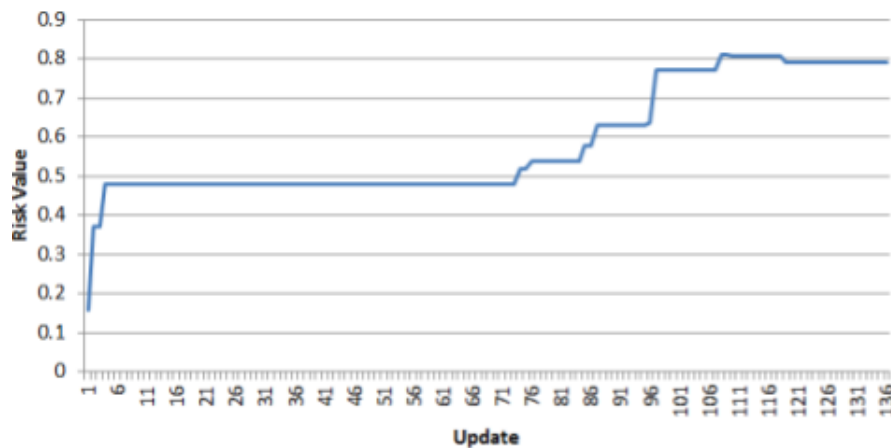


Figure 5.20: Disruption Risk for Vessel 1 over Time

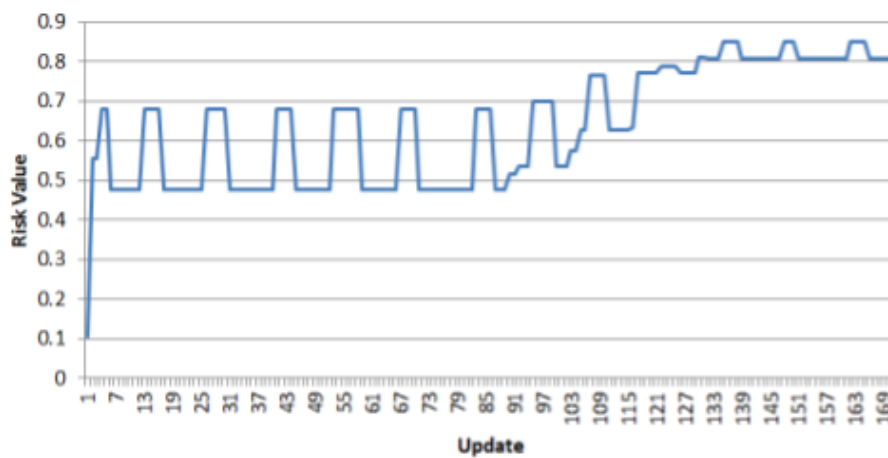


Figure 5.21: Disruption Risk for Vessel 2 over Time

be noticed in the disruption risk resulting from small oscillations in the risk features which result in different rules triggering. This can be alleviated with better disruption risk models.

The solutions generated by the MOEA are interesting and correspond to what was expected. The average, maximum, and minimum risk fitness value of the solutions over time is presented in Figure 5.23, while the same metrics for the cost fitness value over time is given in Figure 5.24. Finally, Figure 5.25 depicts these metrics for the resiliency fitness value over time. All three of these figures reveal important aspects of the dynamic optimization, and the increasing disruption risk of the ships is reflected in there.

The fitness values are the same in the beginning since the MOEA starts with a popula-

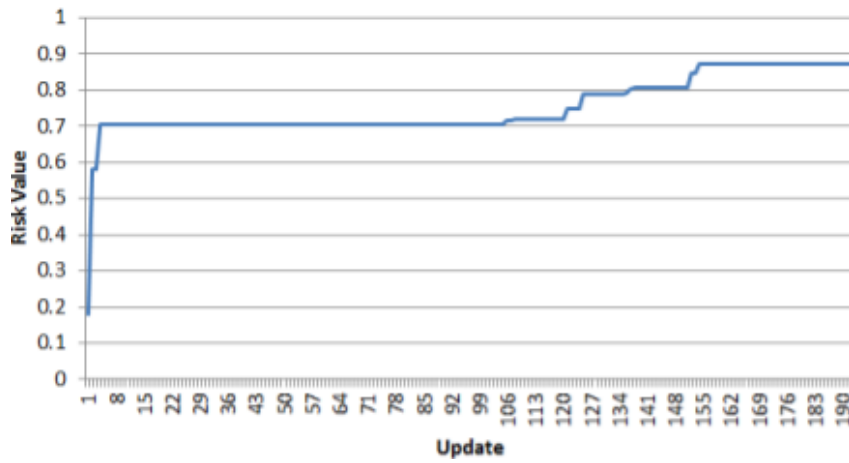


Figure 5.22: Disruption Risk for Vessel 3 over Time

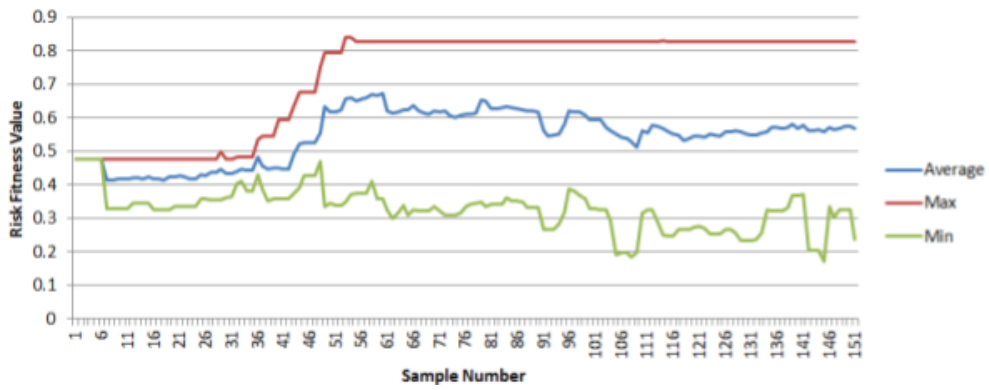


Figure 5.23: MOEA Optimal Solutions Sample Risk over Time

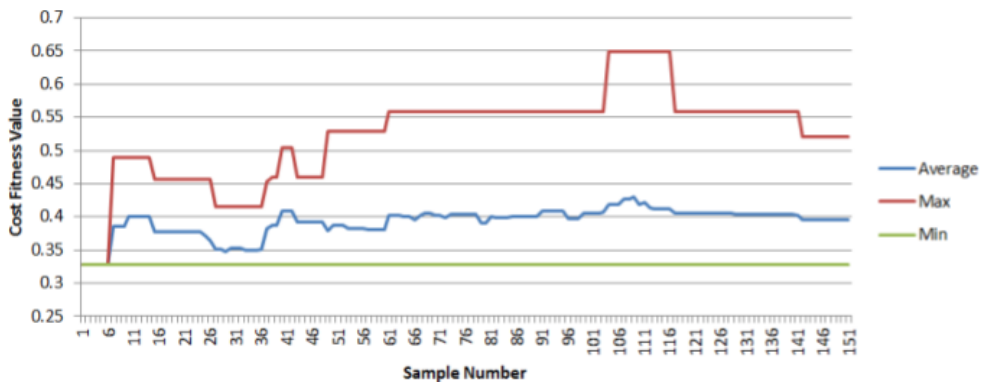


Figure 5.24: MOEA Optimal Solutions Sample Cost over Time

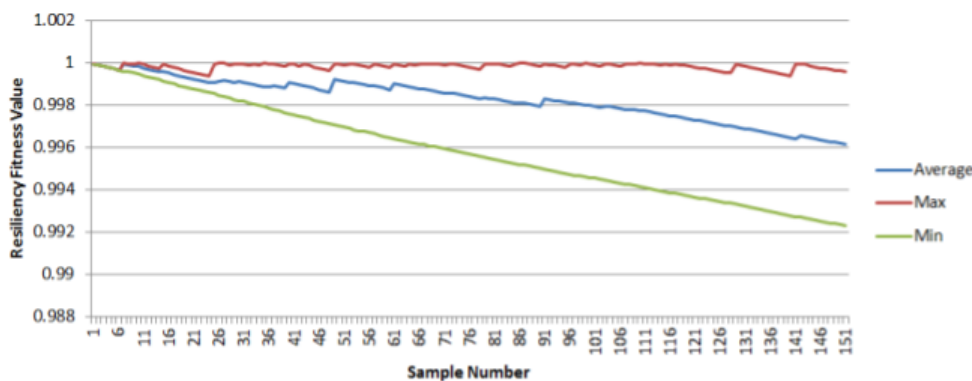


Figure 5.25: MOEA Optimal Solutions Sample Resiliency over Time

tion of cloned solutions made of three Do Nothing actions. As these solutions explore the solutions space, newer solutions with lower risk but higher costs are found. The original solution is present in all three graphs since it has the lowest cost achievable by any solutions, thus dominating in the cost fitness. However, it is constantly one of the least risk mitigating propositions.

In Figure 5.23, the solutions quickly adapt to higher risk values and solutions of various risks are found. Since the risk of suppliers and segmentation actions is itself probabilistic, the graph is fairly noisy but a clear downward trend emerges. The increased risk fitness values resulting from the higher risk of the segments yielded by the Do Nothing actions is in fact barely noticeable for the minimum risk solutions, but is evident in the average and the maximum of all solutions.

An equally revealing graph is presented in Figure 5.24. The financial consequences of the increase in disruption risk are not flagrant. It can be seen that the cost is fairly stable between samples 36 to 66, with a modest increase for the average but a bigger increase in the maximum. It is possible that the population had many individuals which were close to the Pareto-optimal front but got dominated by other solutions until samples around 36 and 46. These solutions became viable as the risk increased while other solutions which relied more on the lower risk of the Do Nothing actions were no longer optimal. The cost remained fairly stable throughout the optimization process.

In Figure 5.25, interesting patterns emerged. The original solution is present as the

Table 5.9: Fitness Values of Select Solutions in Sample 16

Id	Cost	Risk	Resiliency
7	0.3286	0.4776	0.9990
1002565	0.3888	0.3668	0.9995
1355365	0.4566	0.3262	0.9996
2037447	0.4085	0.4700	0.9999

Table 5.10: Fitness Values of Select Solutions in Sample 151

Id	Cost	Risk	Resiliency
7	0.3286	0.8263	0.9936
7417862	0.5210	0.3032	0.9964
9584668	0.5189	0.3026	0.9973
15385715	0.4079	0.4434	0.9995
15980388	0.4058	0.4835	0.9997

ever decreasing min in the graph, but the maximum remains near 1, indicating that new solutions are appearing. The maximum distance from 1 increases as solutions stabilize. The zigzag pattern could be explained as follows: as a solution discovers a new optimal point, new solutions which are mutated from it will explore areas nearby and become dominant. In other words, a single solution discovers a new region and many new solutions are created which decreases the overall resiliency of the dominant solutions. The resiliency value decreases fairly slowly as intended to not overwhelm other fitness values.

Another interesting aspect of the optimization process is the number of Pareto-optimal solutions. This number started at 1, and increased to 24, one short of the population size. The increase in Pareto-optimal individuals happened in clusters, as in, when it did increase, it did so by many individuals.

Table 5.9 has 4 of the 9 solutions in sample 16. All fitness values are fairly close and the base Do Nothing solution is present as solution 16. It has a cost of 0.329, the lowest cost attainable. These solutions can be compared to the chosen 5 solutions out of the 24 in sample 151 in Table 5.10. The base solution is still found, with Id 7. However, its risk has increased as it has no risk mitigating actions.

Table 5.11: Actions in Solutions in Sample 16

Id	Gene Length				
7	1	1	1	N/a	N/a
1002565	1	2	1	1	1
1355365	1	1	1	1	N/a
2037447	1	1	1	1	N/a

Table 5.12: Actions in Select Solutions in Sample 151

Id	Gene Length					
7	1	1	1	N/a	N/a	N/a
7417862	1	1	1	1	1	N/a
9584667	1	1	1	1	1	1
15385715	1	1	4	1	N/a	N/a
15980388	1	2	3	1	N/a	N/a

The length of genes and chromosomes for sample 16 are found in Table 5.11. The first three genes of any solution correspond to the pre-existing 3 Do Nothing actions resulting from the ships already under way. Any additional gene will have a Buy action at its base. Any additional actions in a gene is a Segment action. For example, Solution 1002565 has two Buy actions, and its second gene has 1 Segment action.

The maximum size is 5, with little segmentation. Since the risk is low, the cost of mitigating actions is severely bounding solutions to lower sizes, although some solutions have diverged. The solution 1355365 yields slightly less risky segments but at a much higher cost. However, a variety of solution lengths are present in Table 5.12 and in sample 151.

Solution 9584667 attempts to solve the disruption by creating redundancy with Buy actions. This means the solution will buy more quantities of the contract item from three different vendors, increasing the supply of items and reducing the importance of any one vessel. Solution 15980388 instead tries to solve the situation by segmenting the already existing shipments, causing a reduction in the risk of the segments that are used to transport the items. Both of these are fairly costly, as can be seen in Table 5.10, and both offer trade-offs between cost and risk reduction.

In conclusion, a reaction RMF designed for the mIoT is used in combination with a MOEA that is able to generate optimized up-to-date disruption mitigation solutions. The proposed application of the RMF methodology is able to solve a synthetic maritime supply chain disruption scenario. The features and risk model were able to capture the risk of the disruption and the system was able to synthesize a diverse set of Pareto-optimal solutions.

5.4 Supply Chain Disruption Prevention

The following section applies the RMF methodology to prevent a supply chain disruption by rerouting a ship as presented in Section 4.4. This application of the RMF is different than the others in the sense that it takes a general perspective of the risk and does not base itself on one model. Additionally, the CoA aspect of this case study is suited for the last thesis contribution, the proposed Multi-Objective Ant Colony Optimization (MO-ACO) presented in Section 3.6. While the RMF is more reactive than proactive, this application allows is more proactive by predicting risky situations and preventing them, instead of reacting only to the situation once it happens.

Since this application does not rely on a single model, the first two steps of the methodology will be skipped. This means that these two steps can be applied for any scenario and steps 3 and 4 will be the same. For example, the risk could be from active weather, an oil spill, piracy, or any other risky events that could delay a vessel and consequently cause a supply chain disruption.

5.4.1 Disruption Mitigation Prevention

A vessel's route can be discretized into a series of waypoints that vessel must go through in order to arrive at its final destination. This sequence of waypoints is fundamentally a routing solution and can be evaluated with applicable fitness functions to measure its suitability. The first fitness function, *Distance*, is the most obvious one

and is usually the de facto objective to minimize. It can be written as presented in the following equation, where $\text{haversine}(x)$ is the haversine function [118], and waypoint_i is the i th waypoint of a ship's path.

$$f_{\text{distance}}(\text{solution}) = \sum_{i=1}^{|\text{path}|} \text{haversine}(\text{waypoint}_i, \text{waypoint}_{i-1}) \quad (5.5)$$

However, if risk is also to be considered, a second fitness function should be defined. This fitness function should essentially measure the riskiness of a route. This fitness function is defined in the following equation, or the average risk of all waypoints in the path where the function risk returns the risk of a geographical point.

$$f_{\text{risk}}(\text{solution}) = \sum_{i=1}^{|\text{path}|} \text{risk}(\text{waypoint}_i) / |\text{path}| \quad (5.6)$$

However, waypoints are in latitude and longitude, a continuous space, and must be discretized if a finite sequence of waypoints is to exist. Geohashes are strings that represent geographical areas of the world with a configurable precision ranging from a magnitude of multiple hundred of kilometres, to a few metres ⁵. Consequently, geohash cells with waypoints consisting as the middle of these cells can be used.

This gridding up of the world also solves the problem of assignment of a risk to a geographical point, since risks can now be attributed to areas. These risk values themselves are not known a priori. Yet, ships around the world are mandated to broadcast AIS message periodically with their current location along with other information. Risk values can be computed from the information in these AIS messages, and would act as risk samples over the environment, revealing a risk map through dynamic information sharing.

In order to apply MO-ACO to this problem, an iterative solution construction has to be designed. This solution construction has some information that it can use to guide itself in order to decide which node might be better to visit. First, it knows

⁵Geohash. (n.d.). Retrieved July 19, 2017, from <http://geohash.org/>

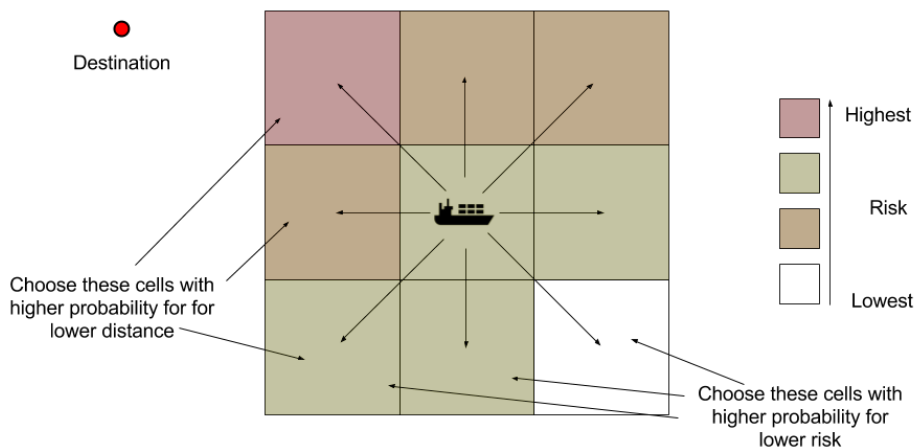


Figure 5.26: Heuristic for Next Node Visitation

that neighbouring geohash cells that reduce the distance to the destination of the ship might optimize one of the fitness functions. Second, it knows that travelling to cells that have lower risk might optimize the second fitness function. Better solutions can be formed if higher probabilities are assigned to cells with lower risk, or cells that are closer to the destination of the ship when the solution constructor decides the next cell to visit in the MO-ACO method. These two facts are sometimes in conflict, but can be resolved by the MO-ACO. This is shown in Figure 5.26. The MO-ACO algorithm is now ready to be applied.

5.4.2 Refinement and Validation

The designed simulation for this application contains a ship route and a risk map, and does not contain any agents. Therefore, only a starting point, an endpoint, and a risk map is required to validate and refine this application. The risk map itself will be generated by simulating two dimensional Gaussian distributions over an area. The area itself is the box between the start and end point of the ship's route.

In order to gauge the performance of this method, it will be compared against the NSGA-II algorithm that has the same fitness functions and solution encoding. A mutation operator that switches one waypoint with one of its feasible neighbours will be used. The crossover operator will combine segments of routes when feasible. A third algorithm that merges MO-ACO and NSGA-II will also be used, where ants are

created for every NSGA-II iteration, and the pheromones are updated per MO-ACO. An iteration of this algorithm starts with new solutions being generated via the MO-ACO algorithm and added to the population, followed by mutation and crossover, truncation back to size, and finally pheromone updates. A fourth algorithm will also be used, where the population will be initialized via MO-ACO and exploited with NSGA-II.

This concludes the application of the RMF methodology for maritime supply chain disruption prevention by applying the appropriate data structures, operators, and fitness functions for the MO-ACO method, as well as a simulation to test it.

5.4.3 Results of MO-ACO for Risk-Aware Ship Routing

An α and β of 1 was used for the MO-ACO algorithm, along with an evaporation rate of 0.25. Five ants were generated per iteration for 50 iterations. A mutation rate of 0.25 and a crossover probability of 0.5 were set for the NSGA-II, with a population of 120 solutions. The NSGA-II was executed for 200 iterations.

The third algorithm, known as the Mixed algorithm, combined both the MO-ACO and NSGA-II algorithms by generating new solutions during each NSGA-II iteration. This algorithm had the same parameters as MO-ACO and NSGA-II, except a population size of 80 was used and 3 new solutions were generated per iteration. This algorithm stopped after 70 iterations.

Finally, the last algorithm, known as the Sequential algorithm, used MO-ACO to generate solutions, then NSGA-II to exploit them. This algorithm used the same parameters as MO-ACO and NSGA-II, except that it had a population size of 100. The MO-ACO algorithm terminated after 30 iterations, and its solutions were the initial population of the NSGA-II algorithm that ran for 120 iterations. The experiment was repeated 90 times with 15 Gaussian risk distributions to generate appropriate risk maps.

The mean number of solutions with 95% confidence intervals are shown in Figure 5.27.

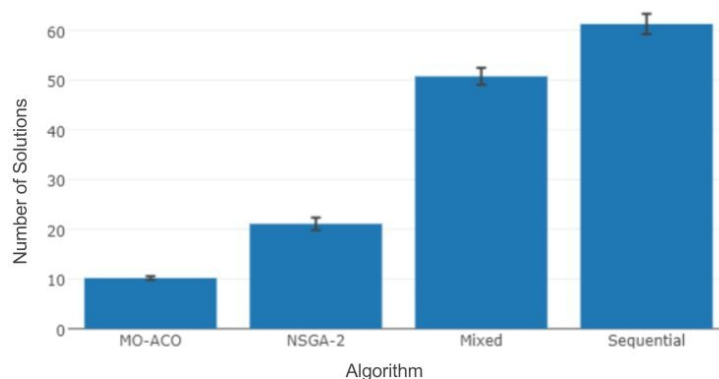


Figure 5.27: Number of Solutions per Algorithm

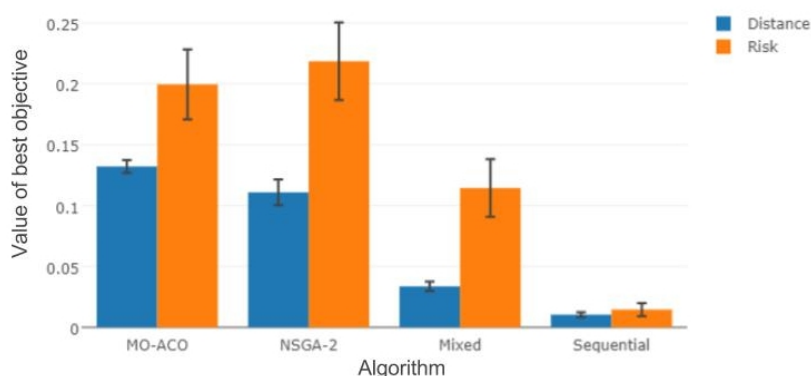


Figure 5.28: Solution Spread Distance to Best

It can be seen that the MO-ACO returns the least number of solutions, while the Sequential algorithm returns the most. The MO-ACO algorithm does not exploit solutions well, so it doesn't return solutions close to existing ones, while the Sequential algorithm is able to do so with the help of NSGA-II. A high number of Pareto-optimal solutions is not necessarily better, but it can indicate a better exploration of the Pareto-front.

Spreading is another metric that measures the best value found for each objective. Multi-objective algorithms should strive to offer a range of solutions with tradeoffs between their objectives. Figure 5.28 shows the mean difference between the best solutions found for each objective per experiment and the best solutions found for each objective by the specific algorithm. The Sequential algorithm again dominates the other algorithms, with the Mixed algorithm in second place. This indicates that the sequential algorithm has a higher number of solutions with a better spread, which means that it might offer the best convergence to the Pareto-front.

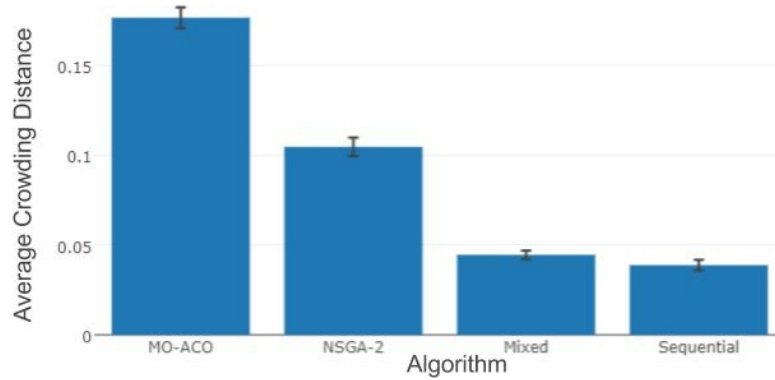


Figure 5.29: Average Crowding Distance

A third metric, *average crowding distance*, is useful when measuring the quality of solutions. Crowding distance is a metric that measures how close solutions are to each other in terms of fitness. If a set of solutions has a low average crowding distance, it may indicate that a certain area of the solution space was exploited but the rest was not explored. The average crowding distance for each algorithm is found in Figure 5.29. We see MO-ACO has the best crowding distance, which degrades for every algorithm afterwards. However, this should have been expected since the number of solutions increases for each algorithm in the order they were presented. Consequently, this metric is not very useful in this scenario.

A fourth metric, *hypervolume* [119], is a metric that measures the area, or volume for problems of dimensions higher than 2, covered by a set of solutions relative to a reference point. Figure 5.30 gives the mean distance to the best hypervolume found per experiment. The Sequential algorithm is evidently better than the three others, with a hypervolume that is, on average, an order of magnitude better. The Mixed algorithm performs about half an order of magnitude, while the MO-ACO algorithm and the NSGA-II algorithms have equal performance.

The comparison between Pareto fronts per experiment is equally telling. Figure 5.31 illustrates the Pareto-front for each algorithm for the 33rd (out of 90) experiment repetition. This figure represents a situation that is seen in many similar figures for the other experiment. The Sequential algorithm performs best with more solutions, better spread, and a better hypervolume, followed by the Mixed algorithm. The

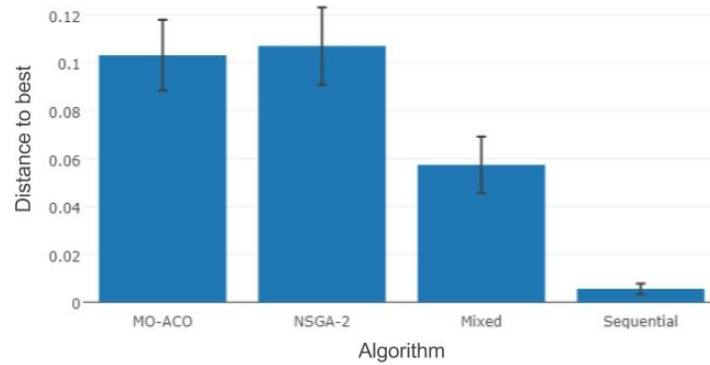


Figure 5.30: Hypervolume Distance to Best Solution Found

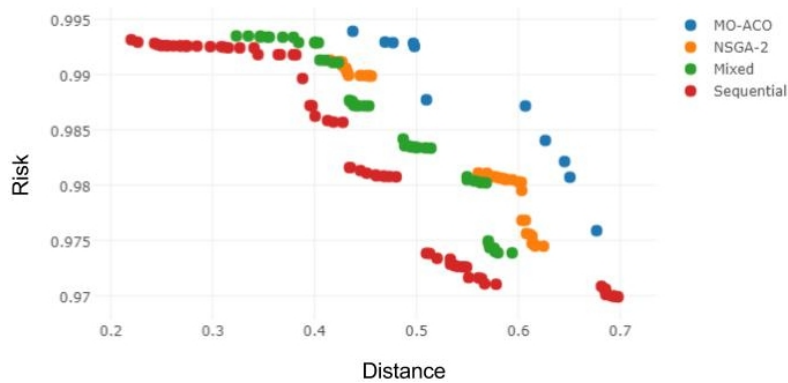


Figure 5.31: Algorithm Pareto-Fronts for Experiment 33

NSGA-II algorithm provides solutions that lack exploration, but performs better than the MO-ACO algorithm, which yields better exploration but lacks exploitation. This again demonstrates that the hybrid Sequential algorithm performs best.

Figure 5.32, Figure 5.33, Figure 5.34, and Figure 5.35 illustrate the solutions in the environment, along with the portion of the risk map that were explored. It can be seen in Figure 5.32 that the MO-ACO algorithm suffers from bad exploitation of its existing solutions, but adequately explores and finds new paths. The second figure, Figure 5.33 shows the results of NSGA-II. The environment is not well explored, and only three paths seems to have been found, but they have been well-optimized. The results of the Mixed algorithm in Figure 5.34 demonstrate that this algorithm is able to better explore and exploit solutions. New solutions found by the MO-ACO portion of the algorithm that might lead to better routes are highlighted. Finally, the results of the Sequential algorithm in Figure 5.35 confirm yet again that this last algorithm outperforms all the others, as it gives the best tradeoff in terms of exploitation and

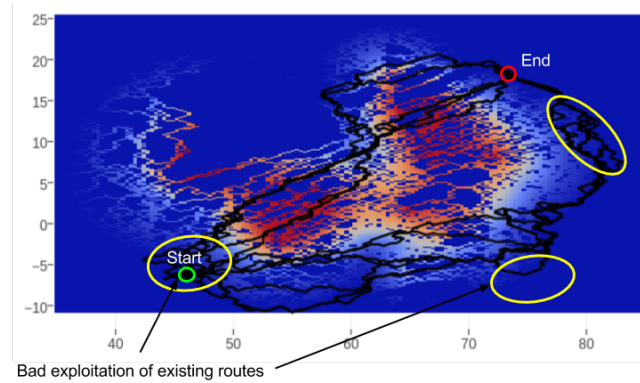


Figure 5.32: MO-ACO Solutions for Experiment 10

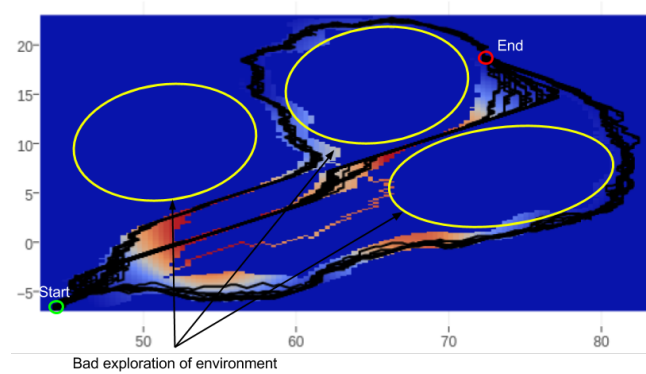


Figure 5.33: NSGA-II Solutions for Experiment 10

exploration.

In summary, optimized routes in terms of distance and risk were able to be found by the proposed algorithms. While the proposed MO-ACO algorithm was unable to significantly outperform the NSGA-II algorithm, the two hybrid methods that used it were able to do so. In particular, the Sequential algorithm significantly outperformed the other three algorithms in terms of number of solutions returned, spreading, and

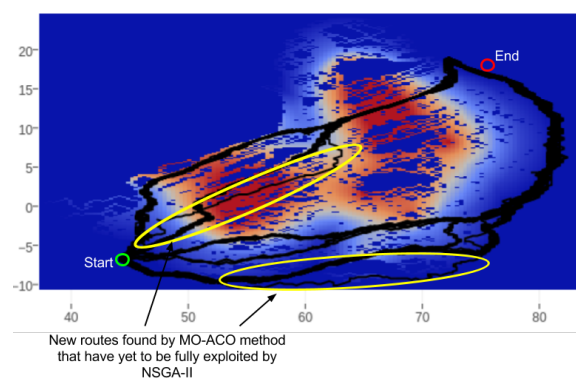


Figure 5.34: Mixed Algorithm Solutions for Experiment 10

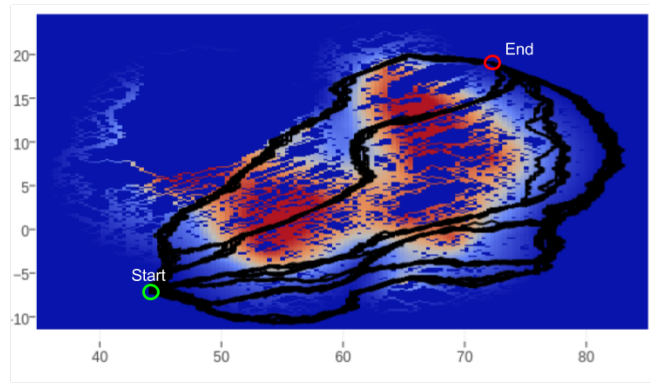


Figure 5.35: Sequential Algorithm Solutions for Experiment 10

hypervolume, which was confirmed when comparing the Pareto-fronts of solutions, as well as when visualizing the solutions.

5.5 Chapter Summary

This chapter provided four applications of the methodology for parts or all of the RMF in each CIP case study presented in Chapter 4, along with simulation results and appropriate discussions. Each of the applications made use of the contributions presented in Chapter 3 in order to better detect, identify, and mitigate risk. The first application made use of the two presented novel auction protocols to help the RMF maintain perimeter coverage with an RSN around a CIP. The second application tackled the problem of protecting ports against maritime smuggling of any kind, such as for weapons, drugs, or people, by including UAVs to create hybrid RSNs.

The third and fourth applications attempted to resolve the problem of disruptions in maritime supply chains, a major component of today's economic and defence infrastructure. The former did so by using a maritime IoT with the rRMF and continuous optimization to detect possible disruptions, then determine applicable mitigating solutions. The latter prevents disruptions by providing a range of solutions that make up the tradeoff between distance and risk, thus giving the ability to reduce the risk while creating expected and controlled delays. These two last applications were much more proactive in nature than the first two, which reacted to events. While the first

two applications concentrated on mitigation the event as it happens, the third and fourth application attempted to predict risk and propose solutions to events that have not yet happened.

The experimental results provided valuable insight into the designed systems, and each highlighted the contributions as proposed in Chapter 3. In the first section of this chapter, it was found that the English protocol yielded better coalition, but the FPSB protocol was best for time and communication costs, with the Dutch-Japanese algorithm falling somewhere in the middle.

The addition of UAVs into RSNs was validated in section 5.2, where the monitoring tasks were visualized. The rRMF for the IoT and continuous optimization were shown to have success in helping to mitigate supply chain disruption in Section 5.3. Finally, different algorithms for risk-aware ship routing were compared in Section 6.4, with the Sequential algorithm proving to be about an order of magnitude better than the NSGA-II and MO-ACO algorithm, with the Mixed algorithm falling somewhere in between.

The next chapter will provide a final discussion on current research trends, as well as future works, and a final conclusion to the thesis.

6 Trends, Future Work, and Conclusion

This last chapter will conclude the thesis by first discussing current research trends that were uncovered during the research for this thesis. Then, a section will discuss future works and extensions of the contributions proposed in this work. Lastly, closing remarks and a final conclusion will be provided.

6.1 Research Trends

During the research done to construct this thesis, a few research trends pertinent to this research were noted. First, CI is an active field of study with multiple avenues that are being researched. Multiple algorithms, particularly hybrid algorithms, are being developed with increasing complexity and efficiency. These algorithms are crucial tools of the risk management framework (RMF), and are often bottlenecks of the system in terms of computation complexity, and consequently an improvement is welcome.

Research in on one of these techniques, artificial neural networks, has seen a resurgence with Deep Learning [44]. These artificial neural networks have proved useful when applied for reinforcement learning, another machine learning method that is being renewed after its introduction in the 1990s [45]. Many of the decisions in the RMF hinge on comparing multi-objective solutions and having an operator manually pick

one. With reinforcement learning, new work into automatizing this process could be done.

Research into soft-data sources and comprehension is increasing. Natural language processing and linked data sources are methods to integrate new data and understand it in similar ways that humans would. With this ability, new soft data ingestion sources such as news articles, textual reports, twitter feeds, and other textual data can be considered and added into the RMF as data streams for the risk models. This last point is important as the type of data is changing.

The internet of things (IoT) is a new concept that is hard to define and is rapidly changing. The number of connected devices is increasing and accelerating, each bringing new sources for all sorts of data. This data is sometimes unreliable, inaccurate, and incomplete. Fusing multiple data streams to get a better awareness of situations becomes inevitable. However, doing so would enable deep and comprehensive knowledge that would be invaluable throughout the RMF.

6.2 Future Work

Every research item is a work in progress, and some work is left for future consideration. Along this thesis, a few have been mentioned. This section will first discuss them, then it will present other research avenues that can be explored in the future.

The presented auction protocols were tested in scenarios with little variations compared to the types of networks that exist in real settings. Consequently, it would be better to simulate the protocols in scenarios that better represent the plethora of WSNs that exist. A more complex RSN node model should be used in the simulations. Additionally, many protocol parameters were set experimentally. Work should be done to determine appropriate stopping criterion instead of using time or coalition size in order to eliminate parameters. This can be accomplished perhaps by using one or more fitness functions for the coalition and stopping criterion based on them.

When integrating UAVs into the RSN with a risk-based perspective, the communication aspect between UAVs was not considered in depth, with only a simple model of connectivity between nodes being used. Yet, communication is one of the principal components of aerial networks [120]. Consequently, more work should be done to consider these problems in the optimizations for monitoring. Research into aerial sensor networks and UAV communication models could be included in the connectivity fitness function.

In the future, research should be done into using machine learning and other tools to automate the methodology as presented. For example, from a set of example risky situations, the risk model could be determined without having to manually determine the intricate details of the FIS model, such as membership function parameters and fuzzy rules. This would also reduce the importance of the fourth step of refinement and validation, since the model would be less susceptible to human bias and could be optimized as situations require. This would alleviate some concerns such as finding better disruption risk models in Section 6.3.

A second improvement would be the use of machine learning such as ANNs to determine appropriate weights to choose multi-objective solutions. Human operators base themselves on acquired knowledge to know that a certain solution is better than another in certain contexts. Consequently, a model that links the environment's state to weights would point to certain solutions that should be better researched. This would allow for faster mitigation, and for an automated process that can be repeated as rapidly as the algorithmic complexity allows.

A third research opportunity lies in the third step of the proposed methodology. Most actions are fundamentally composed of combinations of simpler ones, with some base actions that can be considered as atomic. A set of basic actions could be defined as building blocks for more complicated ones. This would allow for the course of action module to be able to find solutions in a wide range of contexts. This would require greater environment models, and fitness functions that are more universal and fundamental. Research in this area would result in better risk mitigation options for

all the works described in this thesis.

Finally, a last future work avenue lies in the data intricacies of the IoT. Communication between any two entities is impossible unless they understand the language each other is speaking, that is the format, protocols, and semantics of the data that each entity is transmitting. Research should be done in creating a system that is able to automatically infer the type of data, and the information that is present in the data sent from one entity. A first step towards this goal would be to build a comprehensive and flexible taxonomy. Such a system would allow for smarter ingestion of data, and would allow for deep, complex inferences to be made from the existing data. These inferences could prove invaluable to building smarter systems.

6.3 Conclusion

This thesis has proposed a methodology to apply the RMF in order to identify, detect, and mitigate risks related to critical infrastructure protection. It has additionally proposed 5 other main contributions, which led to 3 published conference papers [13, 15, 12] and 2 presentations [14, 16]. The first is the design and implementation of a reactive, distributed version of the reactive RMF specifically geared for the IoT [12]. The second contribution contributed two novel auction protocols that can be used in a pre-optimization attempt to determine a set of robots best suited for the multi-robot task allocation step [13]. The third is a method to integrate unmanned aerial vehicles (UAVs) into robotic sensor networks via a risk-based perspective [15]. The fourth proposed a CI technique to continuously optimize risk mitigating solutions for highly dynamic environments [12]. The fifth is a multi-objective optimization algorithm based on ant-colony optimization, another CI technique [16]. Finally, a minor contribution is the detailing of 4 case studies in critical infrastructure protection with simulations [13, 15, 12, 16]. These scenarios each describe a situation where a decision support system would prove invaluable. The methodology was followed, designing such a system based on a suitable network of nodes combined with the

RMF.

Many more situations remain where decision support systems can be used to solve complex problems that are undecipherable from a human perspective. Along with the trends and future works discussed above, it is clear that much more work is left to be done. Nonetheless, this thesis has helped to push the existing research into an intelligent decision support system into new areas full of new opportunities by combining cutting edge research from a variety of different fields.

Bibliography

- [1] M. R. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human factors*, vol. 37, no. 1, pp. 32–64, 1995.
- [2] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [3] M. Mazo and P. Tabuada, “Decentralized event-triggered control over wireless sensor/actuator networks,” *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456–2461, 2011.
- [4] J. Kacprzyk and W. Pedrycz, *Springer handbook of computational intelligence*. Springer, 2015.
- [5] S. Tran and L. Davis, “Event modeling and recognition using markov logic networks,” in *Proceedings of the European Conference on Computer Vision*. Marseille, France: French National Institute for computer science and applied mathematics, October 2008, pp. 610–623.
- [6] L. A. Zadeh, “Is there a need for fuzzy logic?” *Information Sciences*, vol. 178, no. 13, pp. 2751–2779, 2008.
- [7] R. Falcon, A. Nayak, and R. Abielmona, “An evolving risk management framework for wireless sensor networks,” in *Proceedings of the International Conference on Computational Intelligence for Measurement Systems and Applications*. Ottawa, CA: IEEE, September 2011, pp. 1–6.
- [8] J. McCausland, R. Abielmona, R. Falcon, A.-M. Cretu, and E. M. Petriu, “On the role of multi-objective optimization in risk mitigation for critical infras-

- tructures with robotic sensor networks,” in *Proceedings of the Conference on Genetic and Evolutionary Computation*. Vancouver, CA: IEEE, July 2014, pp. 1269–1276.
- [9] R. Falcon, R. Abielmona, S. Billings, A. Plachkov, and H. Abbass, “Risk management with hard-soft data fusion in maritime domain awareness,” in *Proceedings of the Symposium on Computational Intelligence for Security and Defense Applications*. Cau Giay, VN: IEEE, December 2014, pp. 1–8.
- [10] R. Falcon, B. Desjardins, R. Abielmona, and E. Petriu, “Context-driven dynamic Risk Management for maritime domain awareness,” in *Proceedings of the Symposium Series on Computational Intelligence*. Athens, GR: IEEE, December 2016, pp. 1–8.
- [11] K. Deb, “Multi-objective optimization,” in *Search methodologies*. Springer, 2014, pp. 403–449.
- [12] N. Primeau, R. Falcon, R. Abielmona, and E. Petriu, “Continuous Risk-Aware Response Generation for Maritime Supply Chain Disruption Mitigation,” in *Proceedings of the International Conference on Distributed Computing in Sensor Systems*. Ottawa, CA: IEEE, June 2017.
- [13] N. Primeau, R. Falcon, R. Abielmona, V. Groza, and E. Petriu, “Improving task allocation in risk-aware robotic sensor networks via auction protocol selection,” in *Proceedings of the International Conference on Intelligent Engineering Systems*. Budapest, HU: IEEE, June 2016, pp. 21–26.
- [14] ———, “Improving task allocation in risk-aware robotic sensor networks via auction protocol selection,” July 2016, Presented at the Canadian Tracking and Fusion Group Workshop.
- [15] N. Primeau, R. Abielmona, R. Falcon, and E. Petriu, “Maritime smuggling detection and mitigation using risk-aware hybrid robotic sensor networks,” in *Proceedings of the Conference on Cognitive and Computational Aspects of Situation Management*. Savannah, US: IEEE, March 2017, pp. 1–7.

- [16] N. Primeau, R. Falcon, R. Abielmona, and E. Petriu, "Risk-Aware vessel Routing using Multi-Objective Ant Colony Optimization," July 2017, Presented at the International Conference on Multiple Criteria Decision Making. [Online]. Available: http://sites.telfer.uottawa.ca/mcdm2017/files/2016/05/Scientific_Program_July_11.pdf
- [17] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [18] A. Nayak and I. Stojmenovic, *Wireless sensor and actuator networks: algorithms and protocols for scalable coordination and data communication*. John Wiley & Sons, 2010.
- [19] S. R. Karedla and S. Anuradha, "Localization Error Minimization using GA in MWSN," *International Journal of Advanced Electrical and Electronics Engineering*, vol. 2, no. 4, pp. 17–24, 2013.
- [20] N. Irfan, M. Bolic, M. C. Yagoub, and V. Narasimhan, "Neural-based approach for localization of sensors in indoor environment," *Telecommunication Systems*, vol. 44, no. 1, pp. 149–158, 2010.
- [21] R. Qi, S. Li, T. Ma, and F. Qian, "Localization with a mobile anchor using abcg hybrid algorithm in wireless sensor networks," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 16, no. 6, pp. 741–747, 2012.
- [22] J. S. Liu, S. Y. Wu, and K. M. Chiu, "Path planning of a data mule in wireless sensor network using an improved implementation of clustering-based genetic algorithm," in *Proceedings of the Symposium on Computational Intelligence in Control and Automation*. Singapore, SG: IEEE, April 2013, pp. 30–37.
- [23] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [24] Y. Li and L. E. Parker, "Intruder detection using a wireless sensor network

- with an intelligent mobile robot response,” in *Proceedings of SoutheastCon*. Huntsville, US: IEEE, April 2008, pp. 37–42.
- [25] X. Wu, Y. Wang, and Y. Hu, “A swarm intelligent algorithm based route maintaining protocol for mobile sink wireless sensor networks,” *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [26] B. Desjardins, “Reliable multiple robot-assisted sensor relocation using multi-objective optimization,” Master’s thesis, University of Ottawa, 7 2016.
- [27] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer, 2003, vol. 53.
- [28] K. Deb and H. Gupta, “Searching for robust pareto-optimal solutions in multi-objective optimization,” *Evolutionary Multi-criterion Optimization*, pp. 150–164, 2005.
- [29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [30] X.-S. Yang and S. Deb, “Cuckoo Search via Levy Flights,” in *Proceedings of the World Congress on Nature & Biologically Inspired Computing*. Coimbatore, India: IEEE, December 2009, pp. 210–214.
- [31] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.
- [32] C. A. C. Coello and N. C. Cortés, “Solving multiobjective optimization problems using an artificial immune system,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [33] D. Simon, “Biogeography-based optimization,” *IEEE transactions on evolutionary computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [34] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical

- function optimization: artificial bee colony (abc) algorithm,” *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [35] S. Das, A. Biswas, S. Dasgupta, and A. Abraham, “Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications,” *Foundations of Computational Intelligence Volume 3*, pp. 23–55, 2009.
- [36] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, “Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications,” *Artificial Intelligence Review*, pp. 1–33, 2014.
- [37] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, “The bees algorithm—a novel tool for complex optimisation,” in *Proceedings of the Virtual Internatinal Conference on Intelligent Production Machines and Systems P(3-14 July 2006)*. I*PROMS, July 2011, pp. 454–459.
- [38] B. Wu, C. Qian, W. Ni, and S. Fan, “The improvement of glowworm swarm optimization for continuous optimization problems,” *Expert Systems with Applications*, vol. 39, no. 7, pp. 6335–6342, 2012.
- [39] L. A. Zadeh, “Fuzzy logic= computing with words,” *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 2, pp. 103–111, 1996.
- [40] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, no. 1, pp. 116–132, 1985.
- [41] S.-G. Cao, N. W. Rees, and G. Feng, “Mamdani-type fuzzy controllers are universal fuzzy controllers,” *Fuzzy Sets and Systems*, vol. 123, no. 3, pp. 359–367, 2001.
- [42] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
- [43] B. Yegnanarayana, *Artificial neural networks*. PHI Learning Pvt. Ltd., 2009.
- [44] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

- [45] H. Van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning.” in *Proceedings of the Conference on Artificial Intelligence*. Phoenix, US: AAAI, February 2016, pp. 2094–2100.
- [46] G. A. Carpenter, S. Grossberg, and D. B. Rosen, “Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system,” *Neural Networks*, vol. 4, no. 6, pp. 759–771, 1991.
- [47] J. Robinson, S. Sinton, and Y. Rahmat-Samii, “Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna,” in *Proceedings of the International Symposium on Antennas and Propagation Society International*, vol. 1. San Antonio, US: IEEE, August 2002, pp. 314–317.
- [48] I. Alaya, C. Solnon, and K. Ghedira, “Ant colony optimization for multi-objective optimization problems,” in *Proceedings of 19th IEEE International Conference on Tools with Artificial Intelligence*, vol. 1. Patras, GR: IEEE, October 2007, pp. 450–457.
- [49] A. Jankowska, M. C. Schut, and N. Ferreira-Schut, “A wireless actuator-sensor neural network for evacuation routing,” in *Proceedings of the International Conference on Sensor Technologies and Applications*. Athens, GR: IARIA, June 2009, pp. 139–144.
- [50] H. M. La, T. H. N. T. H. Nguyen, C. H. N. C. H. Nguyen, and H. N. N. H. N. Nguyen, “Optimal flocking control for a mobile sensor network based a moving target tracking,” in *Proceedings of the International Conference on Systems Man and Cybernetics*. San Antonio, US: IEEE, October 2009, pp. 4843–4848.
- [51] P. S. P. Song, J. L. J. Li, K. Li, and J. L. J. Liang, “Hybrid multi-robot path planning strategy based on RRT and fuzzy control in wireless sensor network,” in *Proceedings of the International Conference on Robotics and Biomimetics*. Guilin CN: IEEE, December 2009, pp. 877–881.
- [52] Y.-F. Hu, Y.-S. Ding, L.-H. Ren, K.-R. Hao, and H. Han, “An endocrine co-

- operative particle swarm optimization algorithm for routing recovery problem of wireless sensor networks with multiple mobile sinks,” *Information Sciences*, vol. 300, pp. 100–113, 2015.
- [53] J. Yang, F. Liu, J. Cao, and L. Wang, “Discrete particle swarm optimization routing protocol for wireless sensor networks with multiple mobile sinks,” *Sensors*, vol. 16, no. 7, p. 1081, 2016.
- [54] J. R. Srivastava and T. Sudarshan, “A genetic fuzzy system based optimized zone based energy efficient routing protocol for mobile sensor networks (ozeep),” *Applied Soft Computing*, vol. 37, pp. 863–886, 2015.
- [55] S. Sarangi and S. Kar, “Genetic algorithm based mobility aware clustering for energy efficient routing in wireless sensor networks,” in *Proceedings of the International Conference on Networks*. Delft, NL: IEEE, December 2011, pp. 1–6.
- [56] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, “Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks,” *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4576–4586, 2015.
- [57] J.-h. Zhong and J. Zhang, “Ant colony optimization algorithm for lifetime maximization in wireless sensor network with mobile sink,” in *Proceedings of the Conference on Genetic and Evolutionary Computation*. Philadelphia, US: ACM, July 2012, pp. 1199–1204.
- [58] W. Cai, M. Chen, T. Hara, and L. Shu, “GA-MIP: Genetic algorithm based multiple Mobile Agents itinerary planning in wireless sensor networks,” in *Proceedings of the Annual Wireless Internet Conference*. Singapore, SG: ICST, March 2010, pp. 1–8.
- [59] W. Yiyue, L. Hongmei, and H. Hengyang, “Wireless Sensor Network Deployment Using an Optimized Artificial Fish Swarm Algorithm,” in *Proceedings of*

- the International Conference on Computer Science and Electronics Engineering*. Hangzhou, CN: IEEE, March 2012, pp. 90–94.
- [60] A. Ray and D. De, “An energy efficient sensor movement approach using multi-parameter reverse glowworm swarm optimization algorithm in mobile wireless sensor network,” *Simulation Modelling Practice and Theory*, vol. 62, pp. 117–136, 2016.
- [61] R. Kuawattanaphan, T. Kumrai, and P. Champrasert, “Wireless sensor nodes redeployment using a multiobjective optimization evolutionary algorithm,” in *Proceedings of the International Conference of IEEE Region 10*. Xi’an, CN: IEEE, October 2013, pp. 1–6.
- [62] X. Wang, S. Wang, and J.-J. Ma, “An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment,” *Sensors*, vol. 7, no. 3, pp. 354–370, 2007.
- [63] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, “Rearrangement of mobile wireless sensor nodes for coverage maximization based on immune node deployment algorithm,” *Computers & Electrical Engineering*, vol. 43, pp. 76–89, 2015.
- [64] F. Nematy, A. M. Rahmani, M. Teshnelab, and N. Rahmani, “Ant colony based node deployment and search in wireless sensor networks,” in *Proceedings of the International Conference on Computational Intelligence and Communication Networks*, Bhopal, India, November 2010, pp. 363–366.
- [65] B. J. Farahani, H. Ghaffarian, and M. Fathy, “A fuzzy based priority approach in mobile sensor network coverage,” *International Journal of Recent Trends in Engineering*, vol. 2, no. 1, 2009.
- [66] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, “Dynamic deployment of wireless sensor networks by biogeography based optimization algorithm,” *Journal of Sensor and Actuator Networks*, vol. 1, no. 2, pp. 86–96, 2012.
- [67] B. Desjardins, R. Falcon, R. Abielmona, and E. Petriu, “A multi-objective

- optimization approach to Reliable Robot-Assisted Sensor Relocation,” in *Proceedings of the Congress on Evolutionary Computation*. Sendai, JP: IEEE, May 2015, pp. 956–964.
- [68] S. Singh and A. Kumar, “Novel optimal deployment of sensor nodes using bio inspired algorithm,” in *Proceedings of the International Conference on Advanced Communication Control and Computing Technologies*. Langkawi, MY: IEEE, May 2014, pp. 847–851.
- [69] R. V. Kulkarni and G. K. Venayagamoorthy, “Bio-inspired algorithms for autonomous deployment and localization of sensor nodes,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6, pp. 663–675, 2010.
- [70] H. Bao, B. Zhang, C. Li, and Z. Yao, “Mobile anchor assisted particle swarm optimization (pso) based localization algorithms for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 12, no. 15, pp. 1313–1325, 2012.
- [71] J. Huanxiang, W. Yong, and T. Xiaoling, “Localization algorithm for mobile anchor node based on genetic algorithm in wireless sensor network,” in *Proceedings of the International Conference on Intelligent Computing and Integrated Systems*. Guilin, China: IEEE, October 2010, pp. 40–44.
- [72] M. Gholami, N. Cai, and R. Brennan, “An artificial neural network approach to the problem of wireless sensors network localization,” *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 1, pp. 96–109, 2013.
- [73] D. Herrero and H. Martínez, “Range-only fuzzy voronoi-enhanced localization of mobile robots in wireless sensor networks,” *Robotica*, vol. 30, no. 7, pp. 1063–1077, 2012.
- [74] R. Falcon and R. Abielmona, “A response-aware risk management framework for search-and-rescue operations,” in *Proceedings of the Congress on Evolutionary Computation*. Brisbane, AU: IEEE, June 2012, pp. 1–8.

- [75] J. McCausland, G. Di Nardo, R. Falcon, R. Abielmona, V. Groza, and E. Petriu, “A proactive risk-aware robotic sensor network for Critical Infrastructure Protection,” in *Proceedings of the International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications*. Milan, IT: IEEE, July 2013, pp. 132–137.
- [76] R. Falcon, R. Abielmona, and S. Billings, “Risk-driven intent assessment and response generation in maritime surveillance operations,” in *Proceedings of the International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*. Orlando, US: IEEE, March 2015, pp. 151–157.
- [77] D. Helbing, “Agent-based modeling.” Springer, 2012, pp. 25–70.
- [78] A. Drogoul, D. Vanbergue, and T. Meurisse, “Multi-agent based simulation: Where are the agents?” in *Proceedings of the International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Bologna: Springer, July 2002, pp. 1–15.
- [79] M. Andersson and R. Johansson, “Multiple sensor fusion for effective abnormal behaviour detection in counter-piracy operations,” in *Proceedings of the International Waterside Security Conference*. Carrara, IT: IEEE, November 2010, pp. 1–7.
- [80] A. Filippoupolitis, L. Hey, G. Loukas, E. Gelenbe, and S. Timotheou, “Emergency response simulation using wireless sensor networks,” in *Proceedings of the International Conference on Ambient Media and Systems*. Quebec, CA: ICST, February 2008, p. 21.
- [81] M. Raberto, S. Cincotti, S. M. Focardi, and M. Marchesi, “Agent-based simulation of a financial market,” *Physica A: Statistical Mechanics and its Applications*, vol. 299, no. 1, pp. 319–327, 2001.
- [82] M. A. Niazi and A. Hussain, “A novel agent-based simulation framework for

- sensing in complex adaptive environments,” *IEEE Sensors Journal*, vol. 11, no. 2, pp. 404–412, 2011.
- [83] R. Baldoni, M. Contenti, and A. Virgillito, *The Evolution of Publish/Subscribe Communication Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 137–141.
- [84] B. P. Gerkey and M. J. Matarić, “A formal analysis and taxonomy of task allocation in multi-robot systems,” *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [85] H. C. L. H. C. Lau and L. Z. L. Zhang, “Task allocation via multi-agent coalition formation: taxonomy, algorithms and complexity,” in *Proceedings of the International Conference on Tools with Artificial Intelligence*. Sacramento, US: IEEE, November 2003, pp. 346–350.
- [86] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, “Market-based multirobot coordination: A survey and analysis,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.
- [87] R. P. McAfee and J. McMillan, “Auctions and bidding,” *Journal of Economic Literature*, vol. 25, no. 2, pp. 699–738, 1987.
- [88] B. P. Gerkey and M. J. Mataric, “Sold!: Auction methods for multirobot coordination,” *IEEE transactions on robotics and automation*, vol. 18, no. 5, pp. 758–768, 2002.
- [89] E. Cantú-Paz, “A survey of parallel genetic algorithms,” *Calculateurs paralleles, reseaux et systems repartis*, vol. 10, no. 2, pp. 141–171, 1998.
- [90] N. Atay and B. Bayazit, “Mobile wireless sensor network connectivity repair with k-redundancy,” *Algorithmic Foundation of Robotics VIII*, pp. 35–49, 2009.
- [91] D. Angus and C. Woodward, “Multiple objective ant colony optimisation,” *Swarm intelligence*, vol. 3, no. 1, pp. 69–85, 2009.
- [92] M. Dorigo, V. Maniezzo, A. Colorni, and V. Maniezzo, “Positive feedback as a search strategy,” 1991.

- [93] M. Dorigo and L. M. Gambardella, “Ant colony system: a cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [94] T. Stutzle and H. Hoos, “MAX MIN Ant System and Local Search for the Traveling Salesman Problem,” in *Proceedings of the International Conference on Evolutionary Computation*. Indianapolis, US: IEEE, April 1997, pp. 309–314.
- [95] C. Garcia-Martinez, O. Cordon, and F. Herrera, “An empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP,” in *Proceedings of the International Workshop on Ant Colony Optimization and Swarm Intelligence*. Brussels, BE: Springer, September 2004, pp. 61–72.
- [96] K. Van Moffaert and A. Nowé, “Multi-objective reinforcement learning using sets of pareto dominating policies,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3483–3512, 2014.
- [97] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pp. 95–100, 2001.
- [98] J. Chen, M. Díaz, L. Llopis, B. Rubio, and J. M. Troya, “A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection,” *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1225–1239, 2011.
- [99] H. Karvonen, J. Suhonen, J. Petäjajarvi, M. Hämäläinen, M. Hännikäinen, and A. Pouttu, “Hierarchical architecture for multi-technology wireless sensor networks for critical infrastructure protection,” *Wireless Personal Communications*, vol. 76, no. 2, pp. 209–229, 2014.
- [100] K. Sohraby, D. Minoli, and T. Znati, *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons, 2007.
- [101] N. Haphuriwat, V. M. Bier, and H. H. Willis, “Deterring the smuggling of nu-

- clear weapons in container freight through detection and retaliation,” *Decision Analysis*, vol. 8, no. 2, pp. 88–102, 2011.
- [102] L. Joossens and M. Raw, “Cigarette smuggling in europe: who really benefits?” *Tobacco Control*, vol. 7, no. 1, pp. 66–71, 1998.
- [103] G. A. Antonopoulos and J. Winterdyk, “The smuggling of migrants in greece: An examination of its social organization,” *European Journal of Criminology*, vol. 3, no. 4, pp. 439–461, 2006.
- [104] J. van Laere and M. Nilsson, “Evaluation of a workshop to capture knowledge from subject matter experts in maritime surveillance,” in *Proceedings of the International Conference on Information Fusion*. Seattle, US: IEEE, August 2009, pp. 171–178.
- [105] R. O. Lane, D. a. Nevell, S. D. Hayward, and T. W. Beaney, “Maritime anomaly detection and threat assessment,” in *Proceedings of the International Conference on Information Fusion*. Edinburgh, UK: IEEE, July 2010, pp. 1–8.
- [106] B. van den Broek, A. Smith, E. den Breejen, and I. van de Voorde, “Inference of vessel intent and behaviour for maritime security operations,” in *Proceedings of SPIE Security+ Defence*, vol. 9248. Amsterdam, NL: International Society for Optics and Photonics, September 2014, pp. 92 480E–92 480E–12.
- [107] C. W. Craighead, J. Blackhurst, M. J. Rungtusanatham, and R. B. Handfield, “The severity of supply chain disruptions: design characteristics and mitigation capabilities,” *Decision Sciences*, vol. 38, no. 1, pp. 131–156, 2007.
- [108] M. Fujita and J.-F. Thisse, “Globalization and the evolution of the supply chain: who gains and who loses?” *International Economic Review*, vol. 47, no. 3, pp. 811–836, 2006.
- [109] C. S. Tang, “Robust strategies for mitigating supply chain disruptions,” *International Journal of Logistics: Research and Applications*, vol. 9, no. 1, pp. 33–45, 2006.

- [110] S. Chopra and M. S. Sodhi, “Managing risk to avoid supply-chain breakdown,” *MIT Sloan Management Review*, vol. 46, no. 1, p. 53, 2004.
- [111] A. M. Knemeyer, W. Zinn, and C. Eroglu, “Proactive planning for catastrophic events in supply chains,” *Journal of Operations Management*, vol. 27, no. 2, pp. 141–153, 2009.
- [112] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [113] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [114] M. Christopher and H. Peck, “Building the resilient supply chain,” *The International Journal of Logistics Management*, vol. 15, no. 2, pp. 1–14, 2004.
- [115] S. Guillaume, “Designing fuzzy inference systems from data: An interpretability-oriented review,” *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 3, pp. 426–443, 2001.
- [116] O. Vaněk, M. Jakob, O. Hrstka, and M. Pěchouček, “Agent-based model of maritime traffic in piracy-affected waters,” *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 157–176, 2013.
- [117] J. H. Lilly, *Takagi–Sugeno fuzzy systems*. Wiley Online Library, 2010, pp. 88–105.
- [118] C. C. Robusto, “The cosine-haversine formula,” *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.
- [119] E. Zitzler, D. Brockhoff, and L. Thiele, “The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration,” in *Proceedings of the Evolutionary multi-criterion optimization*. Matsushima, JP: Springer, March 2007, pp. 862–876.
- [120] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying ad-hoc networks (fanets): A survey,” pp. 1254–1270, 2013.