

A Proactive Risk-Aware Robotic Sensor Network for Critical Infrastructure Protection

By:

Jamieson McCausland

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of

Master of Applied Science

in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

University of Ottawa

March 2014

© Jamieson McCausland, Ottawa, Canada, 2014

Abstract

In this thesis a Proactive Risk-Aware Robotic Sensor Network (RSN) is proposed for the application of Critical Infrastructure Protection (CIP). Each robotic member of the RSN is granted a perception of risk by means of a Risk Management Framework (RMF). A fuzzy-risk model is used to extract distress-based risk features and potential intrusion-based risk features for CIP. Detected high-risk events invoke a fuzzy-auction Multi-Robot Task Allocation (MRTA) algorithm to create a response group for each detected risk. Through Evolutionary Multi-Objective (EMO) optimization, a Pareto set of optimal robot configurations for a response group will be generated using the Non-Dominating Sorting Genetic Algorithm II (NSGA-II). The optimization objectives are to maximize sensor coverage of essential spatial regions and minimize the amount of energy exerted by the response group. A set of non-dominated solutions are produced from EMO optimization for a decision maker to select a single response. The RSN response group will re-organize based on the specifications of the selected response.

Acknowledgements

I would like to thank my supervisor, Dr. Emil M. Petriu for his continuous support throughout my research term. Without his inspiration, guidance and support the work in this thesis would not be possible. Secondly, I must thank my co-supervisor Rami Abielmona for his uninterrupted support and patience throughout my research. Working with Rami has been an exemplary experience and has left me with invaluable experience as a researcher. Thirdly, I extend much gratitude towards my co-supervisor Ana-Maria Cretu. Her assistance and guidance was well received throughout my research term. I would like to thank everyone at Larus Technologies including my colleagues Rafael Falcon and Alexandros Stathakis for extended discussions and inspiration. There is no doubt that these discussions benefited the research of this thesis.

I would like to thank my entire family for their continuous support and optimism, as well as my friends for their patience during my extended absence. Lastly, I thank my girlfriend Marie-France Charron for being there and not letting me experience this thesis alone; I love you and thank you so much!

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vii
List of Abbreviations	xii
Chapter 1. Introduction	1
1.1. Motivations	4
1.2. Contributions	6
1.3. Thesis Organization	7
Chapter 2. Background and Related Work.....	9
2.1. Risk Modeling and Risk Assessment	9
2.2. Multi-Robot Task Allocation Techniques	14
2.3. Self-Organization of Robotic Sensor Nodes.....	17
2.4. Chapter Summary	20
Chapter 3. Assessing the Risk of a Robotic Sensor Network.....	22
3.1. Data Sources / Environment	23
3.2. Risk Feature Extraction	25
3.2.1. Detection Risk Feature	27
3.2.2. Degree of Distress Risk Feature	33
3.3. Risk Assessment.....	38
3.4. Chapter Summary	40
Chapter 4. Multi-Robot Task Allocation Auction Technique	41
4.1. Optimization Task Generation.....	44
4.2. Task Announcement	44
4.3. Bid Metric Evaluation	46
4.3.1. Input Variable: Battery Level	47

4.3.2.	Input Variable: Distance	47
4.3.3.	Input Variable: Redundant Coverage	48
4.3.4.	Output Variable: Bid Metric	50
4.4.	Bid Submission	53
4.5.	Finalize Auction Session	54
4.6.	Chapter Summary	56
Chapter 5.	Evolutionary Multi-Objective Optimization for Self-Organization of RSN	58
5.1.	Optimization Criteria	59
5.2.	Potential Responses	60
5.3.	Chromosome Design	62
5.4.	Objective Functions	63
5.4.1.	Coverage of Perimeter Objective Function	64
5.4.2.	Coverage of Region Objective Function	66
5.4.3.	Energy Cost (Response) Objective Function	67
5.5.	Crossover and Mutation Operators	68
5.5.1.	Crossover Operator	68
5.5.2.	Mutation Operator	70
5.6.	Chapter Summary	71
Chapter 6.	Experimentation	73
6.1.	Experimental Setup	73
6.2.	Scenario 1: Factory Complex	75
6.2.1.	Scenario Configuration	75
6.2.2.	Risk Assessment	78
6.2.3.	Robot Selection using Fuzzy-Auction MRTA	82
6.2.4.	Self-Organization using the NSGA-II	86
6.3.	Chapter Summary	90
Chapter 7.	Conclusion	92
7.1.	Summary	92
7.2.	Contributions	93

7.3. Future Research	94
References	95
Appendix A. Fuzzy Membership Declarations	98

List of Figures

Figure 1.1: An adopted Risk Management Framework with the contributions of this thesis mapped to the appropriate framework modules.....	6
Figure 2.1: Real-Time Risk Management Framework. Extracted from [14].....	11
Figure 2.2: Markov Chain showing the possible transition between states: Good, Attacking, and Compromised. Extracted from [15]	12
Figure 2.3: Risk Management Framework Architecture. Extracted from [9]	13
Figure 3.1: A block diagram of the Risk Management Framework extended from the work of [9] and [10].....	22
Figure 3.2: Data Snapshot Ingress at Risk Feature Extraction.....	25
Figure 3.3: The risk feature extraction process	26
Figure 3.4: The detection distance fuzzy input variable for detection risk feature ...	28
Figure 3.5: The detected metal fuzzy input variable for the detection risk feature...	29
Figure 3.6: The detection cross-sectional area fuzzy input variable for detection risk feature	30
Figure 3.7: The detection risk fuzzy output variable for the detection risk feature...	31
Figure 3.8: A block diagram of the detection risk feature FIS.....	32
Figure 3.9: Battery level fuzzy input variable for distress risk feature.....	34
Figure 3.10: Battery health fuzzy input variable for distress risk feature	35
Figure 3.11: The communication distance with closest neighbor fuzzy input variable for distress risk feature	36
Figure 3.12: The distress risk fuzzy output variable for the distress risk feature	37

Figure 3.13: The risk assessment algorithm receiving a risk vector and potentially producing a risk event context. This module is a component in a Risk Management Framework [9]	38
Figure 4.1: The Fuzzy-Auction MRTA Process Cycle	42
Figure 4.2: A flow diagram depicting the process of handling a task announcement message.....	46
Figure 4.3: The battery level fuzzy input variable for the bid metric FIS	47
Figure 4.4: The distance fuzzy input variable for the bid metric FIS	48
Figure 4.5: The redundant sensor coverage fuzzy input variable for the bid metric FIS	50
Figure 4.6: The process executed by an auctioneer robot to handle bid submission messages.....	55
Figure 4.7: After a timeout period the auctioneer will take the top bids within the entire collection of received bids.	55
Figure 4.8: The process of handling a close auction message.....	56
Figure 5.1: The response region of a single robot is a circular region (blue circle) and all randomly generated coordinate locations (grey crosshairs)	61
<i>Figure 5.2: Illustration of robots providing sensor coverage of a security perimeter segment (a) and security perimeter contour (b)</i>	<i>64</i>
Figure 5.3: Robot Node Sensor Coverage of <i>lsecurity</i> segment.....	65
Figure 5.4: A robotic node with circular sensor coverage surveying a subset of a region	67
Figure 5.5: Pseudo code for uniform cross over.....	69

Figure 6.1: A single robotic node consisting of a differential drive platform, a simulated LRF, and a simulated GPS module.....	74
Figure 6.2: Scenario 1 “Factory” CIP Scenario.....	75
Figure 6.3: The definition of the security perimeter encapsulating the facility. Positive x is in the right direction on the page and positive z is downwards on the page.	76
Figure 6.4: The risk-aware RSN robotic nodes as black dots and their respective sensor field of views as blue circular regions.	78
Figure 6.5: Robot sensor readings from randomly selected robots in the RSN, showing (a) the battery level of the selected robots; (b) the battery health of the selected robots; and (c) the communication distance for the selected robots.	79
Figure 6.6: Risk assessment plot for select robotic members of the risk-aware RSN	80
Figure 6.7: Log excerpt from robot 2 where the handling of multiple task announcement message at different times is shown	85
Figure 6.8: Two coalitions formed by robot 1 (group at bottom-right) and by robot 27 (group at top-left).....	85
Figure 6.9: Illustration of a single robot's response region. Acceptable positions are represented by crosshair markers.....	87
Figure 6.10: PAS Solutions for robot 1 coalition.....	90
Figure A.1: An example of a trapezoidal membership function declarations with parameters $A=1$, $B=2$, $C=5$, and $D=10$	98
Figure A.2: An example of a triangular membership function declaration with parameters $A=2$, $B=5$, and $C=7$	99

List of Tables

Table 3.1: Possible measurement sensors for a robotic sensor node	23
Table 3.2: The membership functions for the detection distance fuzzy input variable	28
Table 3.3: The membership functions for the detected metal fuzzy input variable ..	29
Table 3.4: The membership functions of the detection cross-sectional area fuzzy input variable.....	30
Table 3.5: The membership functions for the detection risk fuzzy output variable ..	31
Table 3.6: The fuzzy logic rule-set for the detection risk feature FIS.....	32
Table 3.7: Battery level fuzzy input variable membership functions	34
Table 3.8: Battery health fuzzy input variable membership functions.....	35
Table 3.9: The membership functions for the communication distance fuzzy input variable.....	36
Table 3.10: The membership functions for the distress risk fuzzy output variable ..	37
Table 3.11: The fuzzy logic rule-set for the distress risk feature FIS.....	38
Table 4.1: Task Announcement Message	45
Table 4.2: The membership functional for all fuzzy input and output variable of the bid metric FIS.....	50
Table 4.3: The fuzzy logic rule-set for the bid metric FIS	52
Table 4.4: The bid submission message used by a robotic node to submit a bid to an auctioneer	53
Table 6.1: The scenario parameters used for Scenario 1 "Factory"	77
Table 6.2: Risk event details for robot 1 at $t = 8$	81
Table 6.3: Risk event details for robot 1 at $t = 27$	82

Table 6.4: The optimization task generated by robot 1 based on a high-risk event.	83
Table 6.5: The bid submissions for the auction robot 1; ranked in order of bid value	84
Table 6.6: The bid submission details for auctioneer robot 27; ranked in order of bid value.....	84
Table 6.7: The optimization parameters used to obtain the results in this section. .	87
Table 6.8: The PAS for robot 1 listing the solutions by ID. The robot configuration indicates the activated target point, except when -1 which implies the robot is not considered in the response. The coverage and the cost are normalized scalar quantities and cost is a normalized scalar quantity.	88

List of Abbreviations

AAP	Auction Aggregation Protocol
AHP	Analytical Hierarchy Process
CIP	Critical Infrastructure Protection
CNP	Contract Net Protocol
DM	Decision Maker
EMO	Evolutionary Multi-Objective
FIS	Fuzzy Inference System
GPS	Global Positioning System
HMM	Hidden Markov Models
IA	Instantaneous Assignment
LRF	Laser Range Finder
MAC	Media Access Control
mAh	Milliampere-hour
MOEA	Multi-Objective Differential Evolution Algorithm
MR	Multi-Robot
MRTA	Multi-Robot Task Allocation
MSN	Mobile Sensor Network
MT	Multi-Task
NSGA-II	Non-Dominated Sorting Genetic Algorithm
PDND	Parallel and Distributed Network Dynamics
RMF	Risk Management Framework
RSN	Robotic Sensor Network
SAN	Sensor and Actuator Network
SAP	Simple Auction Protocol
SME	Subject Matter Expert
SNR	Signal-To-Noise Ratio
SOSUS	Sound Surveillance System
SR	Single-Robot
ST	Single-Task
TA	Time-Extended Assignment
WSAN	Wireless Sensor and Actor Network
WSN	Wireless Sensor Network

Chapter 1. Introduction

Sensor networks have played a large role throughout history as an innovative means of collecting data in an environment. A lot of time and research has been allocated into the development of more sophisticated sensor networks. If we consider the timeline of sensor networks, one of the first major deployments of a sensor network was the SOund SURveillance System (SOSUS), a sensor network of acoustic sensors deployed along the ocean floor for the purpose of detecting Soviet submarines [1]. As time has passed, robotic sensor networks have emerged. A Robotic Sensor Network (RSN) is capable of dynamically monitoring its environment with more sophisticated sensors as well performing actions with robotic actuators; transducers that traditional sensor networks are not equipped with. Research is consistently focused on utilizing ever increasing powerful embedded devices to extend the functionality and intelligence of sensor networks. Contemporary embedded devices have mitigated much of the memory and processing power concerns of the past. This no longer inhibits the execution of complex algorithms, which can allow developers to embed more intelligence into smaller form factors.

A sensor network is rarely referred to as a “sensor network”, but called by one of many specific sensor network types: Wireless Sensor Networks (WSNs), Mobile Sensor Networks (MSNs), Robotic Sensor Networks (RSNs), Sensor and Actuator Networks (SANs), Wireless Sensor and Actor Networks (WSANs) and more. Each type of sensor network represents a certain level of functionality and sophistication, which must be considered when choosing one for a specific use

case. Typically the most functionality in a sensor network will be found within an RSN. A network of robotic nodes will represent many onboard sensing instruments, actuators, and processing capabilities. In order to manage large sensor data, manage actions performed by actuators, and execute complex algorithms to enable autonomy, powerful embedded software are required to be integrated within these sensor nodes. A robotic sensor node can be decomposed into four primary components: **sensing**, **computation**, **communication**, and **actuation**.

The **sensing** component allows for the perception of data and acquisition of information about the surrounding environment. Through the use of infrared technology, measuring magnetic fields, and more, robots can, for example, sense the presence of possible trespassers on private grounds. The **computation** component refers to the set of on-board algorithms and processing elements, which can be executed on the sensor node itself. Robotic nodes can thus become more intelligent and can extract features out of the raw data streams (e.g., detected vehicle at a specific coordinate) instead of transmitting raw sensor readings to a base station. The **communication** component allows the robotic node to communicate with neighboring nodes and base-stations. Information communicated can be context for a task requiring collaboration and notifying a base-station of any intruder detections and system states (e.g., battery levels, faults, etc). Finally, the **actuation** component allows the robotic node to interact with its environment. This could be simply changing its pose in the environment or manipulating objects with robotic actuators.

Compared to a conventional sensor network, it is expected that an RSN will be more demanding in terms of computational power, memory, and power consumption. Despite these drawbacks, an RSN can be programmed to process the data from its sensors for useful raw data features. Such data features could be used to extract risk features and perceive whether a risky situation is occurring. More interestingly is that an RSN can be programmed with the ability to react in the environment when exposed to a risky situation. An RSN could respond using the actuators available to the robotic sensor nodes, such as differential drive platforms. This would allow the network topology to change. A change in the sensor network or the environment may necessitate a change in topology. This ability in a sensor network has proven to be very useful in many applications, including those related to Critical Infrastructure Protection (CIP) [2].

The Canadian National Strategy for Critical Infrastructure [2] defines critical infrastructures as: energy and utilities, finance, food, transportation, government, information and communication technology, health, water, safety, and manufacturing. These items are essential for the operations of the country motivating the investments to safe-guard them. The role of sensor networks in CIP exists in literature and [3], [4], [5], [6], [7], and [8] are just some examples where researchers were able to produce promising results. Defending critical infrastructures can be accomplished in many different ways such as deploying robotic sensor nodes to monitor the surrounding area perimeter for trespassers and suspicious vehicles.

The real-world imposes countless risks and the world is inherently full of threats producing risk. It is possible for robotic nodes to be able to sense the risk in their environment. An example of this may be a robot that is unable to function properly or lacks the necessary battery power to remain operational. In the domain of CIP, this could mean loss of essential coverage of a region of high-interest and potentially be exploited by trespassers. Robotic nodes are not immune to hardware glitches, manufacturing mistakes, harsh environmental conditions, or continuous wear and tear. A collection of robotic nodes working in collaboration to self-organize, can provide a means to mitigate risky situations.

In the following sections, the motivations for this thesis will be discussed along with the research objectives.

1.1. Motivations

The involvement of robotics in the 21st Century has been revolutionary. Starting out as simple machines carrying out intricate programming instructions and evolving towards autonomous entities capable of interacting with human beings; the term 'robotics' is still evolving.

In the domain of CIP, RSNs can provide an unprecedented advantage through the creation of autonomous security perimeters able to adapt to the environment. The major motivator to a risk-aware RSN is that there is a lack of risk-awareness in current sensing technologies. There is a need to develop risk models for RSNs in CIP, which can ingest raw data features and produce useful risk features. This can be used to provide an additional layer of information to the monitoring and surveillance systems. These risk models can incorporate sensor

node state statistics based on internal sensor (proprioceptor) readings such as: sensor instrument failures, battery levels, time since last maintenance, time since commission, heat sink temperature and much more. External sensor (exteroceptor) readings can also provide valuable information regarding the environment such as: detection statistics, terrain assessment, weather patterns, hostility measures, etc; requiring specific risk models to produce risk features. Each robotic node can produce valuable risk features without any external assistance.

With a sophisticated RSN and perhaps a large network (that is difficult to manage) there is a need for the capability to identify high-risk sources in the network. Only a perfect system in an ideal environment will be risk-free, so it can be safely assumed that in actual deployment all nodes in an RSN will be subject to some degree of risk. The ability to assess risk through risk features can identify high-risk sources. The advantages of knowing risk-sources in a defence system provides substantial motivation to pursue this research endeavour. The deployment and management of a sensor network can become less dependent on a subject-matter expert (SME). In the event of a sensor failure or an increase in frequency of intruder activity, an SME must consider communication ranges, sensor ranges, and expected lifespan of the sensor nodes to deploy a sensor network in the most efficient manner. The topic of self-organizing sensor networks is a research topic growing in popularity.

1.2. Contributions

The objectives of this thesis is to extend the work of [9] and [10] in parallel with [7] and [11] to develop a Risk Management Framework (RMF) for an RSN, in turn, creates a risk-aware RSN capable of assessing and mitigating risk. A depiction of the adopted RMF is shown in Figure 1.1 along with a visual mapping of this thesis' contributions applied to the framework.

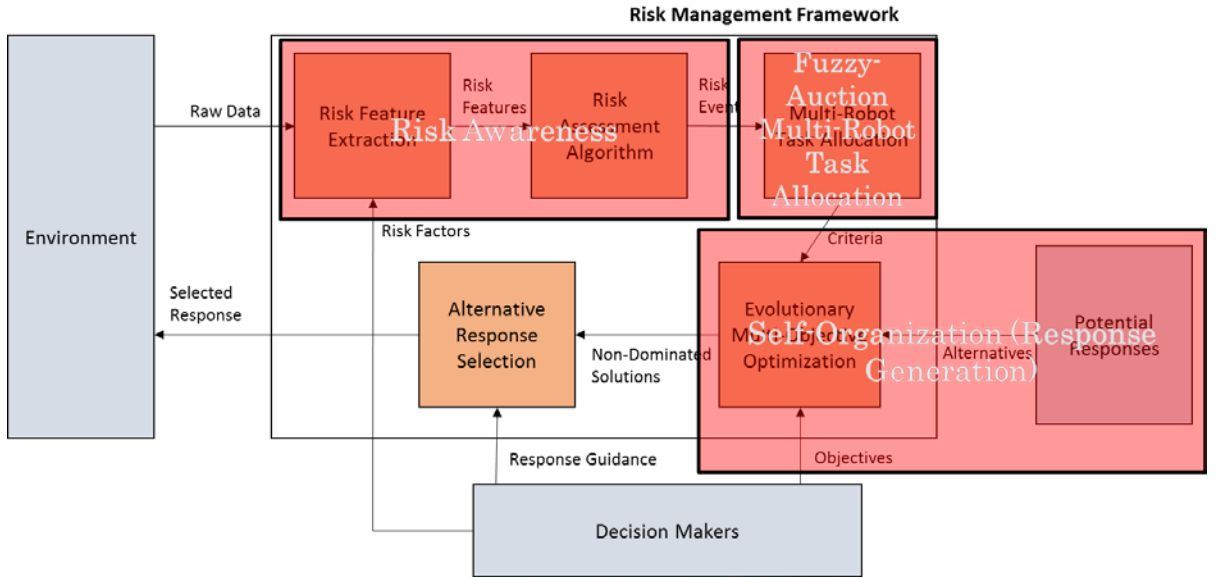


Figure 1.1: An adopted Risk Management Framework with the contributions of this thesis mapped to the appropriate framework modules.

Amongst the wide range of applications for a risk-aware RSN, CIP was chosen as the application. The RMF's modular components will be extended to include: (1) a risk awareness modules for RSNs in the domain of CIP, (2) a Fuzzy-Auction Multi-Robot Task Allocation (MRTA) technique for an RSN in the domain of CIP, and (3) a response selection technique using Evolutionary Multi-Objective (EMO) optimization. The contributions of this work are explicitly defined as:

1. A risk feature extraction module is designed and implemented. Its role is to receive raw data features and extract a set of risk features. A specialized risk model for an RSN for CIP has been implemented.
2. An innovative Fuzzy-Auction MRTA technique had been implemented for selecting nodes to participate in a risk mitigation response. With this scheme, each robotic node will employ an fuzzy-based bid metric calculation;
3. A response generation process using the popular Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [12] using a unique genetic encoding structure to generate optimized robotic node configurations.

The combination of these contributions result in a system that is capable of assessing risk over a large spatial region and respond to concurrent risk events without the requirement of centralized robot management.

1.3. Thesis Organization

This thesis consists of seven chapters. Chapter 2 is a background and related work chapter. The following three chapters cover the three main components of the thesis including: Risk Assessment (Chapter 3), Fuzzy-Auction MRTA (Chapter 4), and Self-Organization (Chapter 5) describing the different modules of the proposed risk management framework for RSNs. The sixth chapter describes the experimentation along with the results. The final chapter concludes the thesis and discusses future research directions related to this topic. Appendix

A provides additional explanations on the definition of fuzzy membership function, useful in the following sections of this thesis.

Chapter 2. Background and Related Work

In this chapter, a literature survey is conducted on the primary topics of interest for this thesis, namely *risk modeling and assessment*, *MRTA*, and *self-organizing sensor networks*. Initially a state-of-the-art review of risk modeling and risk assessment for sensor networks in CIP is presented, followed by a literature survey of various task allocation techniques for RSNs. The objective is to investigate how researchers have been able to assign a group of robots to conduct a single multi-robot task. Lastly, a background survey will be conducted on self-organization techniques.

2.1. Risk Modeling and Risk Assessment

In many problem domains there is a requirement for fault prevention, fault detection, and fault mitigation [13]. It is a real requirement shared by many applications, as we can observe the existence of many fault tolerant protocols, especially in the context of computer networking. In any system lies the possibility of events producing undesirable faults, such as the ones commonly experienced in wireless communication. The latter is an excellent example of a system exposed to a multitude of risk sources. Since wireless communication is an essential component in RSNs, it is discussion-worthy.

A survey of fault management techniques in wireless sensor networks is conducted in [13]. This paper investigates the challenges of resource-constrained sensor nodes, which depend on wireless technology for all communication with neighboring nodes or base stations. An excellent distinction between conventional

wireless devices and wireless sensor nodes highlights four unique challenges faced by WSNs:

1. The need to conserve energy as they are not charged regularly;
2. WSNs are more concerned with reliable event detection than point-to-point reliability;
3. Wireless sensor nodes experience faults at a higher rate; and
4. WSNs using MAC layer protocols must meet the challenges such as sleep intervals for nodes, and can only try mitigating packet collisions instead of completely handling them.

Interesting dynamic behaviours and consequences arise when attempting to meet two very conflicting objectives: maximum operation time on a single battery charge and maintenance of the wireless communications channel. Developing a risk model, which is capable of successfully transforming observable states of a distributed system into a structured unified format is a challenging task. Such a format can be labelled as “risk features” and can allow for a comparison against some criteria to identify the risk. Techniques using Hidden Markov Models (HMMs) approach this challenge [14] and [15], but require a large amount of a priori information.

Tan *et al.* in [14] propose a real-time risk management framework, including a risk assessment module as shown in Figure 2.1.

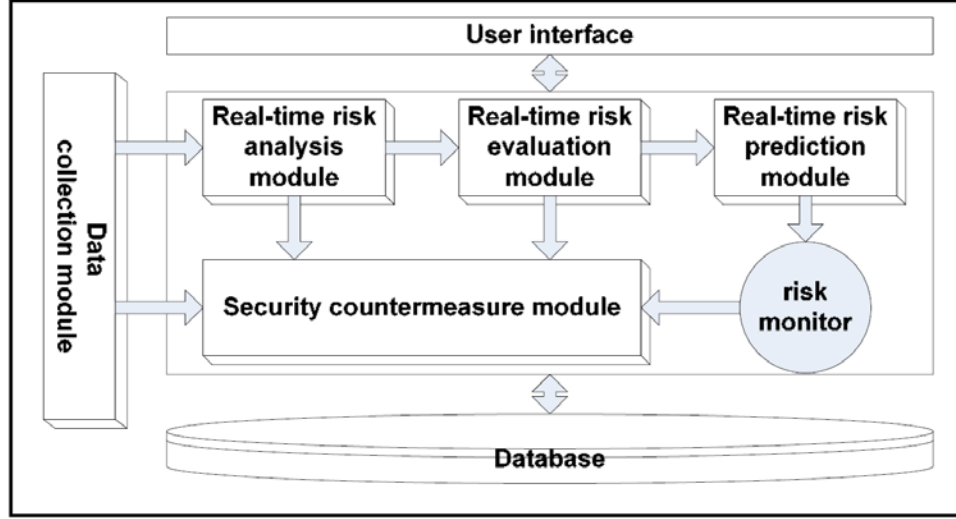


Figure 2.1: Real-Time Risk Management Framework. Extracted from [14]

Their framework consists of a collection of modules: **data collection**, **risk analysis**, **real-time risk evaluation**, **real-time risk prediction**, and **security countermeasure**. In this framework, the authors collect data from the environment (**data collection** module) and can monitor the risk of all assets in the network as well as produce countermeasures to guide system managers to improve the overall security of the system. Modelling the risk is accomplished by proposing three risk elements: *confidentiality*, *integrity*, and *availability* for each asset in the network. Each of these risk elements are then divided into six states to represent the loss degree of the risk element, as defined by the set:

$$S_1 = \{s_0, s_1, s_2, s_3, s_4, s_5\}$$

where s_0 indicates safe status and loss degree of 0%; s_1 to s_4 represents attack status and loss degrees of 20%, 40%, 60%, and 80%, respectively; and finally s_5 represents compromised status with loss degree of 100%. Effectively, each risk element assumes one of three states: *safe*, *attacking*, or *compromised*. Work

conducted in [15] also uses HMMs as part of the risk assessment process, but uses a set of states for each asset, S_2 as defined below:

$$S_2 = \{Good, Attacking, Compromised\}$$

which can be considered a subset of S_1 . Tan *et al* use a fully connected Markov chain, where an asset in the network can transition from any state in S_1 to a different state. Cheng and Ni [15] use the Markov chain as illustrated in Figure 2.2.

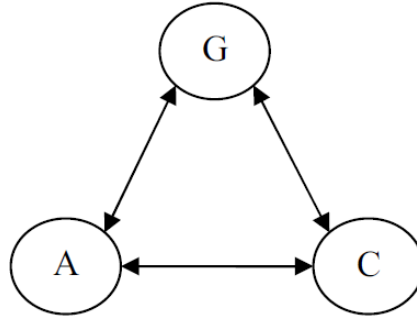


Figure 2.2: Markov Chain showing the possible transition between states: Good, Attacking, and Compromised. Extracted from [15]

As an example, an asset in the network cannot transition from state *good* to *compromised* without transitioning the intermediate state *attacking*.

Authors in [9], [10], [16], and [17] all explore fuzzy-based risk modeling techniques. However, Falcon and Abielmona [9] proposes an evolving human-centric risk management framework applied to WSNs. The framework has an architecture consisting of: **Risk Visualization**, **Risk Assessment**, **Risk Monitoring**, and **Risk Forecasting**, as shown in Figure 2.3. The framework is accessed through a **User Interface module**, is fed data live data through a data source, and produces risk information for a **Security Module**.

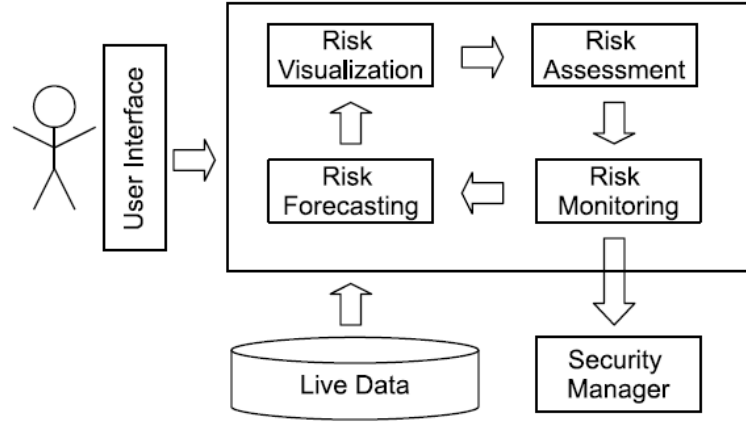


Figure 2.3: Risk Management Framework Architecture. Extracted from [9]

The framework requires the characterization of a set of n risk features defined by set \mathbf{F} in Equation (2.1).

$$\mathbf{F} = \{F_1, F_2, \dots, F_n\} \quad (2.1)$$

Live data fed into the framework is segmented into discrete snapshots. The discrete snapshots are then subjected to a fuzzy risk model, which can provide an overall risk metric for each asset. This is accomplished using Equation (2.2).

$$\tau = \max_{1 \leq i \leq n} \left\{ w_i \cdot \varphi \left(\mu_{X_{Fi}}(x_i) \right) \right\} \quad (2.2)$$

Let w_i is the weight for the i^{th} asset, $\varphi \left(\mu_{X_{Fi}}(x_i) \right)$ is a mapping to a fuzzy membership function $\mu_{X_{Fi}}$. Each wireless sensor node provides two inputs: the battery level and the intrusion distance. The battery input source is fuzzified using a triangular membership function defined by $A = 0$, $B = 0$, and $C = 100$, whereas the intrusion distance input is fuzzified using a trapezoidal membership function defined by $A = 0$, $B = 0$, $C = 2$, and $D = 3$. The same researchers extended the work in [10], applying the framework to the domain of Maritime Search-And-Rescue.

2.2. Multi-Robot Task Allocation Techniques

Some of the most innovative techniques to risk modeling and risk assessment have been reviewed in the previous section. Many of these risk management framework processes do not stop after risk assessment, but proceed to produce solutions to the identified risky situations. In the case of an RSN, a risky sensor node may require replacing, reconfiguration, or maybe assistance from other nodes. This section reviews existing literature on Multi-Robot Task Allocation (MRTA) problems.

Gerkey and Matarić in [18] perform a formal analysis and taxonomy of MRTA. These authors explore the various types of MRTA and the research applied in each group. MRTA problems are typically described by the following criteria:

- Robots
 - Single-Task (ST): robots capable of a single-task at a time;
 - Multi-Task (MT): robots capable of multiple concurrent tasks.
- Tasks
 - Single-Robot (SR): A task requiring a single robot to accomplish;
 - Multi-Robot (MR): A task requiring multiple robots to accomplish.
- Task Assignment
 - Instantaneous Assignment (IA): Available information, the task assignment conducted at a particular time instance;

- Time-Extended Assignment (TA): Information regarding robots and tasks can be received at a later point in time, where they need assignment as they are received.

Given the above criteria, the authors discuss eight types of MRTA problems: ST-SR-IA, ST-SR-TA, ST-MR-IA, ST-MR-TA, MT-SR-IA, MT-SR-TA, MT-MR-IA, and MT-MR-TA. This thesis research will consider the case of ST-MR-IA, which is single-task robots with tasks requiring multiple robots assigned instantaneously upon a high-risk event in the network. The robotic nodes can only participate in a single optimization run at any given time, hence the ST; optimizing the network's topology will typically require the involvement of more than one robot (hence the MR); and robot nodes are assigned tasks shortly after the detection of the high-risk event. According to the analysis of [18], ST-MR-IA MRTA problems typically become a task of *coalition formation*. Given a collection of robots in a family, F , a partitioned set E is acquired through a combinatorial optimization algorithm, to form a task-specific coalition.

A multi-robot cooperation system is introduced in another paper by the same authors [19] called MURDOCH. The MURDOCH multi-robot cooperation system best fits the MT-MR-TA classification of MRTA problems. MURDOCH pursues intentional and *emergent* cooperation, such that each robot does not work explicitly with one another, but cooperate for the purpose of *task allocation*. MURDOCH is developed as a general task allocation system based on a publish/subscribe communication model and is considered a variant of the Contract Net Protocol (CNP) [20] where simple auctions are used to allocate tasks

among the family of robots. Main discussion points of MURDOCH are the publish/subscribe messaging and auction protocol.

The developers of MURDOCH use anonymous messages, which are addressed by content rather than message destination. The sender publishes a message with content and the type of content in the subject of the message. Subject namespaces are also used in MURDOCH, where subjects of messages can pertain to specific namespaces. Messages can be then directed to a particular group of robots listening for certain namespaces only.

The primary components of the MURDOCH task allocation system is based on the underlying distributed negotiation protocol. The tasks are allocated after a first-price and single-round auction session. When a task becomes available, MURDOCH will communicate this with the family of robots using a **task announcement** message. This is conducted by the auctioneer who then will await bids from the robot population. Before a robot member of the population can submit a bid, they must execute a **metric evaluation**, which is a utility metric quantifying their fitness of the announced task. Post metric evaluation, a robot member can execute **bid submission** – a response (bid) message is sent to the auctioneer. The auctioneer will wait for a sufficient amount of time to process each received bid. Once a winner is determined by the auctioneer it will broadcast a **close of auction** message to the robot population. In MURDOCH only a single winner is chosen among the received bids. The selected winner has now acquired a contract to complete the task initially announced. The auctioneer will monitor the progress of the winning individual and renew the contract as appropriate. It is through this

procedure that if the robot carrying out the allocated task is doing so insufficiently, it can be reassigned through an independent auction session.

Authors Mezei *et al.* in [21] pursue efforts to improve upon existing MRTA-auction based techniques such as [22], [19], and [23]. Five new protocols are introduced: *k-SAP*, *SAAP*, *k-SAAP*, *k-AAP*, and *RFT*. Each of these new protocols introduces a unique characteristic, which can improve upon a Simple Auction Protocol (SAP). A simple auction protocol resorts to simple flooding of auction messages throughout the entire network, which can congest the communication network and increase latency in the task allocation process. The work in [21] proposes to restrict robot selection to a subset of the total robot population. This subset would be a collection of robots localized around the task requiring allocation. This is accomplished through permitting only k hops from the auctioneer. For large robotic networks, this can greatly increase the response time of MRTA and reduce the network congestions inflicted by auction sessions. Another improvement made is the use of *auction aggregation protocol (AAP)*. In AAP the bidding process spread outwards to k -hop neighboring nodes until no robots with better fitness can be envisioned. Additional nodes that are assessed for their fitness potential on the task are “invited” to the auction, otherwise the process stops.

2.3. Self-Organization of Robotic Sensor Nodes

The development of self-organization techniques for the various types of sensor networks has been pivotal. The concept of a sensor network self-deploying

and correcting deployments automatically is a mutual goal among many researchers. State-of-the-art techniques in this research domain will be discussed.

The authors of [24] propose a new dynamic model for managing the mobility of a mobile sensor network (MSN). A Parallel and Distributed Network Dynamics (PDND) algorithm is proposed, which can execute on each robot individually. The PDND uses a mathematical model to define the laws of motion for each robot formulated using the steepest descent method in optimization. With a MSN in a non-optimal configuration, the system is said to have potential energy. The PDND algorithm executes to reduce this potential energy, allowing the network to “settle” into an optimized state. The authors propose that an MSN could be attracted to a region of interest or even a moving target; through the execution of the PDND algorithm, the robots will establish a new topology based on the defined network dynamics. Opposite (i.e., repulsive) forces can also exist, repelling a member of the MSN away from a region or specific object. These attraction and repellent forces are the result of specific potential functions. A potential function essentially measures the amount of resistance of a mobile unit move change positions.

Jin *et al.* in [25] use a Multi-Objective Differential Evolution Algorithm (MOEA) to satisfy a set of objective functions. These objective functions are defined as:

1. Area coverage rate;
2. Network redundant coverage area; and
3. Energy consumption of sensing

The average coverage rate, $R_{area}(C)$ is described by Equation (2.3), which provides a ratio of the grid points covered by a sensor node C and the total area of grid points representing a particular region of interest, $m \times n$.

$$R_{area}(C) = \frac{\sum_{x=1}^m \sum_{y=1}^n P_{cov}(x, y, C)}{mn} \quad (2.3)$$

The sensor network redundant coverage area, $S_{redundant}$, is defined in Equation (2.4). This function evaluates how many times a specific grid points at (x, y) is being surveyed by other nodes, measuring the amount of redundant coverage.

$$S_{redundant} = \sum_{x=1}^m \sum_{y=1}^n \left(\sum_{i=1}^N P_{overlap}(x, y, c_i) - 1 \right) \quad (2.4)$$

where $P_{overlap}$ is defined in Equation (2.5). This equation returns the value one if robot node c_i and any other node both cover the location at x and y , otherwise a zero is returned.

$$P_{overlap}(x, y, c_i) = \begin{cases} 1 & \text{if } (x, y) \in c_i, \text{ and } (x, y) \in c_j, i, j \in [1, N], i \neq j, \\ 0 & \text{other} \end{cases} \quad (2.5)$$

The third objective function these authors consider is the amount of energy being consumed to operate the sensors. The energy demands of the sensor are proportional to the desired range of the sensor. Equation (2.6) captures the energy demands of the sensor network.

$$E_{sensor} = u \sum_{i=1}^N r_i^2 \quad (2.6)$$

where u represents the sensor parameters and r_i is the sensor radius of node i . The authors use an overall fitness function based on a weighted-sum approach as described in Equation (2.7).

$$f_{obj}(x) = w_1 n_1 f_1(x) + w_2 n_2 f_2(x) \quad (2.7)$$

Let w_1 and w_2 be the weight coefficients and n_1 and n_2 are used as normalizing coefficients for the functions $f_1(x)$ and $f_2(x)$, respectively. The work in [25] employs a genetic encoding of robot coordinate values (i.e., real numbers), a coordinate for each sensor node in the network. Sensor nodes can be disabled by assigning the coordinate (0, 0) for a particular asset in the chromosome. Thus, the optimization algorithm will search for a solution using a subset of sensor nodes, if possible. However, the optimization algorithm always considers all sensor nodes which can become very cumbersome for very large sensor networks.

2.4. Chapter Summary

This chapter provided an overview of the three main topics of this thesis research: *risk modelling*, *multi-robot task allocation*, and *self-organizing sensor networks*. The first section of the report discussed previous research in the topic of risk modelling and risk assessment, while trying to keep as close to the domain of sensor networks as possible. Much risk assessment work is done in the field of computer network security and firewalls which is not directly applicable in this thesis, however, only a few researchers have begun applying a risk management framework to the domain of robotic sensor networks.

The second section discussed previous research in the topic of multi-robot task allocation. This topic represents some of the major contributions of this thesis.

Existing protocols such as MURDOCH [19] have been discussed, which employ a market-based approach to task allocation. This has inspired the further exploration of market-based task allocation through the use of simple auctions.

The third section discussed existing techniques in self-organizing sensor networks. Decentralized techniques are typically more advantageous in self-organizing sensor networks problems. Much of the research reviewed employs a distributed algorithm to govern the movement behaviour of the sensor network's nodes. Only few researchers have utilized evolutionary multi-objective optimization techniques for the purposes of self-organization.

Chapter 3. Assessing the Risk of a Robotic Sensor Network

This chapter will introduce the techniques used by the risk-aware RSN to model and assess risk from raw data. A RMF developed in [9] and [10], will be a foundation for this work. The framework is illustrated in Figure 3.1, which clearly shows each component and the relationships between components.

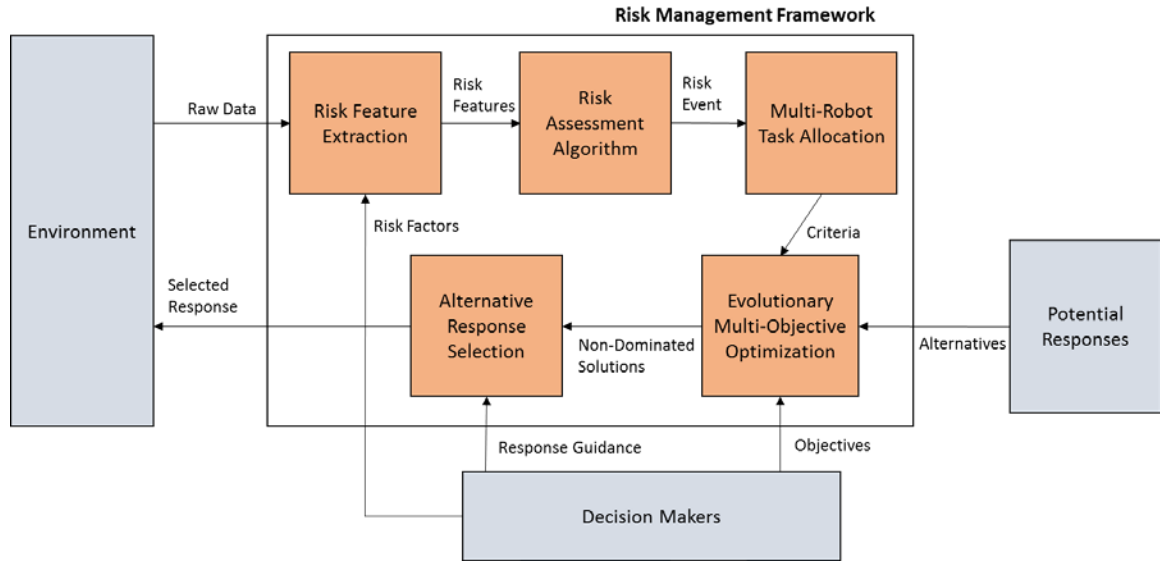


Figure 3.1: A block diagram of the Risk Management Framework extended from the work of [9] and [10]

This chapter discusses the various components of the RMF. In the first section, the data sources (i.e., environment) used and how this data is prepared for processing is presented. In the next section a set of risk features are defined and the process of extracting these features out of raw data is discussed. Finally, we present how a risk assessment module ingests the risk features and assigns a risk metric to a robotic node.

3.1. Data Sources / Environment

Let us consider a single robotic node, S_{node}^i , which is the i^{th} robotic node in the RSN. Thus, an RSN is a family of robotic nodes, $F = \{S_{node}^1, S_{node}^2, \dots, S_{node}^{N_{nodes}}\}$. A robotic node is equipped with sensor instruments in order to gather information regarding the environment. These sensors can be classified as *proprioceptors* and *exteroceptors* measuring sensors. Internal sensors retrieve data regarding the internal components of the sensor node itself which can be interpreted as measuring vitals of the sensor node unit. Table 3.1 highlights a few examples of these types of sensors. External sensors retrieve information external to the sensor node. In the context of CIP, this involves the detection of objects in close proximity to an established defence perimeter.

<i>Internal Measurement Sensors</i>	<i>External Measurement Sensors</i>
<ul style="list-style-type: none"> • Voltmeter • Ammeter • Thermal Sensors • Humidity Sensors • 3-D Accelerometers • Global Position System (GPS) Module • Wheel Encoders • Electronic Compass 	<ul style="list-style-type: none"> • Laser Range Finder (LRF) • Sonar • Infrared Proximity • Infrared Motion Detection • Magnetic Field • Vision (optic/infrared) • Microphones • Gas Detector • Metallic Presence Detector • Electric Field Measurement

Table 3.1: Possible measurement sensors for a robotic sensor node

Within the category of internal sensors, let us consider the measurements of the robot node's battery management system and sensor node's wireless communication device system. A robotic node is in constant need of energy to function and requires the ability to communicate with surrounding robotic nodes to establish a communication network. Let $x_{batt}(t)$ (units mAh) represent the remaining energy in the battery. Let $x_{bCap}(t)$ (units of mAh) represent the total battery capacity. Finally, let $x_{SNR}(t)$ (unit-less) be the signal to noise ratio measured at the robotic node. Let t represent the time since the robot's initial condition at t_o . Let t_o represent initial time.

Within the category of external sensors, let us consider sensors to aid in the detection of threatening objects in proximity to the robotic nodes. Let $x_{dist}(t)$ (units meters) represent the distance of the closest detected object to the robot node. Let x_{area} (units meters-squared) represent the cross sectional area of the object at distance $x_{dist}(t)$. Lastly, let $x_{metal}(t)$ represent the presence of metal in proximity of the robot node.

Data for a single robotic node is packed into a data snapshot to be processed by the risk feature extraction module as was done in [9] and [10] and as depicted in Figure 3.2.

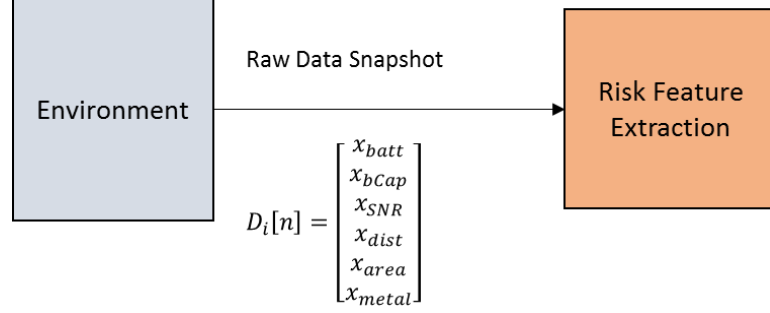


Figure 3.2: Data Snapshot Ingress at Risk Feature Extraction

The data snapshot $D_i[n]$, uses a discrete time index n . Raw data from sensor percepts are sampled at a constant period, τ_{sample} . Therefore, the system can start at time t_o , which makes $n = 0$. With a data snapshot prepared, the data can now be processed by risk feature extraction.

3.2. Risk Feature Extraction

A data snapshot, $D_i[n]$ has been produced by the robot node which is then passed to the risk feature extraction module (see Figure 3.2). It is not guaranteed that the contents of the data snapshot reveal the risk in the environment and within the robot node itself. Through the use of an appropriate risk model, risk features can be extracted out of the data snapshots. The process of risk feature extraction represents the transformation into the risk feature space. This is a prerequisite for the risk assessment algorithm [9] and [10].

The risk feature extraction process (depicted in Figure 3.3) begins by mapping each element of the ingested data snapshot at discrete time n , $D_i[n]$, to inputs of some transformation system. This thesis uses a fuzzy-based transformation system, specifically a Mamdani Fuzzy Inference System (FIS), Ψ . Each data element in $D_i[n]$ will be mapped to a fuzzy input variable, $\varphi(x[n])$. A

fuzzy input variable is represented by a set of membership functions as shown in Equation (3.1). It is through these membership functions that the raw data from the sensor node can be fuzzified.

$$\varphi(x[n]) = \{\mu_1(x[n]), \dots, \mu_N(x[n])\} \quad (3.1)$$

where N is the number of membership functions used to represent that fuzzy input variable.

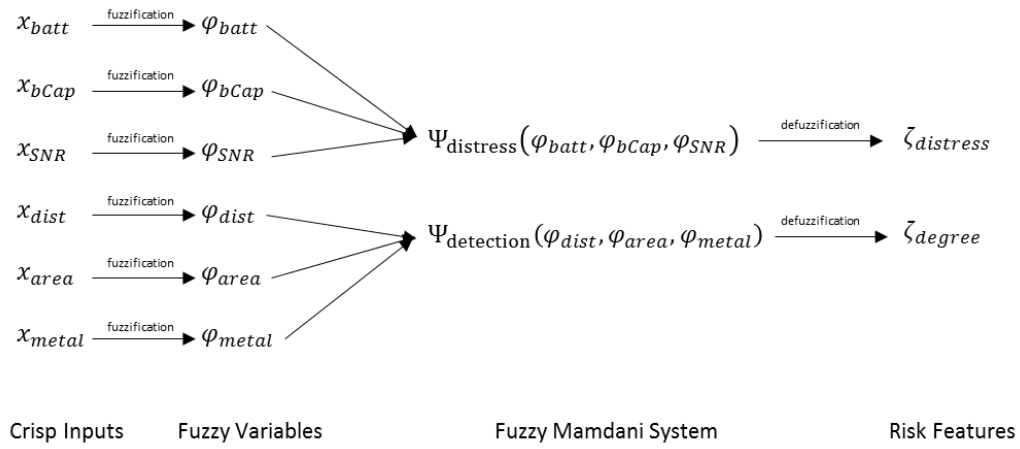


Figure 3.3: The risk feature extraction process

The input fuzzy variables are used in the FIS as well as a set of fuzzy rules and a fuzzy output variable, $\varphi_{risk} = \Psi(\cdot)$. The defuzzification of this fuzzy output variable provides the crisp risk metric.

The two risk features chosen for implementation in this thesis are the **distress**, $\zeta_{distress}$, risk feature and the **detection**, $\zeta_{detection}$, risk feature. Although many more risk features can be considered, in the context of CIP, the distress of the network and the detection of intruders are most important to capture.

3.2.1. Detection Risk Feature

The detection risk feature, $\zeta_{detection}$ is a metric extracted out of raw data to indicate the presence of risk as a consequence of detecting the presence of an object within the sensor coverage of a robotic sensor node. The raw data used is x_{dist} , x_{area} , and x_{metal} , which represent the distance to the detected object, cross-sectional area of the detected object, and the presence of metal in the detected object. This information is retrieved from a Laser Range Finder (LRF) and a metal detection sensor.

The motivation behind the aforementioned data inputs is to generate a risk metric for object detection. This risk metric will be generated considering the proximity distance of the object, as well as size and composition to help classify the detection. Additional knowledge regarding the detected object could greatly impact the risk inferred by the detection. For instance, a small and non-metal object detected by a sensor node, may need the object to get very close before raising a high-risk event. However, a large metal object may cause the robotic sensor node to fire a high-risk event at a distance greater than the small and non-metal object.

A Mamdani FIS, $\Psi_{detection}$, is used to calculate the detection risk metric from the aforementioned data inputs. This process begins by defining fuzzy input variable $\varphi(x)$, which will fuzzify crisp value x . The first fuzzy input variable is $\varphi_{dist}(x_{dist})$, which uses crisp input x_{dist} . If we let D_x be the domain of some variable x , then $D_{x_{dist}} = \{x_{dist} \in \mathbb{R} | x \geq 0\}$, where $D_{\varphi_{dist}} = D_{x_{dist}}$.

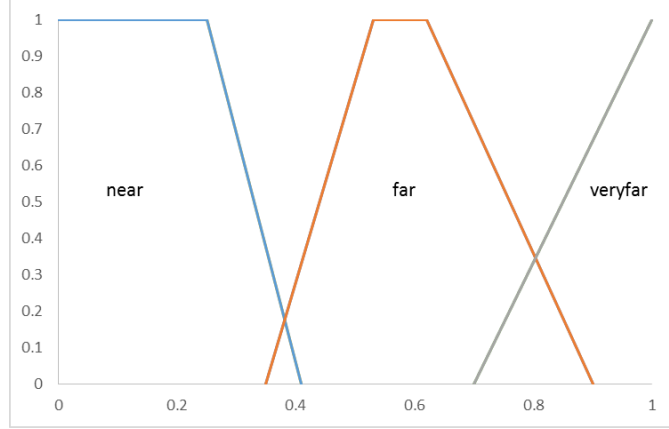


Figure 3.4: The detection distance fuzzy input variable for detection risk feature

The fuzzy input variable, φ_{dist} using a set of membership functions: $\mu_d^{near}(x_{distance})$, $\mu_d^{far}(x_{distance})$, and $\mu_d^{veryfar}(x_{distance})$; depicted in Figure 3.4 and described in Table 3.2. The membership functions are defined using A, B, and C parameters from triangular membership functions. Trapezoidal membership functions use an additional D parameter (see Appendix A). The values described within Table 3.2 have been acquired experimentally.

Membership Function, $\mu_{distance}$	Definition	
	Function Type	Parameters
$\mu_d^{near}(x_d)$	Trapezoidal	A=0.0, B=0.0, C=0.25, D=0.41
$\mu_d^{far}(x_d)$	Trapezoidal	A=0.35, B=0.53, C=0.62, D=0.9
$\mu_d^{veryfar}(x_d)$	Triangular	A=0.7, B=1.0, C=1.0

Table 3.2: The membership functions for the detection distance fuzzy input variable

The second fuzzy variable is $\varphi_{metal}(x_{metal})$, which uses crisp input x_{metal} . Let $D_{x_{metal}} = \{x_{metal} \in \mathbb{R} \mid 0 \leq x_{metal} \leq 1\}$. The crisp value is read and normalized from the metal detection sensor equipped on the robotic node. The fuzzy input variable, φ_{metal} uses two membership functions: $\mu_{metal}^{strong}(x_{metal})$ and

$\mu_{metal}^{weak}(x_{metal})$. These membership functions are depicted in Figure 3.5 and described in Table 3.3. The membership functions of Table 3.3 were acquired experimentally.

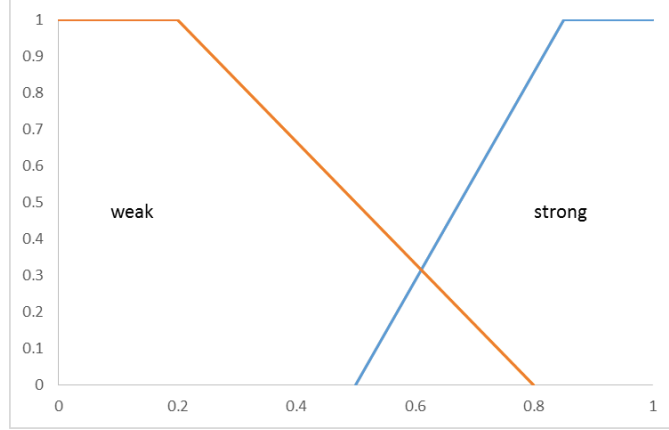


Figure 3.5: The detected metal fuzzy input variable for the detection risk feature

Membership Function, μ_{metal}	Definition	
	Function Type	Parameters
$\mu_{metal}^{strong}(x_{metal})$	Trapezoidal	A=0.0, B=0.58, C=1.0, D=1.0
$\mu_{metal}^{weak}(x_{metal})$	Trapezoidal	A=0.35, B=0.53, C=0.62, D=0.9

Table 3.3: The membership functions for the detected metal fuzzy input variable

The third fuzzy variable is $\varphi_{area}(x_{area})$, which represents the cross-sectional area of the detected object, x_{area} (in units meters-squared, m^2). Let the domain be $D_{x_{area}} = \{x_{area} \in \mathbb{R} \mid x_{area} \geq 0\}$. This fuzzy input variable consists of three membership function: $\mu_{area}^{small}(x_{area})$, $\mu_{area}^{medium}(x_{area})$, and $\mu_{area}^{large}(x_{area})$. These membership functions are depicted in Figure 3.6 and Table 3.4. These membership function have been defined experimentally.

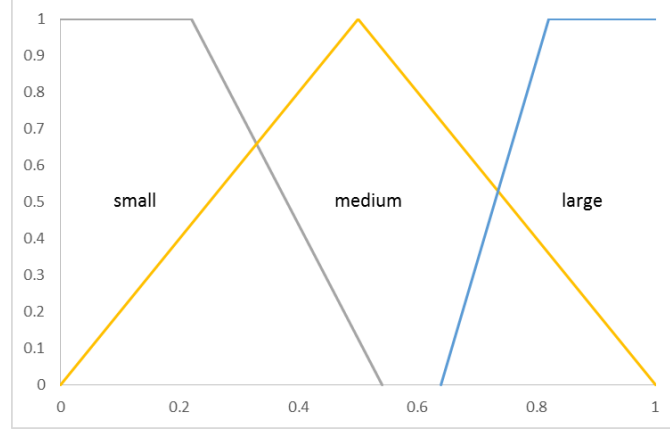


Figure 3.6: The detection cross-sectional area fuzzy input variable for detection risk feature

Membership μ_{area}	Function,	Definition	
		Function Type	Parameters
$\mu_a^{small}(x_a)$		Trapezoidal	A=0.0, B=0.0, C=0.22, D=0.54
$\mu_a^{medium}(x_a)$		Triangular	A=0.0, B=0.5, C=1.0
$\mu_a^{large}(x_a)$		Trapezoidal	A=0.64, B=0.82, C=1.0, D=1.0

Table 3.4: The membership functions of the detection cross-sectional area fuzzy input variable

The detection risk metric, $\zeta_{detectionRisk}$ is the result of the defuzzification process of the output variable. In the case of a Mamdani FIS, this output variable is a fuzzy variable $\varphi_{detectionRisk}$. Let $\zeta_{detectionRisk}$ have the domain, $D_{\zeta_{detectionRisk}} = \{\zeta_{detectionRisk} \in \mathbb{R} \mid 0 \leq \zeta_{detectionRisk} \leq 1\}$, with no units. A maximum value of 1 indicates the complete presence of detection risk, whereas the minimum value of 0, indicates the complete absence of detection risk. The membership functions of the output fuzzy variable are: $\mu_{detection}^{low}(\cdot)$, $\mu_{detection}^{lowmedium}(\cdot)$, $\mu_{detection}^{medium}(\cdot)$, $\mu_{detection}^{high}(\cdot)$, and $\mu_{detection}^{veryhigh}(\cdot)$. These functions are depicted in Figure 3.7 and defined in Table 3.5. The definition of the membership function have been acquired experimentally.

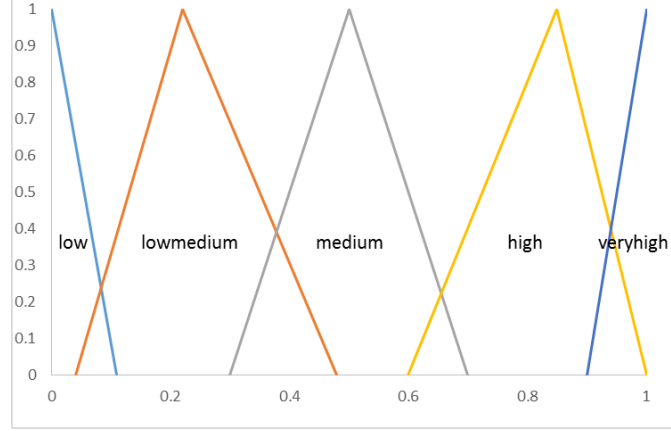


Figure 3.7: The detection risk fuzzy output variable for the detection risk feature

Membership Function, μ	Definition	
	Function Type	Parameters
$\mu_{risk}^{low}(\cdot)$	Triangular	A=0.0, B=0.0, C=0.11
$\mu_{risk}^{lowmedium}(\cdot)$	Triangular	A=0.04, B=0.22, C=0.48
$\mu_{risk}^{medium}(\cdot)$	Triangular	A=0.3, B=0.5, C=0.7
$\mu_{risk}^{high}(\cdot)$	Triangular	A=0.6, B=0.85, C=1.0
$\mu_{risk}^{veryhigh}(\cdot)$	Triangular	A=0.9, B=1.0, C=1.0

Table 3.5: The membership functions for the detection risk fuzzy output variable

The FIS, $\Psi_{detection}$ (depicted in Figure 3.8) is transforming a three-dimensional input space to a one-dimensional quantity. This quantity is the introduced detection risk metric, $\zeta_{detectionRisk}$. A set of fuzzy rules are defined within $\Psi_{detection}$ to select the output fuzzy variable membership functions. These rules are defined within Table 3.6.

Fuzzy Rule	Weight
if (Distance is Near) then (Risk is High)	0.2

if (Distance is Far) then (Risk is LowMedium)	0.2
if (Distance is Near) and (Area is Large) then (Risk is VeryHigh)	0.2
if (Distance is Far) and (Area is Large) then (Risk is High)	0.2
if (Distance is Near) and (Metal is Strong) and (Area is not small) then (Risk is VeryHigh)	0.2
if (Distance is Far) and (Metal is Strong) and (Area is not Small) then (Risk is Medium)	0.2
if (Distance is VeryFar) then (Risk is Low)	0.7

Table 3.6: The fuzzy logic rule-set for the detection risk feature FIS

The fuzzy system $\Psi_{detection}$ uses the **minimum** AND method, the **maximum** OR method, the **minimum** IMPLICATION method, and **maximum** AGGREGATION method. The **centroid** defuzzification method is used for the detection risk feature.

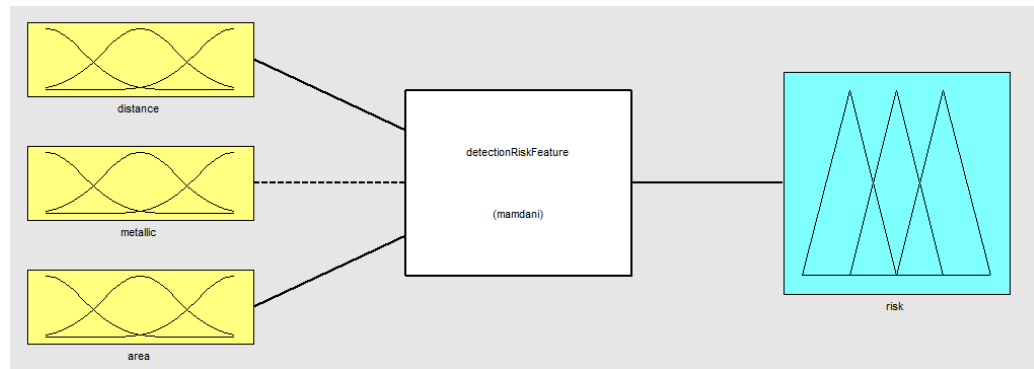


Figure 3.8: A block diagram of the detection risk feature FIS

3.2.2. Degree of Distress Risk Feature

This risk feature utilizes the proprioceptors of the robotic node to assess the degree of distress of the robotic node. The degree of distress risk feature, $\zeta_{distress}$ is the amount of risk generated within the robotic node itself. This risk feature begins by mapping the available data sources into normalized metrics. Let us consider the following data sources: **battery level**, $x_{batt_{level}}$, **battery health**, $x_{batt_{health}}$, and **communication distance**, x_{comm} . These three data sources represent some of the major contributors of risk for a wireless robotic node. Since the robotic nodes are wireless, they depend on an independent power supply. With a depleted power supply, the robotic node will not be able to remain a functioning member of the RSN. Therefore, the remaining electrical energy is inversely proportional to the distress risk contribution for that node. Battery health is another great data source for risk assessment of a robotic node. Over time and depending on how the battery is used, the energy capacity of the battery will decrease. Thus, a fully-charged battery in poor health is a similar threat to a healthy battery with a poor charge. Both can result in power failure for the robotic node. The third contributor to distress risk is the communication distance of the node. This is the distance to either the closest base station or neighboring robotic node. An increase in x_{comm} is proportional to an increase in average power usage and proportional to the contributed amount of risk.

Let us define a Mamdani FIS, $\Psi_{distress}$ to extract the degree of distress risk feature from the selected raw data sources. This FIS uses three fuzzy input variables: $\varphi_{batt}(x_{batt})$, $\varphi_{battHealth}(x_{battHealth})$, and $\varphi_{commDist}(x_{commDist})$; the FIS

uses one single fuzzy output variable $\varphi_{distressRisk}(\cdot)$. The first fuzzy input variable is φ_{batt} , which uses crisp input x_{batt} . This is the normalized battery level on the robotic node. Let the domain be, $D_{x_{batt}} = \{x_{batt} \in \mathbb{R} \mid 0 \leq x_{batt} \leq 1\}$.

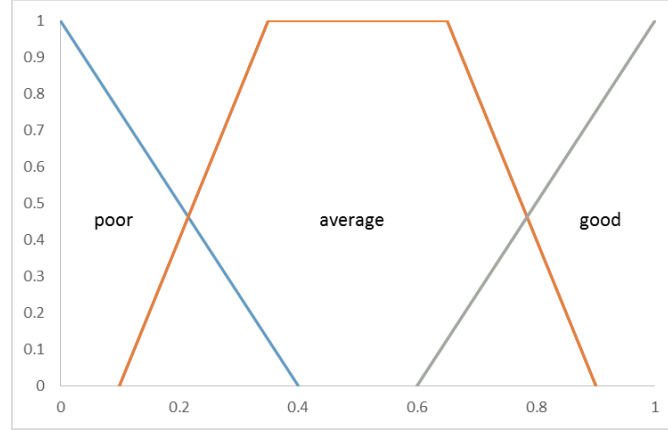


Figure 3.9: Battery level fuzzy input variable for distress risk feature

The membership functions (as shown in Figure 3.9) are: $\mu_{battery}^{poor}$, $\mu_{battery}^{average}$, and $\mu_{battery}^{good}$. These functions are defined in Table 3.7 and have been acquired experimentally.

Membership Function, μ	Definition	
	Function Type	Parameters
$\mu_{batt}^{poor}(x_{batt})$	Triangular	A=0.0, B=0.0, C=0.4
$\mu_{batt}^{average}(x_{batt})$	Trapezoidal	A=0.1, B=0.35, C=0.65, D=0.9
$\mu_{batt}^{good}(x_{batt})$	Triangular	A=0.6, B=1.0, C=1.0

Table 3.7: Battery level fuzzy input variable membership functions

The second fuzzy variable is $\varphi_{battHealth}$, which uses crisp input value $x_{battHealth}$. This crisp value is a normalized quantity of the battery's ability to hold a charge. Let the domain be $D_{x_{battHealth}} = \{x_{battHealth} \in \mathbb{R} \mid 0 \leq x_{battHealth} \leq 1\}$,

where a quantity of 1 represents a new battery capable to charge to its rated capacity. A value of 0 represents a battery that can no longer hold a charge. The fuzzification of this crisp input uses three membership functions: $\mu_{battHealth}^{replace}$, $\mu_{battHealth}^{average}$, and $\mu_{battHealth}^{good}$. These membership functions are depicted in Figure 3.10 and described in Table 3.8. The membership functions for the battery health fuzzy input variable were defined experimentally.

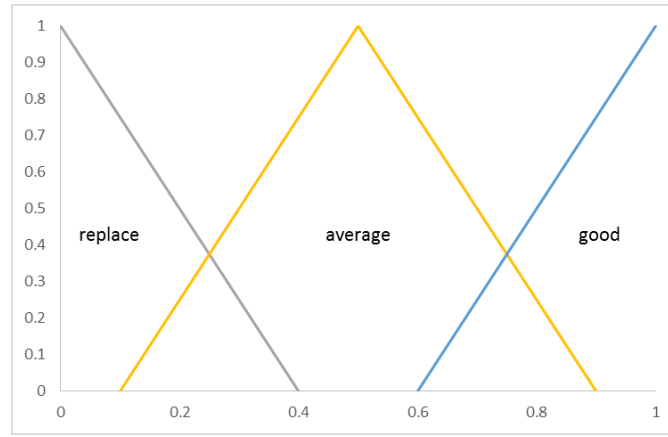


Figure 3.10: Battery health fuzzy input variable for distress risk feature

Membership Function, μ	Definition	
	Function Type	Parameters
$\mu_{battHealth}^{replace}(x_{bCap})$	Triangular	A=0.0, B=0.0, C=0.4
$\mu_{battHealth}^{average}(x_{bCap})$	Triangular	A=0.1, B=0.5, C=0.9
$\mu_{battHealth}^{good}(x_{bCap})$	Triangular	A=0.6, B=1.0, C=1.4

Table 3.8: Battery health fuzzy input variable membership functions

The third and final fuzzy input variable is $\varphi_{commDist}$, which uses crisp input $x_{commDist}$. The input $x_{commDist}$ is the communication distance to the closet neighboring robotic node. This value is a significant source of distress for the

robotic node as the more distance separates a robotic node with its neighbours the more difficult it is for the node to communicate with others. This can lead to dropped connections and excess battery usage, compromising the longevity of the sensor network. The domain of the crisp input value is $D_{x_{commDist}} = \{x_{commDist} \in \mathbb{R} \mid x_{commDist} > 0\}$. The fuzzy input variable, $\varphi_{commDist}$ uses three membership functions: $\mu_{commDist}^{close}$, $\mu_{commDist}^{inrange}$, and $\mu_{commDist}^{outofrange}$. These membership functions are depicted in Figure 3.11 and described in Table 3.9. The communication distance membership functions were defined experimentally.

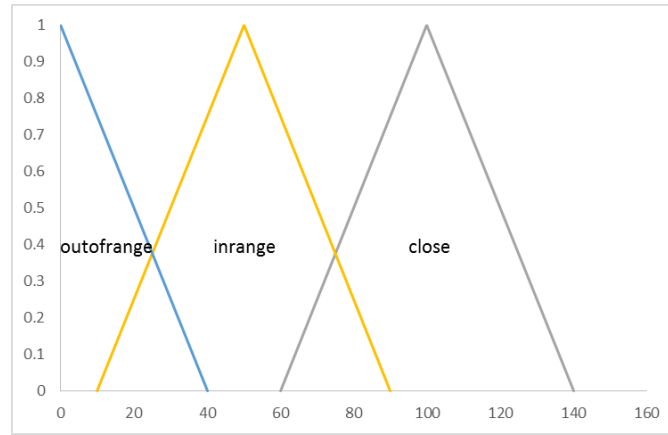


Figure 3.11: The communication distance with closest neighbor fuzzy input variable for distress risk feature

Membership Function, μ	Definition	
	Function Type	Parameters
$\mu_{commDist}^{outofrange}(x_{comm})$	Triangular	A=60.0, B=100.0, C=140.0
$\mu_{commDist}^{inrange}(x_{comm})$	Triangular	A=10.0, B=50.0, C=90.0
$\mu_{commDist}^{close}(x_{comm})$	Triangular	A=0.0, B=0.0, C=40.0

Table 3.9: The membership functions for the communication distance fuzzy input variable

The FIS, $\Psi_{distressRisk}$ uses a single output variable, which provides the distress risk metric of the robotic node upon defuzzification. As $\Psi_{distressRisk}$ is a

Mamdani FIS, the output variable, $\varphi_{distressRisk}$ is a fuzzy variable with a domain $D_{\zeta_{distressRisk}} = \{\zeta_{distressRisk} \in \mathbb{R} \mid 0 \leq \zeta_{distressRisk} \leq 1\}$, with no units. A maximum value of 1 indicates the complete presence of distress risk, whereas a minimum value of 0 represents the complete absence of distress risk. The variable, $\varphi_{distressRisk}$ utilizes five triangular membership functions where are defined in Table 3.10 and depicted in Figure 3.12. The output distress risk fuzzy variable membership functions were defined experimentally.

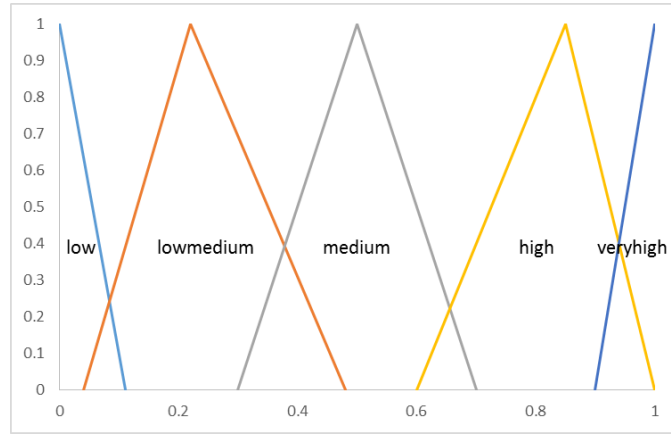


Figure 3.12: The distress risk fuzzy output variable for the distress risk feature

Membership Function, μ	Definition	
	Function Type	Parameters
$\mu_{distress}^{low}(\cdot)$	Triangular	A=0.0, B=0.0, C=0.11
$\mu_{distress}^{lowmedium}(\cdot)$	Triangular	A=0.04, B=0.22, C=0.48
$\mu_{distress}^{medium}(\cdot)$	Triangular	A=0.3, B=0.5, C=0.7
$\mu_{distress}^{high}(\cdot)$	Triangular	A=0.6, B=0.85, C=1.0
$\mu_{distress}^{veryhigh}(\cdot)$	Triangular	A=0.9, B=1.0, C=1.0

Table 3.10: The membership functions for the distress risk fuzzy output variable

A set of fuzzy rules are defined within $\Psi_{distressRisk}$ to select the output fuzzy variable membership functions. These rules are presented in Table 3.11.

Fuzzy Rule	Weight
if (BatteryLevel is Poor) then (Risk is VeryHigh)	1.0
if (BatteryHealth is Replace) then (Risk is VeryHigh)	1.0
if (CommDist is OutOfRange) then (Risk is VeryHigh)	1.0
if (BatteryLevel is Good) then (Risk is Low)	1.0
if (BatteryHealth is Good) then (Risk is Low)	1.0
If (CommDist is Close) then (Risk is Low)	1.0

Table 3.11: The fuzzy logic rule-set for the distress risk feature FIS

This FIS uses a **minimum** AND method, **maximum** OR method, **minimum** IMPLICATION method, and **maximum** AGGREGATION method. The **centroid** DEFUZZIFICATION method used for the distress risk feature.

3.3. Risk Assessment

The risk assessment module [9] is depicted in Figure 3.13. It evaluates the risk input features and outputs risk events. This module is capable of generating the necessary context to begin a network response for a high-risk detection.

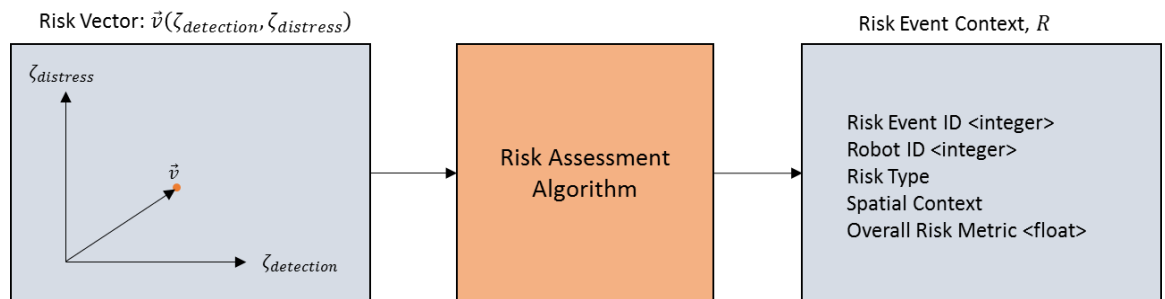


Figure 3.13: The risk assessment algorithm receiving a risk vector and potentially producing a risk event context. This module is a component in a Risk Management Framework [9]

Each risk feature produces a single output value on the interval of $[0,1]$. Each data snapshot will contain a risk vector $\vec{v} = [\zeta_{detection} \ \ \zeta_{distress}]$, which is ingested by the risk assessment algorithm. Given a risk vector, \vec{v} we want to generate a risk event, \mathbf{R} . Each risk feature added should represent an additional dimension in the risk feature space. In other words, a new risk feature should not add redundant risk information. A risk event, \mathbf{R} , provides: *robot id*, *risk type*, *spatial information*, and *severity*, as defined below:

1. Robot Id, ϕ_{id} : The robotic node that detected the risk;
2. Risk Type, $\phi_{riskType}$: Obtained by interpreting \vec{v} ; $\{NODE_DISTRESS, POTENTIAL_INTRUSION\}$;
3. Spatial Information, $\phi_{spatial}$: Spatial information pertaining to the risk; and
4. Overall Risk Metric, $\phi_{overallRisk}$: The overall risk metric at the time of high-risk event generation.

An overall risk metric, ϕ_{risk} for a robot node is determined using the maximum component of v , as shown in Equation (2.2).

$$\phi_{overallRisk} = \max(\zeta_{detection}, \zeta_{distress}) \quad (3.2)$$

This is a simple and effective means of getting the overall risk of a robot node. The risk type is selected based on which risk vector component is largest. Let $\phi_{riskType} \in \{NODE_DISTRESS, POTENTIAL_INTRUSION\}$ and is defined as shown in Equation (2.3).

$$\phi_{riskType} = \begin{cases} NODE_DISTRESS & \zeta_{distress} > \zeta_{detection} \\ POTENTIAL_INTRUSION & else \end{cases} \quad (3.3)$$

3.4. Chapter Summary

This chapter provided an overview of the risk modeling and risk assessment process for the risk-aware RSN. It is very important to discuss the environment of the robotic nodes in a CIP scenario as well as the available sensor instruments available. By observing the external and the internal environments of the robotic nodes, useful sensor percepts are generated. The methods by which the sensor percepts are transformed into risk features were presented. This was necessary to explain how an overall risk metric can be assigned to a single robot node, possibly invoking a high-risk event trigger. It is this trigger that instigates the RSN's unique ability to generate newly optimized node topologies using EMO optimization.

Chapter 4. Multi-Robot Task Allocation Auction Technique

This chapter discusses how to capitalize on the risk detections produced by the techniques discussed in Chapter 3. With risk sources detected in the sensor network, the use of actuation can allow the robotic nodes to interact with the environment and mitigate the detected risks. A network response is formulated with hopes to mitigate the risk types of Sect. 3.2. Such a response calls upon the robotic members of the RSN, F , to assume new locations in the environment to form a more efficient (i.e., less risky) network topology.

The objective of this chapter is to begin the formulation of the network response by creating autonomous groups of robotic members within the RSN. These groups of robot nodes are called **coalitions**, C . These coalitions are centralized around the detected risks and each coalition executes an optimization task, T . This problem is typically called Multi-Robot Task Allocation (MRTA), which involves the allocation of robots to tasks. In Section 2.2 of this report, the taxonomy of MRTA problems was discussed from the work of [18]. One of the types of MRTA problems is called a **Single-Task, Multi-Robot, and Instantaneous Assignment** (ST-MR-IA) MRTA problem. This best fits the description of the problem at hand since the following are true:

1. An RSN contains multiple robots,
2. Each robot will be able to participate in a single optimization task
3. Each optimization task requires multiple robots
4. Robots are allocated to tasks instantaneously

A fuzzy-based MRTA technique is introduced in this chapter, which will effectively take a complete set, F and partition the set into the aforementioned coalitions. Let C_i be the i^{th} coalition of robotic nodes, where $C_i \subseteq F, \forall i \in \{1 \dots N_{tasks}\}$ for N_{tasks} active optimization tasks. A robotic node can only be assigned to a single group and as a result can only participate in a single optimization task, such that $C_0 \cap C_1 \cap \dots \cap C_{N_{tasks}} = \emptyset$. A coalition is formed at the time a high-risk event has been detected by a robotic member of F . Robot members are assigned to a coalition instantaneously by means of fuzzy-auction MRTA. One condition of the fuzzy-auction MRTA, is that no robot member can be added to a coalition after the initial robots are assigned.

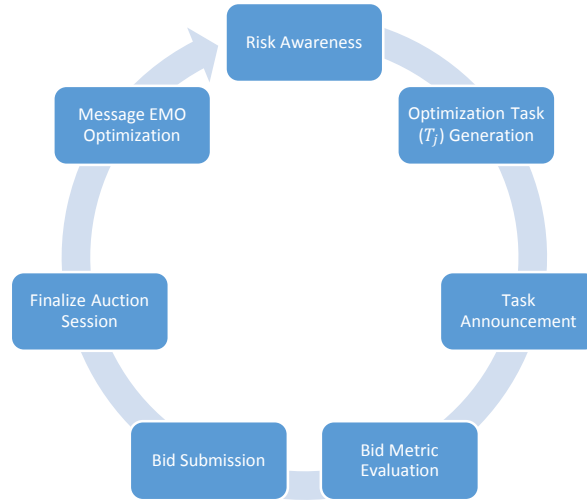


Figure 4.1: The Fuzzy-Auction MRTA Process Cycle

If we consider the process cycle shown in Figure 4.1, the MRTA process of the RMF must follow a series of sequential states before the optimization can begin. The fuzzy-auction MRTA process cycle begins with *risk awareness* identifying a high-risk event. This high risk event is translated into an appropriate optimization task by a self-appointed auctioneer through *optimization task*

generation. At this point the auctioneer will notify all robotic members of the RSN by means of *task announcement*. RSN robot members are presented a chance to undergo a *bid metric evaluation* and then proceed with a *bid submission* back the auctioneer. The auctioneer will eventually *finalize the auction session* and notify the appropriate winners.

In this implemented auction protocol each robotic member of the RSN is able to assume one of three possible robot roles: *idle*, *auctioneer*, or *participant*. In the *idle* role, the robotic member performs its scheduled tasks (e.g., sensor scans) while observing for high-risk events or auction sessions. A robot in the *auctioneer* role, means that the robotic node has detected a high risk event. As an auctioneer, the robotic node has the responsibility to generate an optimization task and host a bidding session to recruit a subset of robots from the total population F to form a coalition, C_j . The auctioneer will then manage the coalition to mitigate risk, R_j . Within the auctioneer role, a robotic member will be in one of four possible states: `ANNOUCE_TASK`, `AWAIT_BIDS`, `CLOSE_AUCTION`, or `OPTIMIZE` – these will be explained further in the sections to follow.

A robot in the *participant* role is a robot that has participated in coalition forming and is a member of a response group. A participant can be in one of four states: `EVALUATE_BID`, `AWAIT_AUCTION_RESULTS`, `AWAIT_INSTRUCTIONS`, or `TIME_OUT`. A participant awaits instructions from the auctioneer in the optimization phase and carries out any necessary commands (e.g., movement vectors). This entire process begins with the auctioneer generating an optimization task.

4.1. Optimization Task Generation

During the process of risk feature extraction and risk assessment conducted by each robot of the RSN, the auction MRTA role of each robot will remain set to *idle* until some risky event is detected. Upon the identification of some risky event, R_j , where j is $N_{risks} + 1$. N_{risks} is the number of detected risks by the risk awareness modules in Figure 3.1 of a single robot, S_{node} . As a result of the detected risk, R_j , the robotic node transition from the idle role to the auctioneer role, to begin the coalition forming process. The initial state for an auctioneer is `ANNOUCE_TASK`. The optimization task, T_j must be communicated to all other robots in the RSN, at which point an auction session hosted by S_{node} can be considered open. The goal of the auction session is to allow robotic members of the RSN to bid on task, T_j in order to reduce the detected risk R_j . Optimization task T_j will contain the necessary optimization criteria to compute an optimal placement of robot members belonging to coalition, C_j . The optimal placement of the coalition members will be guided by the objectives of the optimization task.

4.2. Task Announcement

A risk, R_j , that has been detected by some robotic node S_{node}^j must now proceed with the announcement of optimization task T_j . Auctioneer S_{node}^j , is in state `ANNOUCE_TASK`. In order to construct a coalition, C_j , of robotic nodes, task T_j must be advertised to all members of F . This is accomplished by means of a message broadcast, involving every robotic node, $S_{node}^i, \forall i \neq j$. Each member of the RSN that receives a **task announcement**, $M_{annouce}^j$ message will broadcast it

once to be received by all neighbouring nodes. All redundant task announcement messages are dropped to prevent communication flooding.

The auctioneer is the first robot to broadcast the task announcement message to other listening robots. The auctioneer is also the robot who will await bid responses from other RSN members. The task announcement message is responsible for providing RSN nodes with the required context to evaluate their bid metric in a given auction session. The task announcement message provides the following context (Table 4.1):

1. Auctioneer ID;
2. Risk event type;
3. Risk metric; and
4. Spatial Context

Message Component	Example
Task ID	<i>Integer</i>
Auctioneer ID	<i>Integer</i>
Optimization (Task) Objectives Flags	<ul style="list-style-type: none"> • ReduceMovementEnergy • MaximizeCoverageOfRegion • MinimizeCoverageOfRegion
Spatial Context, P_{R_j}	$x, y, z \in \mathbb{R}$

Table 4.1: Task Announcement Message

Only robotic members in the *idle* role are able to process task announcement messages, otherwise a robotic node is already a participant in a task or an auctioneer hosting an auction session. The handling of a task announcement message is depicted in the flow diagram of Figure 4.2.

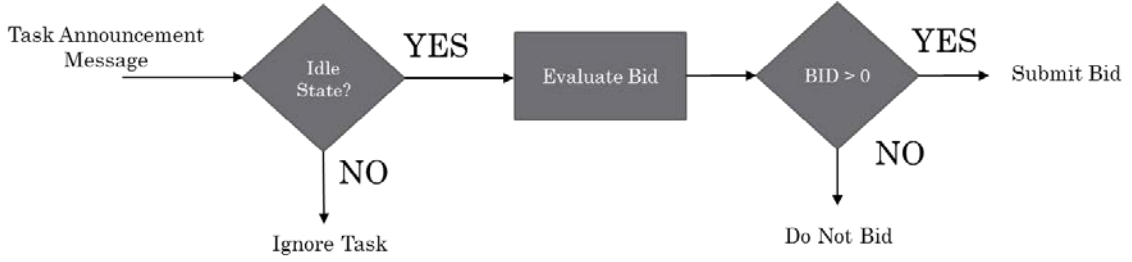


Figure 4.2: A flow diagram depicting the process of handling a task announcement message.

If idle, a robotic node may receive a message $M_{annouce}^j$. This is an announcement message from node index j corresponding to the j^{th} task and forming robot coalition, C_j . The node must process the message and evaluate a bid metric (**bid metric evaluation**) based on the node current state and the information within $M_{annouce}^j$. Upon the generation of a bid metric (as described in the next section) a robot node can decide whether or not to place a bid to the auctioneer. At this stage, the auctioneer has announced T_j , and transitions from state ANNOUCE_TASK to AWAIT_BIDS.

4.3. Bid Metric Evaluation

In this stage of the process, a robot node undergoing bid metric evaluation is still considered as idle. A robot member of the RSN can receive multiple task announcement messages and is required to generate a bid metric for each one. The bid metric, y_{bid} generated by a robotic member, which can be interpreted as the robots fitness for an optimization task. A submitted bid metric of 1.0, indicates to the auctioneer that a robotic node is well-fit for the optimization task. The auctioneer is allowed to rank all bids received based on the total bids received.

A Sugeno FIS is used to by each robotic node to determine its bid metric in an auction session. Let the FIS be denoted by Equation (4.1).

$$y_{bid} = \Psi_{bid}(x_{batt}, x_{dist}, x_{red}) \quad (4.1)$$

4.3.1. Input Variable: Battery Level

The *battery level metric*, x_{batt} is the crisp input to the fuzzy input variable $\varphi_{batt}(x_{batt})$. The battery level metric is an incentive of the robotic node to bid based on battery's energy capacity. The domain of x_{batt} is $D_{x_{batt}} = \{x_{batt} \in \mathbb{R} \mid 0 \leq x_{batt} \leq 1\}$. For example, $x_{batt} = 0$ would indicate that the robotic node battery is completely depleted, whereas $x_{batt} = 1$ would represent a fully-charged battery. The input x_{batt} is fuzzified using fuzzy input variable, φ_{batt} , which consists of a set of membership functions: $\mu_{batt}^{poor}(x_{batt})$, $\mu_{batt}^{average}(x_{batt})$, and $\mu_{batt}^{good}(x_{batt})$. These functions are depicted in Figure 4.3 and are described in Table 4.2. The membership function definitions were acquired experimentally.

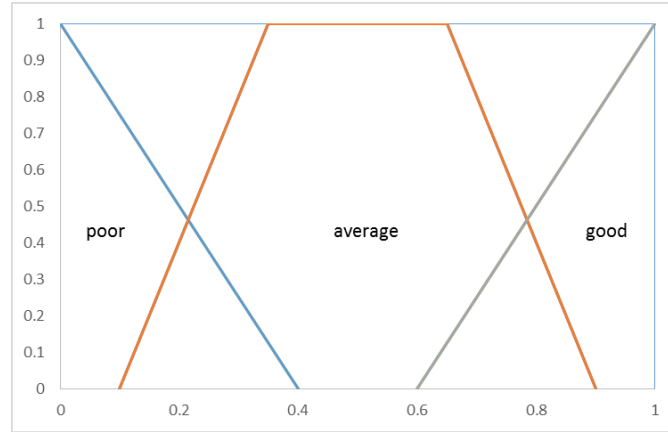


Figure 4.3: The battery level fuzzy input variable for the bid metric FIS

4.3.2. Input Variable: Distance

The *distance metric*, x_{dist} represents the incentive for the robotic node to bid based on the i^{th} node's distance to the risky event, $d_{R_j}^i$. This metric is inversely

propositional to the actual distance away from the risky event, such as defined in Equation (4.2).

$$x_{dist} = \begin{cases} \frac{\gamma_{dist} - d_{NID}}{\gamma_{dist}} & d_{NID} \leq \gamma_{dist} \\ 0.0 & else \end{cases} \quad (4.2)$$

Let $\gamma_{dist} \in \mathbb{R}$ represent the upper distance bound considered. Distance values are mapped to the interval $[0, 1]$, where $x_{dist} = 1$ implies no distance separation of the node to the risky event. On the other hand, $x_{dist} = 0$ implies the robotic node is spatially separated by a distance of at least γ_{dist} . The input value x_{dist} is the crisp input the fuzzy input variable, $\varphi_{dist}(x_{dist})$, which uses membership functions: $\mu_{dTOnid}^{near}(x_{dTOnid})$ and $\mu_{dTOnid}^{far}(x_{dTOnid})$. These membership functions are shown in Figure 4.4 and are defined in Table 4.2. The membership function definitions were acquired experimentally.

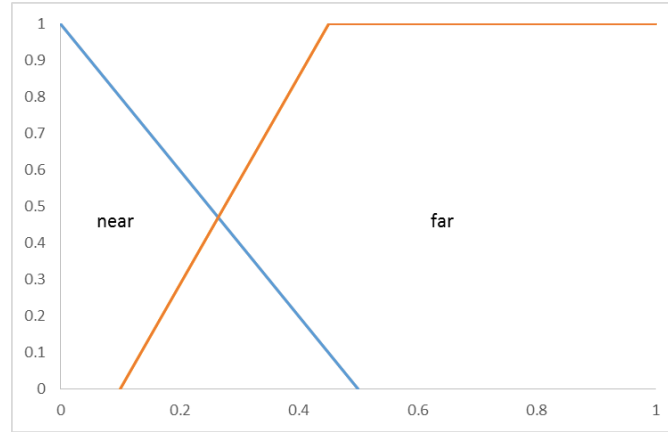


Figure 4.4: The distance fuzzy input variable for the bid metric FIS

4.3.3. Input Variable: Redundant Coverage

The final input is the *redundancy metric*, x_{red}^i which attempts to capture the incentive to place a bid based on the amount of spatially redundant sensor

coverage of member i with all neighboring sensor nodes of F . Each deployed member of the RSN will attempt to monitor its local region of any intrusions or signs of trouble. Let the region monitored by robotic node i , be Ω_{surv}^i , which is a two-dimensional region on the ground surface.

$$\Omega_{inter}^{i,j} = \Omega_{surv}^i \cap \Omega_{surv}^j \neq \emptyset, j \neq i \quad (4.3)$$

Given Equation (4.3), a region of intersection can be generated given all members of F , which represents a region of terrain that is *redundantly* monitored by multiple sensor nodes. The union of all the intersection regions will produce a new region of the redundant coverage, Ω_{red}^i :

$$\Omega_{red}^i = \Omega_{surv}^i \bigcup_{j \neq i} \Omega_{surv}^j \quad (4.4)$$

Therefore, using Equation (4.4) we can define the input x_{red}^i as shown in Equation (4.5).

$$x_{red}^i = \frac{|\Omega_{red}^i|}{|\Omega_{surv}^i|} \quad (4.5)$$

From Equation (4.5) we can clearly see that the input x_{red}^i is a simple ratio of the region redundantly surveyed by F with S_{node}^i and the region surveyed by S_{node}^i . The input value, x_{red} is the crisp input for fuzzy input variable, $\varphi_{red}(x_{red})$. This fuzzy input variable uses membership functions: $\mu_{red}^{low}(x_{red})$, $\mu_{red}^{medium}(x_{red})$, and $\mu_{red}^{high}(x_{red})$. These membership functions are depicted in Figure 4.5 and are defined in Table 4.2. The membership function definitions were acquired experimentally.

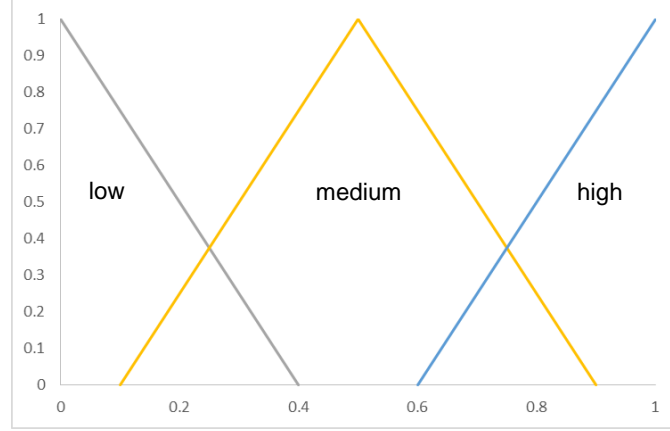


Figure 4.5: The redundant sensor coverage fuzzy input variable for the bid metric FIS

Membership Function, μ	Definition	
	Function Type	Parameters
$\mu_{batt}^{poor}(x_{batt})$	Triangular	A=0.0, B=0.0, C=0.4
$\mu_{batt}^{average}(x_{batt})$	Trapezoidal	A=0.1, B=0.35, C=0.65, D=0.9
$\mu_{batt}^{good}(x_{batt})$	Triangular	A=0.6, B=1.0, C=1.0
$\mu_{dToNid}^{near}(x_{dToNid})$	Triangular	A=0.0, B=0.0, C=0.5
$\mu_{dToNid}^{far}(x_{dToNid})$	Trapezoidal	A=0.1, B=0.45, C=1.0, D=1.0
$\mu_{red}^{low}(x_{red})$	Triangular	A=0.0, B=0.0, C=0.4
$\mu_{red}^{medium}(x_{red})$	Triangular	A=0.1, B=0.5, C=0.9
$\mu_{red}^{high}(x_{red})$	Triangular	A=0.6, B=1.0, C=1.0
$\mu_{bid}^{ideal}(\cdot)$	Constant	1.0
$\mu_{bid}^{available}(\cdot)$	Linear Equation	$\frac{x_{batt}}{3} + \frac{x_{dToNid}}{3} + \frac{x_{red}}{3}$
$\mu_{bid}^{reject}(\cdot)$	Constant	0.0

Table 4.2: The membership functional for all fuzzy input and output variable of the bid metric FIS.

4.3.4. Output Variable: Bid Metric

The output variable, y_{bid}^i of system Ψ_{bid} is the availability of the i^{th} robotic node in the RSN. Given a risky event, R_j and the consequent optimization task, T_j , then y_{bid}^i is a gauge of availability to T_j . In other words, S_{node}^i calculates its fitness to optimization T_j based on: available battery power, distance to the event, and the amount of redundant coverage from surrounding RSN members. Table 4.2 describes the membership functions used in all three fuzzy input variables of Ψ_{bid} as well as three output functions to produce a crisp output. The output functions are labelled as: *ideal*, *available*, and *reject*. Based on a set of fuzzy rules (see Table 4.3) defined within Ψ_{bid} , the appropriate output function is selected to produce a crisp output value.

Fuzzy Rule	Weight
if (Battery is Poor) and (Distance is Near) and (RedCoverage is Low) then (Bid is Reject)	1.0
if (Battery is Poor) and (Distance is Far) and (RedCoverage is Low) then (Bid is Reject)	1.0
if (Battery is Poor) and (Distance is Near) and (RedCoverage is Medium) then (Bid is Available)	1.0
if (Battery is Poor) and (Distance is Far) and (RedCoverage is Medium) then (Bid is Reject)	1.0
if (Battery is Poor) and (Distance is Near) and (RedCoverage is High) then (Bid is Available)	1.0
if (Battery is Poor) and (Distance is Far) and (RedCoverage is High) then (Bid is Reject)	1.0
if (Battery is Average) and (Distance is Near) and (RedCoverage is Low) then (Bid is Ideal)	1.0
if (Battery is Average) and (Distance is Far) and (RedCoverage is Low) then (Bid is Reject)	1.0
if (Battery is Average) and (Distance is Near) and (RedCoverage is Medium) then (Bid is Ideal)	1.0
if (Battery is Average) and (Distance is Far) and (RedCoverage is Medium) then (Bid is Reject)	1.0
if (Battery is Average) and (Distance is Near) and (RedCoverage is High) then (Bid is Ideal)	1.0
if (Battery is Average) and (Distance is Far) and (RedCoverage is High) then (Bid is Ideal)	1.0
if (Battery is Good) and (Distance is Near) and (RedCoverage is Low) then (Bid is Ideal)	1.0

if (Battery is Good) and (Distance is Far) and (RedCoverage is Low) then (Bid is Available)	1.0
if (Battery is Good) and (Distance is Near) and (RedCoverage is Medium) then (Bid is Ideal)	1.0
if (Battery is Good) and (Distance is Far) and (RedCoverage is Medium) then (Bid is Available)	1.0
if (Battery is Good) and (Distance is Near) and (RedCoverage is High) then (Bid is Ideal)	1.0
if (Battery is Good) and (Distance is Far) and (RedCoverage is High) then (Bid is Ideal)	1.0

Table 4.3: The fuzzy logic rule-set for the bid metric FIS

The rules defined in Table 4.3 attempt to capture the expertise of a human operator in deciding, which robotic nodes to allocate to task T_j for coalition C_j . Undesirable conditions, such as a node with a POOR battery, that is FAR away from the risky EVENT, with LOW redundant coverage will use output function REJECT (i.e., $y_{bid} = 0.0$). There also are more favourable conditions such as GOOD battery, NEAR the risky event and MEDIUM to HIGH redundant coverage which use the IDEAL output function (i.e., $y_{bid} = 1.0$). Rules are defined to capture the in-between conditions, such as what happens when a robotic node has a POOR battery, but is NEAR the risky event, with HIGH redundant coverage. A POOR battery is not IDEAL, but the node is considered AVAILABLE. Such nodes meeting conditions in between REJECT and IDEAL use the AVAILABLE output function (i.e., $y_{bid} = \frac{x_{batt}}{3} + \frac{x_{dToNid}}{3} + \frac{x_{red}}{3}$).

Once robotic node, S_{node}^i has computed its bid metric, y_{bid}^i , then this information must be sent back to the auctioneer hosting the auction session for T_j so that a coalition C_j can be formed and risky event R_j can be mitigated.

4.4. Bid Submission

A robotic member in the *idle* state will process each task announcement message received and will determine a bid metric as explained in the previous section. If a node S_{node}^i is *idle* and processes a task announcement message, $M_{annouce}^j$, will generate a bid metric, y_{bid}^i . A **bid submission** message, $M_{bid}^{j,i}$ is used as a means of submitting the generated bid metric, y_{bid}^i to an auctioneer. This message requires the transmission of:

1. Node Identifier; and
2. Bid Metric

Table 4.4 shows the contents of the bid submission message.

Bid Message Component	Values
Node ID	integer
BID Metric	0.0 to 1.0

Table 4.4: The bid submission message used by a robotic node to submit a bid to an auctioneer

If possible, this message would not be transmitted via broadcasting protocols, but routed directly to auctioneer, S_{node}^j . Section 4.3.4 stated that there are three possible output functions, which can be selected by the fuzzy inference system, Ψ_{bid} : *ideal*, *available*, and *reject*. A robotic node will only submit a bid (i.e., transmit a bid submission message) for output function *ideal* and *available*. If *reject* was selected as the output function in Ψ_{bid} , which implies that $y_{bid}^i = 0$, then the robotic member will remain silent and remain its *idle* state. A robotic node is not restricted to bidding in a single auction session. For each task announcement

message, a bid metric will be produced and sent (if not reject). Upon sending each bid submission message, each robotic node will await a **finalize auction session**, $M_{finalize}^j$ message from any of the active auctioneers. This message will notify a robotic node if it has been selected to join a coalition for a risk mitigation task.

4.5. Finalize Auction Session

This stage of the process concludes the fuzzy-auction MRTA process. Up to this point, an auction session was initiated due to a risky event. This instantiated a broadcast announcement of the auction session and then a bid submission process by each robotic node capable of participating in an auction session. During all of this, the auctioneer remains in state `AWAIT_BIDS`, processing each bid submission it receives. An auctioneer will remain in state `AWAIT_BIDS` until the following stop condition is satisfied: the elapsed time in state `AWAIT_BIDS` exceeds a timeout threshold ($t_{bid} > \delta_{bid}^{timeout}$).

When the stop condition is met, the auctioneer will transition to state `CLOSE_AUCTION` or regress to `ANNOUNCE_TASK`. In the case of the stop condition, the auctioneer will track the time since $M_{annouce}^j$ was sent to all robots in the RSN. If no bids are not received by the time $\delta_{bid}^{timeout}$ has elapsed, the auctioneer will regress to the `ANNOUNCE_TASK` state and re-open the auction. Essentially the robot auctioneers are able to re-open auction if no bids have been placed. As an auctioneer received bid submission messages, they are handled in the manner depicted in Figure 4.6.

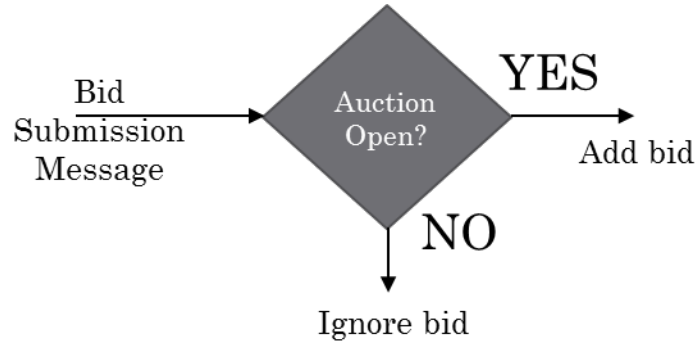


Figure 4.6: The process executed by an auctioneer robot to handle bid submission messages.

During the processing of a single bid submission message, the robot must first check whether it is hosting an open auction session. If no auction session exists, the received bid will be discarded. Otherwise, the bid will be added to the collection of received bids. At the moment of a triggered event to stop listening for bids, the received bids are sorted in descending order based on bid metric value of each bid. The top N_{bids} will determine the winning robots and must be notified by the auctioneer. Figure 4.7 illustrates this.

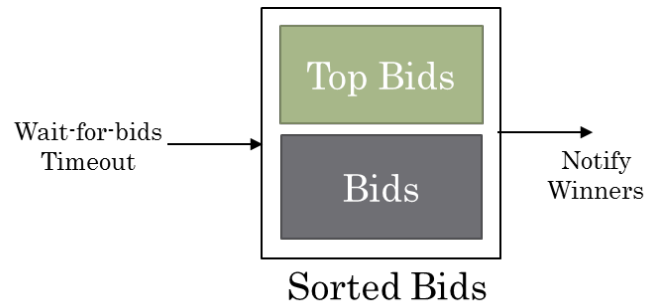


Figure 4.7: After a timeout period the auctioneer will take the top bids within the entire collection of received bids.

The auctioneer is aware of the winning robots and must transition from state `AWAIT_BIDS` to state `CLOSE_AUCTION`, where each winning robot will be notified. Winning robots are notified through a *close auction message* representing the final steps of the coalition forming process. Recipients of a close auction message will undergo a decision process as shown in Figure 4.8.

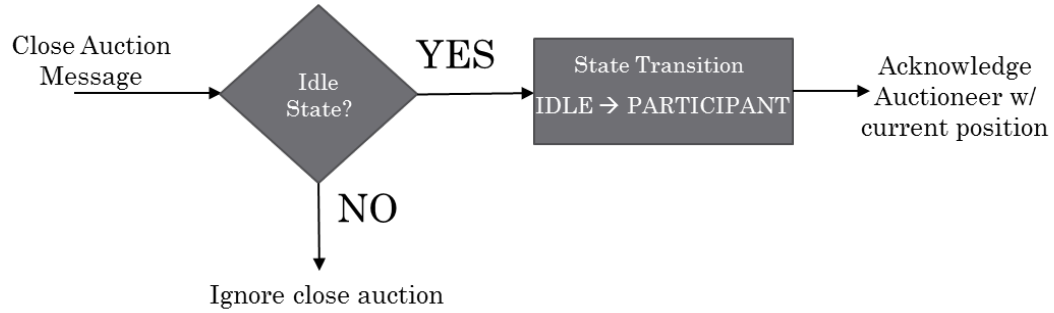


Figure 4.8: The process of handling a close auction message.

Upon receiving a close auction message, a robot first must verify that it is current in the *idle* role. Only a robot in the *idle* role is able to join a coalition. If the robot is *idle*, then the robot will transition its role from *idle* to *participant*. An acknowledgement is sent back to the auctioneer along with the robot's current position. The current position is necessary information in order to execute the optimization task, T_j . It is at this point that the coalition forming process concludes and what is left is an independent robot response group.

4.6. Chapter Summary

In this chapter, a fuzzy auction-based task allocation algorithm is discussed for multi-robot task allocation. This chapter was segmented into five sections, each discussing a primary stage of the algorithm as depicted in Figure 4.1. The first section defined an optimization task in which the MRTA algorithm is selecting robotic nodes. A task, T is generated as a consequence of a detected risk threat in the network. This task gets advertised to the robotic population by a self-assigned auctioneer in a task announcement message. The task announcement process is the main focus of section two.

The third section described how available robotic nodes interpret the task announcement message. Robotic nodes will undergo a bid metric evaluation

process. Nodes take into account their battery level, distance to the risky event, and current amount of redundant coverage in order to calculate a single bid metric.

The fourth section discussed how these bids are transmitted back to the auctioneer. Based on the bid metric produced in section three, a node decides whether or not to place a bid. Nodes which produce a bid metric, y_{bid} using the *reject* output fuzzy membership function will not place a bid.

For nodes placing a bid in an auction session, the bid submission process was discussed in the fourth section of this chapter. A simple (bid submission) message is constructed by the robotic node containing the node's identification information and its submitted bid value. This message is transmitted and is to be received by the auctioneer.

The fifth section explains the conclusion of the task allocation process. Received bid submission messages are processed by the auctioneer and ranked. The ranking is sorting the bid submissions based on the bid metrics where the greater the bid metric value the better fitted an asset is to the optimization task. When more than N_{bids} are received, the best N_{bids} are considered. At this point, a maximum of N_{bids} robotic nodes have been allocated to the optimization task, T .

Chapter 5. Evolutionary Multi-Objective Optimization for Self-Organization of RSN

In this chapter, the self-organization ability of the RSN will be described in detail while linking its usability to the techniques described in the two previous chapters. The ability of an RSN to understand risk and raw data features from sensor instruments enables an awareness for non-optimal network configurations. The risk awareness techniques described in Chapter 3 allow the RSN to perceive the risks of the deployment based on: *degree of distress risk* and *potential intrusion risk*. The fuzzy-auction MRTA technique described in Chapter 4 allocates the necessary robotic members of the RSN into response groups, called *coalitions*. Each coalition, C_j , exists for the purpose of a task, denoted T_j , which is the result of a identified risk, R_j . Let the objectives of the task, T_j , be the set, O_j , where $O_j \subseteq \mathbf{O}$, where \mathbf{O} Equation (5.1) is the set of all considered objectives.

$$\mathbf{O} = \{f_{coverage_{perim}}(\cdot), f_{coverage_{region}}(\cdot), f_{energy_{cost}}(\cdot)\} \quad (5.1)$$

The problem discussed in this chapter involves tasks requiring the concurrent *optimization* of *multiple* objective functions. Unfortunately these objectives will likely be conflicting ones, such as the case with optimizing coverage and reducing the energy exerted by the network. The Non-Dominated Sorting Genetic Algorithm II (NSGA-II) [12] – an **Evolutionary Multi-Objective (EMO) Optimization Algorithm** – searches the solution space for optimal, non-dominated solutions for T_j . An EMO approach to this problem because it proved successful in the application discussed in [10]. This approach can also accommodate additional optimization objectives as required without the need to

reformulate the entire optimization algorithm. As depicted in Figure 3.1, the MRTA and potential responses provide the necessary optimization criteria and response alternatives to the EMO optimization. This chapter is going to discuss the following:

1. The **optimization criteria** generated from risk assessment;
2. The **potential responses** feeding into the NSGA-II;
3. The **objective functions'** criteria from the decision makers;
4. The chromosome design of the NSGA-II; and
5. The mutation and crossover operators.

5.1. Optimization Criteria

The optimization criteria provide the essential context to the optimization problem. In Chapter 3, a risk model and risk assessment technique was introduced. The RSN is in some initial state and is undergoing strenuous risk evaluations to highlight potential threats in the network. The initial state of the RSN, refers to the initial topological configuration of the network, which includes the position of each robotic member. The context included in the optimization criteria includes:

1. Defensive Region, $\Omega_{defense}$;
2. Risk Region, Ω_{risk} ;
3. Risk Type, T_{risk} ; and
4. Coalition, C

A **defensive region**, $\Omega_{defense}$ is the spatial region that the RSN is deployed to monitor and defend. In the case of CIP, the defensive region would encapsulate

the critical infrastructure. The robotic sensor nodes are equipped with sensor instruments to observe a subset of the defensive region, as given by $\Omega_{coverage}^i \subset \Omega_{defense}$ for the i^{th} robotic node in the RSN. Although the surveillance of as much of the defensive region as possible is of importance, more care is taken into the detection of unauthorized personnel and vehicles entering and leaving the defensive region. That being said, for $\Omega_{defense}$ a contour can be defined, which can represent a **security perimeter** around $\Omega_{defense}$. The **risk region**, Ω_{risk} is a spatial region representing the risky event detected by a robotic member of the RSN. The **risk type**, T_{risk} is a categorization of a detected risk event. In Chapter 3, multiple risk features were defined to observe different forms of risk in the RSN's environment. This included *internal* and *external* risks, which both affect the objectives of the network.

5.2. Potential Responses

The NSGA-II will search the solution space for non-dominated Pareto optimal solutions. This solution space is not infinite, but contains a finite number of solutions after the potential responses are generated. Given coalition, \mathcal{C} the number of possible locations for all robotic nodes is expressed by Equation (5.2).

$$n_{solutions}^{tgtPts} = (n_{tgtPts})^{nc} \quad (5.2)$$

The number of possible combinations of selected robots in a coalition is expressed by Equation (5.3).

$$n_{solutions}^{enabled} = 2^{nc} \quad (5.3)$$

Thus, there are $n_{solutions}$, expressed by Equation (5.4) to be explored by the optimization algorithm.

$$n_{solutions} = n_{solutions}^{tgtPts} n_{solutions}^{enabled} \quad (5.4)$$

where $n_C \in \mathbb{Z}$ represent the number of robotic nodes in a coalition. The variable $n_{tgtPts} \in \mathbb{Z}$ is the number of coordinate locations within the response region, $\Omega_{response}$, of each robotic node in C .

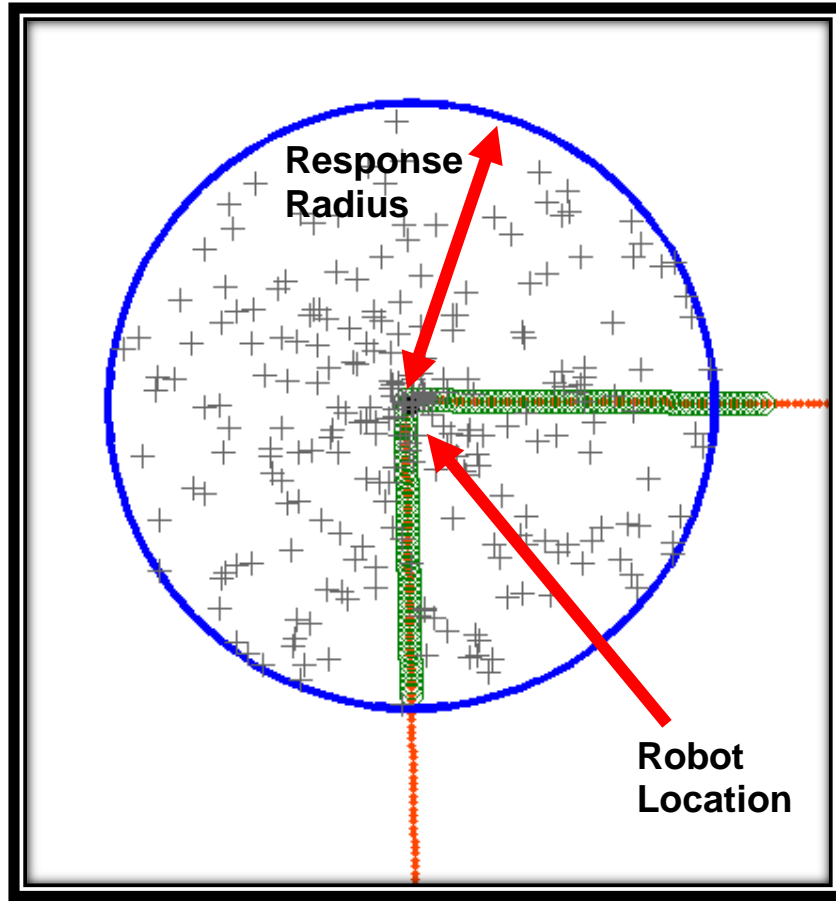


Figure 5.1: The response region of a single robot is a circular region (blue circle) and all randomly generated coordinate locations (grey crosshairs)

The potential response of each robotic member in C are defined by the set of coordinate points, $\mathcal{S}_{tgtPts} = \{P_{tgt_1}^i, P_{tgt_2}^i, \dots, P_{tgt_{N_{tgtPts}}}^i\}$ (depicted in Figure 5.1),

which are constrained by the shape of $\Omega_{response}$. A circular response region was considered, centered about the location of each robot with a radius $R_{response}^i$. Each coordinate location within set \mathcal{S}_{tgtPts} is randomly generated using a uniform distribution. For each single target point $P_{tgt_k}^i(x_{tgt_k}^i, y_{tgt_k}^i)$ the x- and y-coordinate values can be described by (5.5) and (5.6):

$$x_{tgt_k}^i = r_{tgt_k}^i \cos(\theta_{tgt_k}^i) \quad (5.5)$$

$$y_{tgt_k}^i = r_{tgt_k}^i \sin(\theta_{tgt_k}^i) \quad (5.6)$$

where $r_{tgt_k}^i \sim U(0, R_{response}^i)$ and $\theta_{tgt_k}^i \sim U(0, 2\pi)$.

5.3. Chromosome Design

The design of the chromosome captures one possible response by the RSN and its robotic members. A network response is a set of coordinate locations selected for each robotic node that is enabled for action. Each robotic node marked as inactive in a network response remains at its initial location. Robotic members marked as selected will assume a new location based on the selected location index in the chromosome. The selected location index selects a single location from the set of random target locations, \mathcal{S}_{tgtPts} . As such, a chromosome is a collection of genes $\{\alpha_1, \alpha_2, \dots, \alpha_{N_{bids}}\}$, which contain two pieces of information: **robot selected**, $\delta_{enabled}^k$ and **coordinate index**, ϕ_{crdIdx}^k . The NSGA-II will seek solutions which involve the best combination of enabled nodes in the response as well the target locations for each of the enabled nodes. Let a chromosome be defined as expressed in Equation (5.7).

$$Chromosome = [\alpha_1, \alpha_2, \dots, \alpha_{N_{bids}}] \quad (5.7)$$

The set of selected coordinate indices for each robotic node enabled in the response, represents a new network topology for the coalition, C_j . By changing the network topology, the sensor-scanned region of each enabled robotic node will change, targeting the mitigation of any risks detected within the network. The NSGA-II will evolve network responses by trying countless variations, as per the evolutionary process, of coordinate indices for each node in C_j , with each generation of the genetic algorithm producing offspring to converge on Pareto optimal solutions. The work of [7] only considered a chromosome of coordinate indices; however this usually produced solutions with unnecessary energy expenditures. By considering a second layer of information to indicate the involvement of a robotic node in a response, the GA is able to search for solutions using subsets of robotic member of C_j . The goal is to find the simplest solution to meet the fitness of the objective functions. For example, if only one robotic node is required to respond to meet the fitness requirement of the objective functions, then that should be the most suitable response for the coalition.

5.4. Objective Functions

As discussed in the beginning of this chapter, a set of all possible objective functions have been defined in Equation (5.1) for any given task T_j . A task T_j is generated with the intention to modify a segment, C_j topology to mitigate the detected risk. The objective functions assigned to T_j , denoted O_j , will directly affect how the network topology will be changed. Each objective function will add a very unique characteristic to the topology changes of C_j and usually the set O_j will

contain objective functions that conflict with one another, such as the simultaneous minimization of redundant coverage while minimizing energy expenditures.

5.4.1. Coverage of Perimeter Objective Function

One possible goal of an optimization task, T , is to increase the sensor coverage over a security perimeter (see Figure 5.2 (a)) or a security perimeter segment (see Figure 5.2 (b)). The coverage of perimeter objective function evaluates the amount of the security perimeter or segment of the security perimeter is actually surveyed by the robotic members of C_j . Hence, this objective function is a crucial performance tracker of an optimization task trying to enhance the coverage of a security perimeter or a security segment.

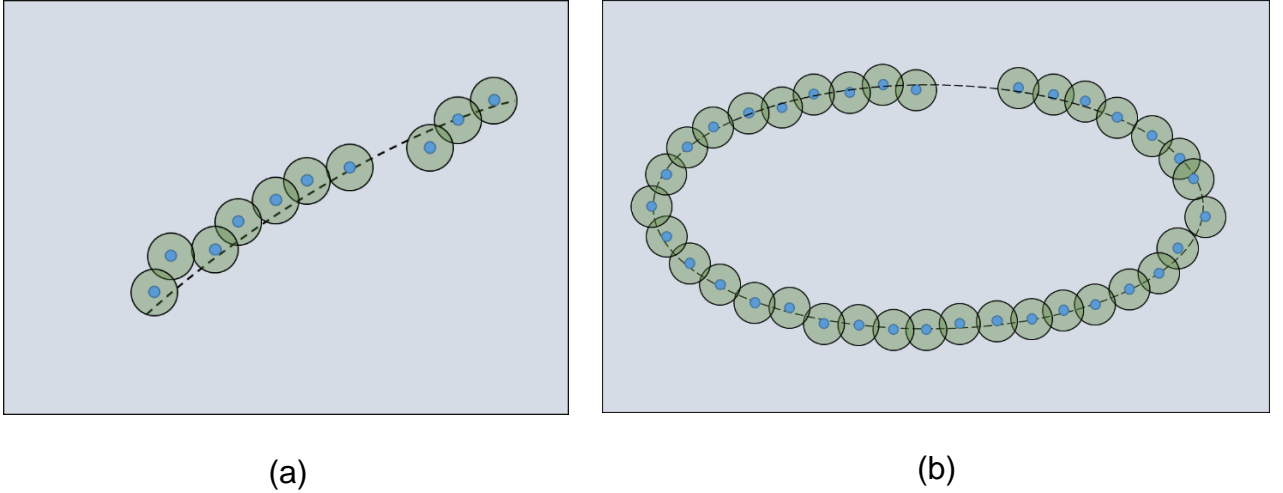


Figure 5.2: Illustration of robots providing sensor coverage of a security perimeter segment (a) and security perimeter contour (b)

A task using this objective function can focus sensor coverage along a specific contour or segment to erect a virtual fence around some critical infrastructure (e.g., embassy, military base, protected zone, etc.). Given a line

segment over $l_{security}$ it can be represented by discrete points along the line, as described in Equation (5.8).

$$l_{security} \xrightarrow{\text{discretize}} S_{points}^{security} = \{P_{security}^1, P_{security}^2, \dots, P_{security}^{N_{points}}\} \quad (5.8)$$

With a set of points, S_{points}^i and a sensor coverage region $\Omega_{coverage}^i$ of $S_{node}^i \in F$, the $S_{surveyed}^{security}$, where $S_{surveyed}^i \subseteq S_{points}^{security}$ can be determined which represents the set of discrete security points surveyed by S_{node}^i (see Figure 5.3).

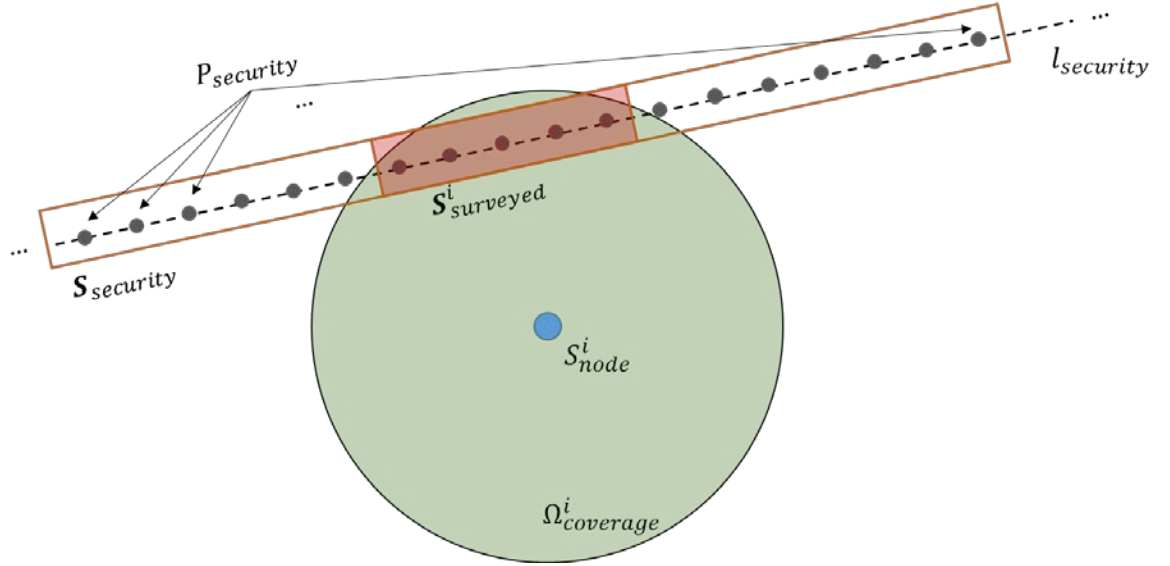


Figure 5.3: Robot Node Sensor Coverage of $l_{security}$ segment

The function $f_{coverage_{perim}}(l_{security}, \Omega_{coverage}^i)$ takes in the following parameters: security line segment and the coverage region of the robotic node to produce an output, $y_{coverage_{perim}}$ with range of 0.0 to 1.0. A set of discrete points are generated producing set $S_{security}$ containing N_{points} (user defined) elements. Given the definition of $\Omega_{coverage}^i$, a set $S_{surveyed}^i$ (representing the number of elements) of $S_{security}$ intersect with $\Omega_{coverage}^i$. Effectively this is determining the

number of discrete points of $l_{security}$ that are within the sensor coverage region of robotic node i . If we let $\Omega_{coverage}^i$ be a circular region as shown in Figure 5.3, then the coverage region can be represented by a center coordinate, $P_{node}^i(x_{node}^i, y_{node}^i)$ and a radius, r_{sensor} . Let $P_{security}^m \in S_{surveyed}^i$ iff the equality in Equation (5.9) is valid.

$$|P_{node}^i - P_{security}^m| \leq r_{sensor} \quad (5.9)$$

Thus, we can define the output of the objective function $f_{coverage_{perim}}$ with Equation (5.10).

$$y_{coverage_{perim}}^i = f_{coverage_{perim}}(l_{security}, \Omega_{coverage}^i) = \frac{|S_{surveyed}^i|}{|S_{security}|} \quad (5.10)$$

5.4.2. Coverage of Region Objective Function

This objective function, $f_{coverage_{region}}(\cdot)$ evaluates the performance of a robotic node's coverage of a specific region, $\Omega_{security}$. The region $\Omega_{security}$ can be defined as a ***spatial region over ground requiring an increased sensor coverage***. This region can be autonomously defined based on custom algorithms or manually defined by a human operator. As seen in section 5.4.1, where the coverage over a line was considered, the line segment $l_{security}$ was discretized into evenly distributed points. A similar approach is taken here, where the security region $\Omega_{security}$ is discretized as described by Equation (5.11).

$$\Omega_{security} \xrightarrow{\text{discretize}} S_{points}^{security} = \{P_{security}^1, P_{security}^2, \dots, P_{security}^{N_{points}}\} \quad (5.11)$$

The desired output of this objective function is $y_{coverage_{region}} \in [0,1]$ representing the intersection region, $\Omega_{security} \cap \Omega_{coverage}^i$ as depicted in Figure 5.4.

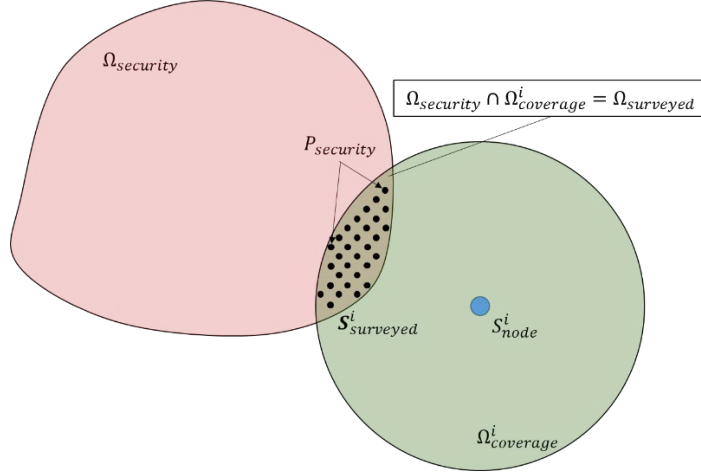


Figure 5.4: A robotic node with circular sensor coverage surveying a subset of a region

This objective function is a function of $\Omega_{coverage}^i$, $\Omega_{security}$, and $P_{tgt}^i \in \mathcal{S}_{tgtPts}^i$ such that $f_{coverage_{region}}(\Omega_{security}, \Omega_{coverage}^i, P_{tgt_k}^i)$. Considering $\Omega_{coverage}^i$ as a circular region as depicted in Figure 5.4 with center $P_{node}^i(x_{node}^i, y_{node}^i)$ and radius, r_{sensor} we can determine the set $\mathcal{S}_{surveyed}^i$, which are the discrete points from Equation (5.11) within sensor range of the robotic node. Therefore, let $P_{security}^m \in \mathcal{S}_{surveyed}^i$ iff the inequality in Equation (5.9) proves true. The output of this objective function can now be properly represented in Equation (5.12).

$$y_{coverage_{region}}^i = f_{coverage_{region}}(\Omega_{security}, \Omega_{coverage}^i, P_{tgt_k}^i) = \frac{|\mathcal{S}_{surveyed}^i|}{|\mathcal{S}_{security}|} \quad (5.12)$$

5.4.3. Energy Cost (Response) Objective Function

The objective function $f_{energy}(\cdot)$ calculates the approximated energy expenditure of a robotic node given a new spatial location, P_{tgt}^i , and the node's

original location, P_{node}^i . With Equation (5.13), a distance metric, d_{tgt}^i can be calculated for the i^{th} robotic node.

$$d_{tgt}^i = |P_{node}^i - P_{tgt_k}^i| \quad (5.13)$$

In combination with approximate battery consumption rates for that robotic node a cost metric, y_{energy}^i can be calculated. Let $\delta_{movement}^i$ be the battery consumption rate (in percentage) per meter travelled. The output of this objective function is expressed in Equation (5.14).

$$y_{energy}^i = f_{energy}(P_{node}^i, P_{tgt_k}^i) = d_{tgt}^i \delta_{movement}^i \quad (5.14)$$

5.5. Crossover and Mutation Operators

Genetic algorithms employ crossover and mutation operations in order to produce new members of the population. In section 5.3, the chromosome design was introduced as a set of genes, α , each containing an integer-based gene, φ_{crlidx}^k , and a Boolean-based gene, $\delta_{enabled}^k$. The pairing of these two pieces of information represents one possible spatial configuration of the robotic nodes in a coalition, C_j . This spatial configuration also happens to represent a network response or solutions to the optimization task T_j . Therefore, the manipulation of φ_{crlidx}^k or $\delta_{enabled}^k$ will change the spatial configuration of C_j . These two gene values can be modified by two separate and independent operations: **crossover** and **mutation**.

5.5.1. Crossover Operator

The crossover operation allows multiple chromosomes to combine genetic information to produce multiple new offspring members. Given objective functions

O_j for task T_j the crossover is used to exchange φ_{crdIdx} and $\delta_{enabled}$ values aiming to produce a new chromosome with better fitness of each element in O_j . If we consider a task with objective function $f_{coverage_{region}}(\cdot)$ and $f_{energy}(\cdot)$, it may be the case where two different members best fit one objective function each. For instance, $Chromosome_m$ maximizes $f_{coverage_{region}}$ and $Chromosome_n$ ($m \neq n$) minimizes f_{energy} . Performing a crossover operation of chromosome member indices m and n will hopefully produce a new chromosome member that can achieve a high fitness to both objective functions.

A chromosome consists of two layers: *robot selected*(δ) and *coordinate index* (φ). This work uses dual crossover operation involving a *uniform* crossover on φ and a *single-point* crossover on δ . The uniform crossover exchanges gene information one-by-one randomly from a parent. This is a more exploratory approach as the structure of the gene information of either parent is not retained. In other words, the child chromosome contains very small gene segments from all parent chromosomes. Data in the coordinate index layer of the chromosome is exchanged in this manner. As described in pseudo code of Figure 5.5, when initializing the gene values of a new chromosome member (child) while iterating over each gene value, genes are randomly exchanged from parent A or parent B.

```

for all coordinateIndex_genes in child
  if (randomDouble > 0.5)
    set child gene value at i from parent A gene value at i;
  else
    set child gene value at i from parent B gene value at i;
  next i;

```

Figure 5.5: Pseudo code for uniform cross over

In the case of the asset enabled layer in each chromosome a single-point crossover operation is used. A single cross over point, $x_{crossPt}$ must be randomly chosen, such that $x_{crossPt} \in \{1, 2, \dots, N_{bids}\}$. The gene values from both parents are split at this point which creates four sets as follows:

$$\mathbf{s}_{parentA}^{left} = \{\delta_{parentA}^1, \dots, \delta_{parentA}^{x_{crossPt}}\}$$

$$\mathbf{s}_{parentA}^{right} = \{\delta_{parentA}^{x_{crossPt}+1}, \dots, \delta_{parentA}^{N_{bids}}\}$$

$$\mathbf{s}_{parentB}^{left} = \{\delta_{parentB}^1, \dots, \delta_{parentB}^{x_{crossPt}}\}$$

$$\mathbf{s}_{parentB}^{right} = \{\delta_{parentB}^{x_{crossPt}+1}, \dots, \delta_{parentB}^{N_{bids}}\}$$

Once these sets are generated, we can produce the gene values for the child chromosomes A and B:

$$\mathbf{s}_A^\delta = \mathbf{s}_{parentA}^{left} \cup \mathbf{s}_{parentB}^{right}$$

$$\mathbf{s}_B^\delta = \mathbf{s}_{parentB}^{left} \cup \mathbf{s}_{parentA}^{right}$$

Now two offspring chromosomes have been created based on two parent chromosomes, which are added to the chromosome population as potential solutions to optimization task T_j .

5.5.2. Mutation Operator

Individuals in the population may be mutated, in each generation, as to help explore the solution space and generate new unique solutions. Although a particular individual and genetic information within may represent a solution with

high fitness to the objective functions of the task, a slight perturbation of some genetic information may increase the fitness. For this reason, the genetic information of each individual is subject to a random value test to decide if a piece of genetic information is to be mutated. One of the optimization algorithm input parameters is the probability of mutation denoted $\rho_{mutation}$. A simple bit flip operation occurs for gene δ^k that is randomly selected for mutation. In the case of φ^k that is selected for mutation, a new integer value of the set $\{1, 2, \dots, N_{bids}\}$ is assigned.

5.6. Chapter Summary

In this chapter an overview is provided on the topic of the self-organization behavior using EMO optimization. The NSGA-II is selected for this optimization. The first section of the chapter described the input criteria for the NSGA-II. Thus far, we define a risk event, \mathbf{R} , which invokes the creation of optimization task, \mathbf{T} . A robotic coalition, \mathbf{C} is formulated around \mathbf{T} . The input criteria includes: \mathbf{T} , \mathbf{C} , and $\Omega_{defense}$, which effectively defines the optimization algorithm problem.

The optimization algorithm will not explore an infinite search space, but will be limited to a set of potential responses. This is discussed in the second section of the chapter. A circular response region, centralized about the robot node's center and radius is a function of maximum mobility range. Random coordinates are generated within these regions, representing all possible locations an asset can traverse to.

The genetic encoding structure used in the NSGA-II is discussed in the third section of this chapter. The NSGA-II initializes a population of individuals encoded

with a genetic schema to represent a single possible node topology within \mathcal{C} . Random coordinates discussed in section two, can be referenced using an index value. Chromosomes are coded with integer values representing coordinate location indices. The integer-based genes within the chromosomes can also be enabled or disabled using a single bit value.

Section four describes the fitness functions used in producing a comparable metric for each solution generated within the individual population. The set \mathcal{O} and all possible subsets (excluding the empty set) are possible objective functions to be used in an optimization task. The fifth section described the mutation and crossover operators performed on the individual population.

Chapter 6. Experimentation

This chapter will provide an evaluation of the proposed risk-aware RSN comprised of three major system components: risk assessment, fuzzy-auction MRTA, and EMO optimization. The entire system can be evaluated by analyzing the risk mitigation response regarding the coverage recovery and the minimization of energy. The fuzzy-auction MRTA process can be evaluated by measuring the reduction of complexity in the optimization problem. MRTA technique will be compared to against a rule-based selection method.

6.1. Experimental Setup

The RSN used to assess the feasibility of a risk-aware RSN was configured in a computer simulator, Microsoft Robotics Developer Studio 4® (MRDS) [26]. A single computer is used to host the simulations, namely a Dell Precision M4700. This unit is comprised of an Intel® Core™ i7-3840QM and 20 GB of RAM. As all simulation was conducted in MRDS, the operation system used was Windows 7 64-bit edition. The implementation required the use of various software packages, the first being General Matrix.NET [27], which is a C# wrapped library of Jama (JaVa Matrix package). General Matrix.NET was used to perform linear algebra operations. Additionally Fuzzy Logic Library for Microsoft.Net was used to provide Mamdani and Sugeno FIS. The implementation of the NSGA-II is based on work conducted at Larus Technologies [10].

The virtual robotic node (shown in Figure 6.1) consists of the following components:

- Simulated Battery

- Differential Drive Platform
- Simulated LRF sensor
- Simulated GPS Module

The simulated battery uses a simple linear decay model to simulate the robotic node's energy demands for regular operations. The usage of sensing devices such as the GPS module or the LRF module will incur addition battery energy. The same situation occurs for any actuation performed by the robotic device, such as enabling the different drive platform required for a robot to self-relocate.

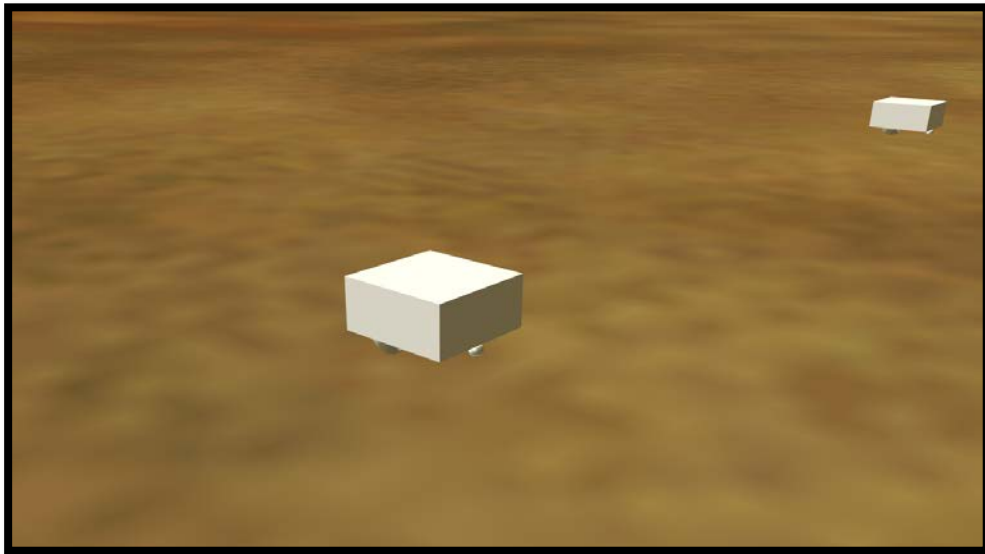


Figure 6.1: A single robotic node consisting of a differential drive platform, a simulated LRF, and a simulated GPS module.

In the scenario used to evaluate the risk-aware RSN, a series of structures will be placed in the simulation environment, representing some form of critical infrastructure. Encapsulating the structures creates a virtual security perimeter, which is defined by drawing a closed contour around the critical infrastructure. To ensure that the initial condition of the risk-aware RSN is ideal, robot nodes are

placed along the virtual perimeter in a non-uniform manner such that as much as possible of the security perimeter is surveyed by the RSN.

6.2. Scenario 1: Factory Complex

The first scenario used in the evaluation of the risk-aware RSN is based on the “Factory” simulation (see Figure 6.2) environments provided with MRDS 4.



Figure 6.2: Scenario 1 “Factory” CIP Scenario

In the proposed scenario a risk-aware RSN comprised of 47 robotic nodes is used. This scenario allows an efficient testing of the concurrent response selection ability for a large-scale deployment of the risk-aware RSN.

6.2.1. Scenario Configuration

This scenario uses a collection of robotic nodes to erect a defense perimeter around the factory complex. Each robotic node uses a simulated

proximity sensor, which is capable of detecting objects up to a distance of 7.5 meters. Each robotic node is allowed to respond up to a distance of 55 meters. Both of these values have been selected experimentally. Robotic nodes are placed non-uniformly along the virtual security perimeter marked in Figure 6.3 by dotted red line segments connecting the user defined points drawn as large solid red dots. The values chosen for user defined points, identified as P1-P4, are displayed in Table 6.1. A linear interpolation provides additional points, 500 in this case, connecting the user-defined points.

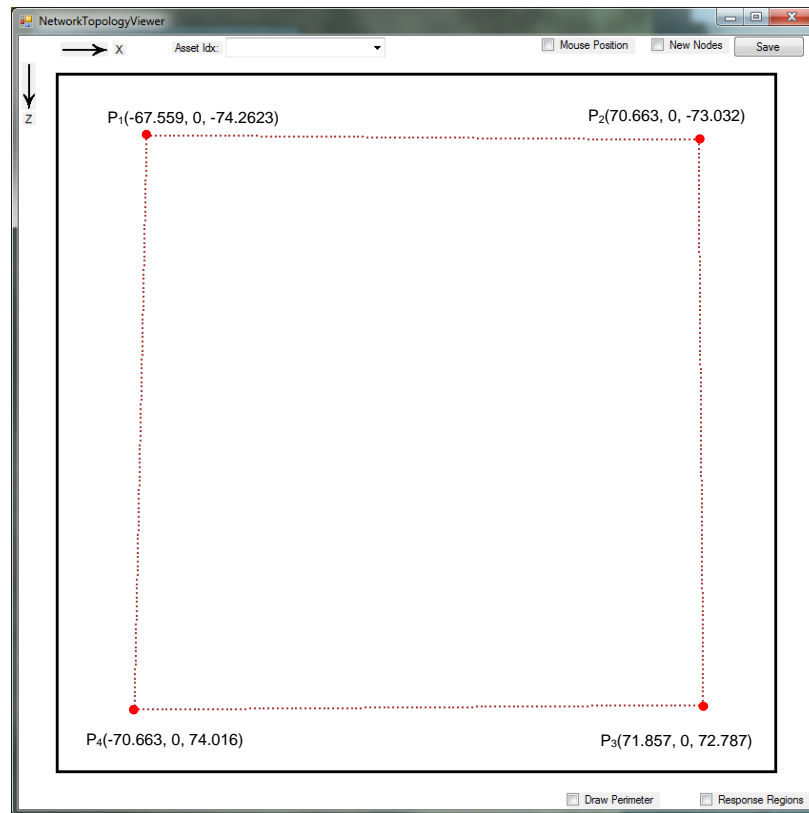


Figure 6.3: The definition of the security perimeter encapsulating the facility. Positive x is in the right direction on the page and positive z is downwards on the page.

Scenario Parameter	Definition	
	Unit	Value
Security Perimeter	Meters, m	P ₁ (-65.560,-74.262), P ₂ (70.663, -73.033), P ₃ (71.857, 72.787), P ₄ (-70.663, 74.016)
Security Perimeter Interpolated Points		500
Maximum Robot Response	Meters, m	55
Robot Battery Usage Rate	% / s	0.5%
Robot Sensor Field of View Radius	Meters, m	7.5

Table 6.1: The scenario parameters used for Scenario 1 "Factory"

The risk-aware RSN is depicted in Figure 6.4 showing all 47 robotic nodes unevenly distributed along the user-defined security perimeter. In the figure, a robotic node is represented by a small black circle and the blue circular region surrounding it represents the sensor field of view of the robotic unit. The scenario is configured such that the initial condition of the risk-aware RSN is surveying the entire security perimeter. The goal of the experimental run is to maintain the sensor coverage of the virtual security perimeter.

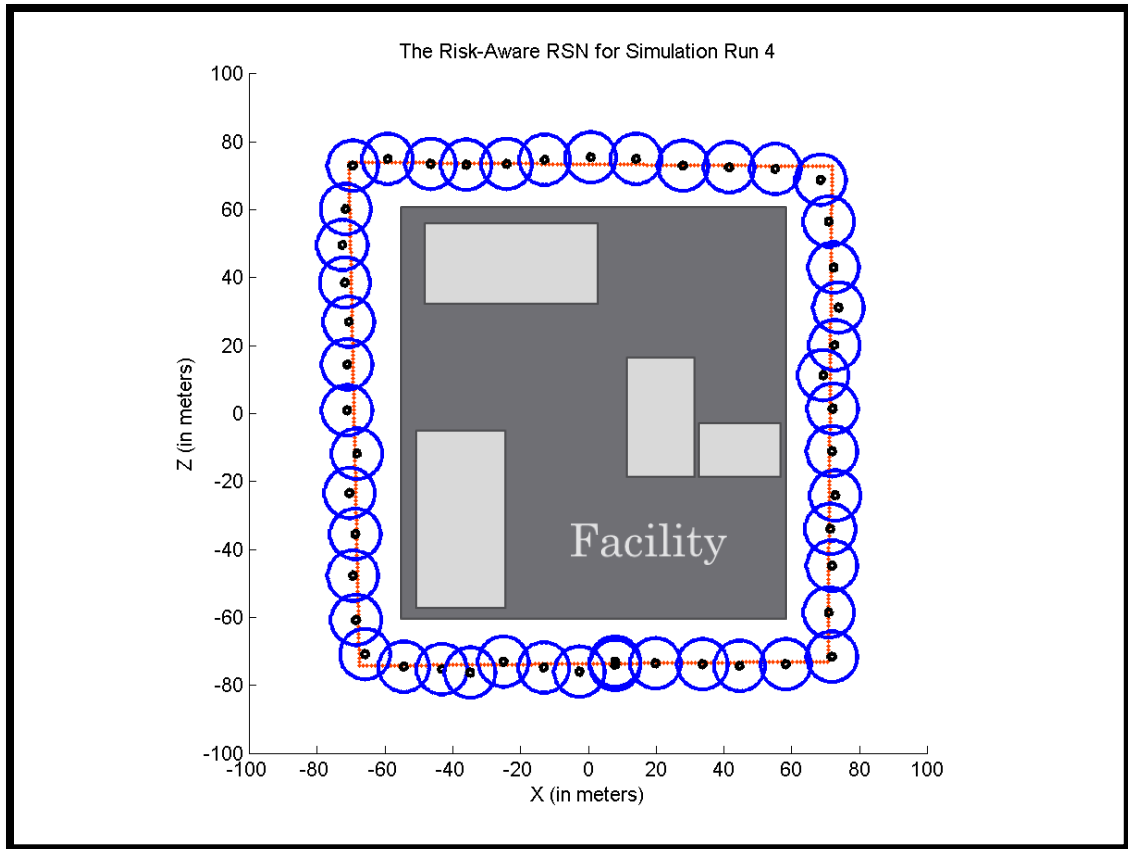


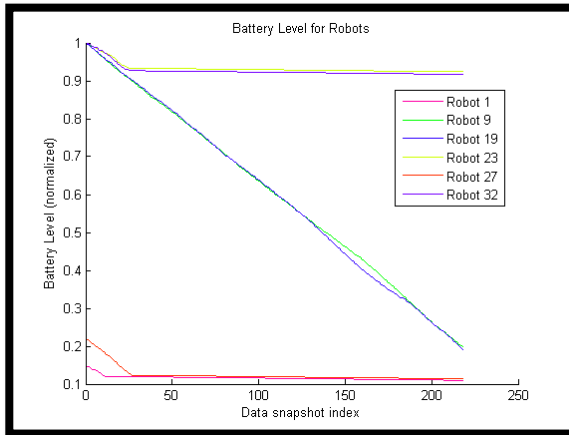
Figure 6.4: The risk-aware RSN robotic nodes as black dots and their respective sensor field of views as blue circular regions.

The first point of observation is the risk assessment. Here the perceived risk by select robotic nodes will be discussed.

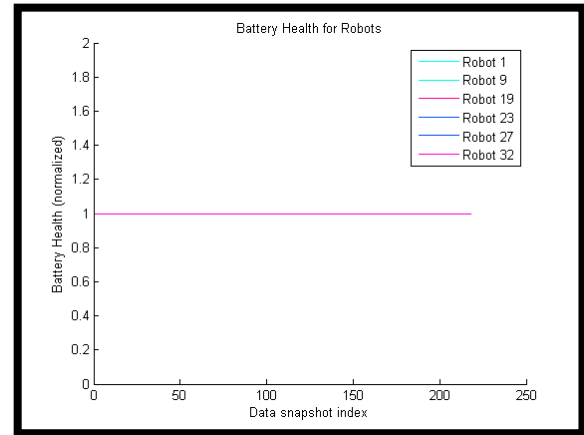
6.2.2. Risk Assessment

A subset of the robotic nodes was selected to observe sensor readings and risk assessment metrics. These robots, selected at random for observations, are: robot 1, robot 9, robot 19, robot 23, robot 27, and robot 32. Robots 1 and 27 are initialized with weak batteries 15% and 22%, unlike the remaining 45 robots which are fully charged. Figure 6.5 is a series of plots, which visualize the battery level

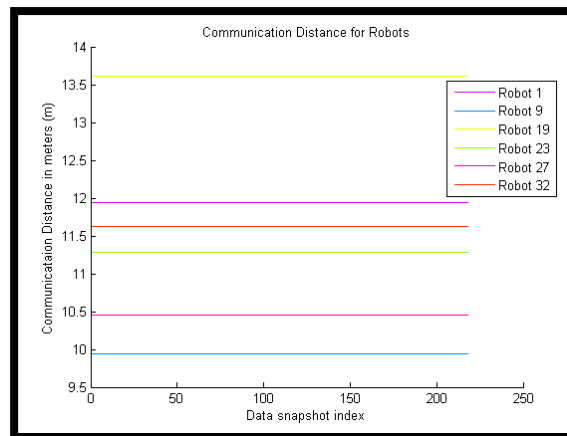
(Figure 6.5a), battery health (Figure 6.5b), and communication distance to the closest neighbor (Figure 6.5c) for the mentioned robots, respectively.



(a)



(b)



(c)

Figure 6.5: Robot sensor readings from randomly selected robots in the RSN, showing (a) the battery level of the selected robots; (b) the battery health of the selected robots; and (c) the communication distance for the selected robots.

As mentioned, robots 1 and 27 begin the scenario with very low battery levels. This can be seen in Figure 6.5 (a), where the battery level of robot 1 is in pink and the orange for robot 27. The battery level is very low for both robots and is the primary risk contribution factor for each robot. In Figure 6.6, a plot of the overall risk is shown for the same selected robots. The overall risk values are

produced by the risk assessment techniques in Chapter 3. The overall risk metrics of robots 1 and 27 have been highlighted as well as the risk threshold set to 0.75.

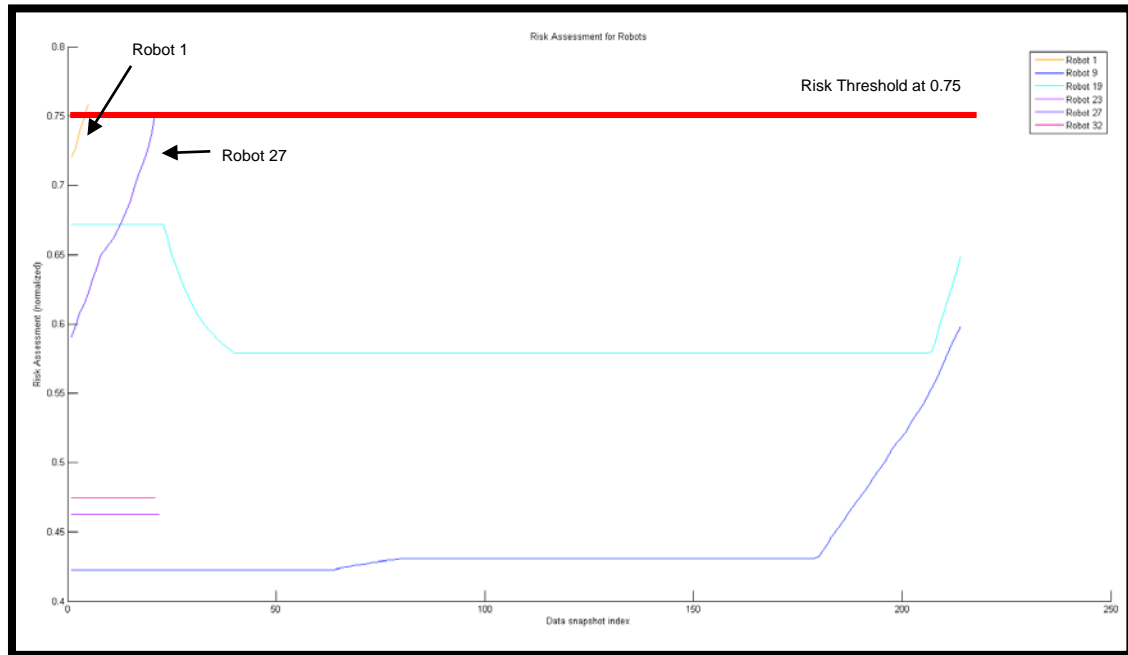


Figure 6.6: Risk assessment plot for select robotic members of the risk-aware RSN

Robot 1 identifies a high-risk event at $t = 8.0$ seconds. The event type is **distress risk event** as a result of its low battery. The high-risk event was triggered with an overall risk of 0.756, exceeding the allowable risk threshold by 0.006. The spatial context for this risk is a collection of virtual perimeter points that robot 1 is providing sensor coverage for. If robot 1 becomes offline, then these points – belonging to the virtual security perimeter – will be unmonitored. This results in a gap in the defence perimeter around the critical infrastructure. The details of the risk event can be found in Table 6.2.

Risk Event	Value
Robot ID	1

Risk Type	NODE_DISTRESS
Spatial Information (units in meters)	{P(13.070, -73.545), P(14.222, -73.534), P(15.374, -73.524), P(16.525, -73.514), P(17.677, -73.504), P(18.829, -73.493), P(19.981, -73.483), P(21.133, -73.473), P(22.285, -73.463), P(23.437, -73.452), P(24.588, -73.442), P(25.740, -73.432), P(26.892, -73.422)}
Overall Risk Metric (normalized value)	0.756

Table 6.2: Risk event details for robot 1 at $t = 8$.

A similar event is triggered by robot 27 on account of its low battery at $t = 27$ seconds. The high-risk distress event was raised with an overall risk of 0.758, exceeding the allowable risk threshold by 0.008. The coverage provided of the virtual perimeter becomes the spatial context for this risk event, given that if this robot goes offline, those perimeter points may be completely un-surveyed by any other robot.

Risk Event	Value
Robot ID	27
Risk Type	NODE_DISTRESS
Spatial Information (units in meters)	{P(-52.124, 73.856), P(-53.282, 73.866), P(-54.441, 73.876), P(-55.600, 73.886), P(-56.759, 73.896), P(-57.917, 73.906), P(- 59.076, 73.916), P(-60.234, 73.926), P(-61.393, 73.936), P(- 62.552, 73.946), P(-63.711, 73.956), P(-64.869, 73.966), P(- 66.028, 73.976)}

Overall Risk Metric (normalized value)	0.758
---	-------

Table 6.3: Risk event details for robot 1 at $t = 27$.

Two robotic nodes in the risk-aware RSN know of the existence of one risk-event each, which pertains to their own distress. These are two independent risk-event requiring two independent responses. The selection of robotic members to participate in a risk mitigation response is discussed next.

6.2.3. Robot Selection using Fuzzy-Auction MRTA

In this section, the use of the fuzzy-auction MRTA technique is considered for the selection of robotic members to construct coalitions. In this current instance, robot 1 and robot 27 have identified themselves as high-risk in the category of distress. As per the fuzzy-auction MRTA technique, each robot in distress will be promoted to the state of *auctioneer*. In this state, a robotic node is capable of initiating the formation of a coalition in order to mitigate the risk detected. In this scenario, robot 1 creates an optimization task (as shown in Table 6.4) based on the information received in the risk event (the information in Table 6.2).

Optimization Task	Value
Auctioneer ID	1
Objective	NODE_DISTRESS
Spatial Information (units in meters)	{P(13.070, -73.545), P(14.222, -73.534), P(15.374, -73.524), P(16.525, -73.514), P(17.677, -73.504), P(18.829, -73.493), P(19.981, -73.483), P(21.133, -73.473), P(22.285, -73.463),

	P(23.437, -73.452), P(24.588, -73.442), P(25.740, -73.432), P(26.892, -73.422)}
Overall Risk Metric (normalized value)	0.756

Table 6.4: The optimization task generated by robot 1 based on a high-risk event

The optimization task created by robot 1 (Table 6.4) is transmitted over the RSN in a *task announcement* message. Each robotic member of the risk-aware RSN receives the task announcement message (if the robot is *idle*) and must calculate its bid value. Table 6.5 shows the battery level (normalized), the redundant coverage (normalized) and the distance to the risk all at the time of receiving the task announcement message. Using the Sugeno FIS described in Chapter 4, the bid metric shown in the last column of Table 6.5 and is crisp output value of the FIS. This metric is sent back to the auctioneer using the *bid submission* message. In this simulation, the following robots (in order) have placed bids: robot 3, robot 44, robot 46, robot 2, robot 4, and robot 45.

Auction 1 Bid Submissions	Battery	Redundant Coverage	Distance to Risk	Bid Value
Robot 2	0.968	0.384	20.595	1.000
Robot 46	0.633	0.583	29.632	0.828
Robot 3	0.980	0.385	31.582	0.801
Robot 45	0.967	0.461	40.047	0.569
Robot 4	0.971	0.154	45.181	0.415
Robot 44	0.969	0.538	51.961	0.121

Table 6.5: The bid submissions for the auction robot 1; ranked in order of bid value

Similarly to 1, robot 27 will undergo the same process. The details of the bid submission for robot 27 are shown in Table 6.6.

Auction 27 Bid Submissions	Battery	Redundant Coverage	Distance to Risk	Bid Value
28	0.924	0.400	17.370	1.000
29	0.913	0.462	23.779	1.000
26	0.861	0.461	19.486	0.993
25	0.893	0.538	29.991	0.856
30	0.906	0.538	31.673	0.835
31	0.893	0.417	40.399	0.542
24	0.893	0.462	41.921	0.490
32	0.893	0.250	50.347	0.323
23	0.893	0.308	53.144	0.171

Table 6.6: The bid submission details for auctioneer robot 27; ranked in order of bid value.

With all of the bids submitted to the respective auctioneers, each auctioneer can close the auction session by sending the *close auction* message. A robotic member will transition from role *idle* to *participant*, upon receiving the first *close auction* message from an auctioneer. At this point, all other *task announcement* messages received by the robotic node will be ignored. This process is illustrated in the log message in Figure 6.7, where robot 2 handles a *task announcement* message from robot 1 and responds with a bid. Shortly after (about a second), robot 2 transitions from *idle* to *participant* indicating that it received a *close auction*

message. As robot 27 raising its high-risk event after robot 1 does, it announces a task which is ignored by robot 2.

63519204095	Initializing Multi Robot Task Allocation...Robot ID is 2
63519204095	Parameters:
	MAX_FUZZY_DISTANCE_INPUT = 55
	AWAIT BIDS TIMEOUT = 1500
	BID QUOTA = 10
63519204103	Handling Task Announcement Message from robot 1
63519204103	Calculate BID metric...
	Batt:0.968298614025116
	RedCov:0.384615361690521
	Distance:20.595308303833
	Bid:1
63519204103	Submitted bid to auctioneer
63519204104	Changing role from IDLE to PARTICIPANT
63519204104	Responding to Close Auction Message with position and identifier
63519204121	Ignoring task announcement, since the robot has the role of PARTICIPANT

Figure 6.7: Log excerpt from robot 2 where the handling of multiple task announcement message at different times is shown

At this point, two independent coalitions have been established: one by robot 1 and the other by robot 27, as shown in Figure 6.8.

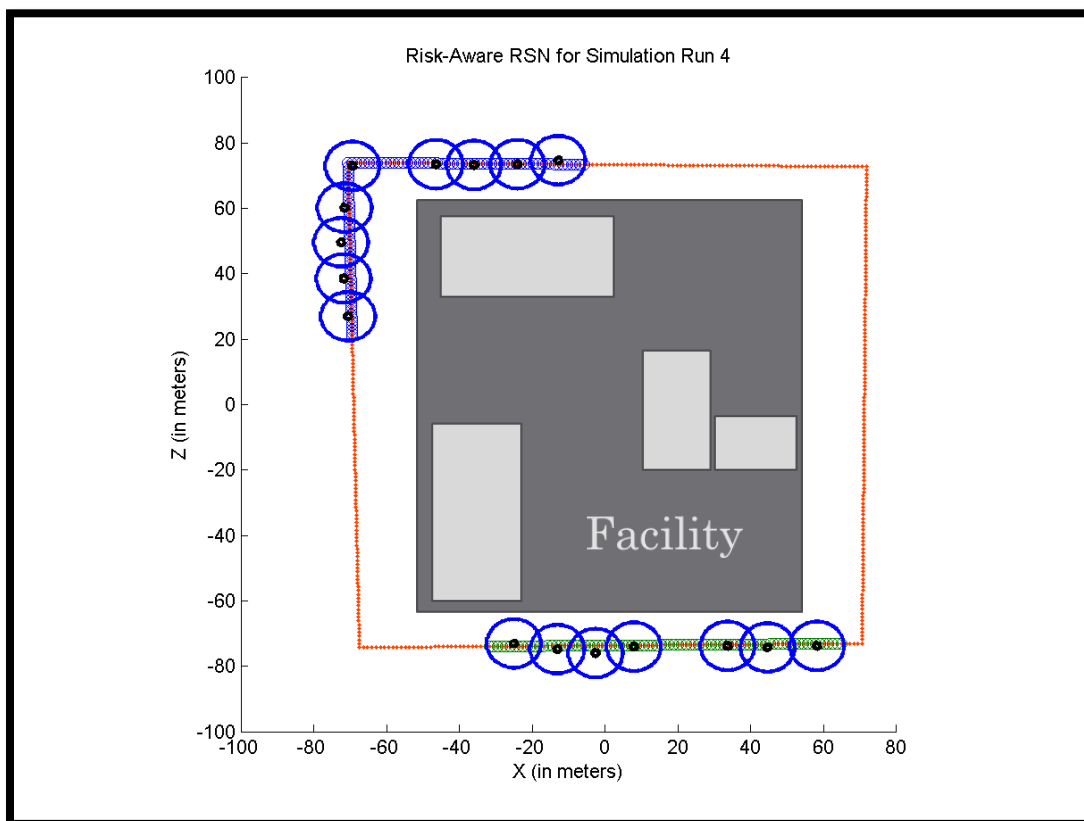


Figure 6.8: Two coalitions formed by robot 1 (group at bottom-right) and by robot 27 (group at top-left).

Figure 6.8 illustrates both coalitions. The coalition formed by robot 1 is located at the bottom right of the figure and the coalition formed by robot 26 is at the top-left of the figure. All other robotic nodes have been hidden for the sake of clarity. The security perimeter, visualized as a series of small red dots forming a polygon in Figure 6.8 is marked in green for robot 1's coalition and in blue for robot 27's coalition. Each coalition is responsible for providing sensor coverage to all of these marked virtual perimeter points.

The next stage of the response is executing the optimization task generated at the fuzzy-auction MRTA state (see Table 6.3 and Table 6.4).

6.2.4. Self-Organization using the NSGA-II

In this section, the performance of the NSGA-II will be evaluated for the search of non-dominated solutions, which ensure a maximum coverage of the virtual perimeter and minimize the energy cost in relocating each robot. Table 6.7 describes the parameters used for the NSGA-II.

Optimization Parameters	Value
Number of Target Points	250
Population Size	50
Crossover Probability	0.85
Mutation Probability	0.1
Maximum Pareto Archive Set Size	50
Stop Criterion	Stop after 200 generation of the population

Maximum Robot Response	55 m
------------------------	------

Table 6.7: The optimization parameters used to obtain the results in this section.

As illustrated in Figure 3.1, the EMO Optimization block in the diagram requires optimization criteria and alternatives before optimization can begin. The criteria are fed from the fuzzy-auction MRTA algorithm by generating the optimization task (seen in the previous section). The alternatives represent all possible solutions considered in the scope of this optimization. Thus, the alternatives form the solution space, which the NSGA-II will explore for Pareto optimal solutions.

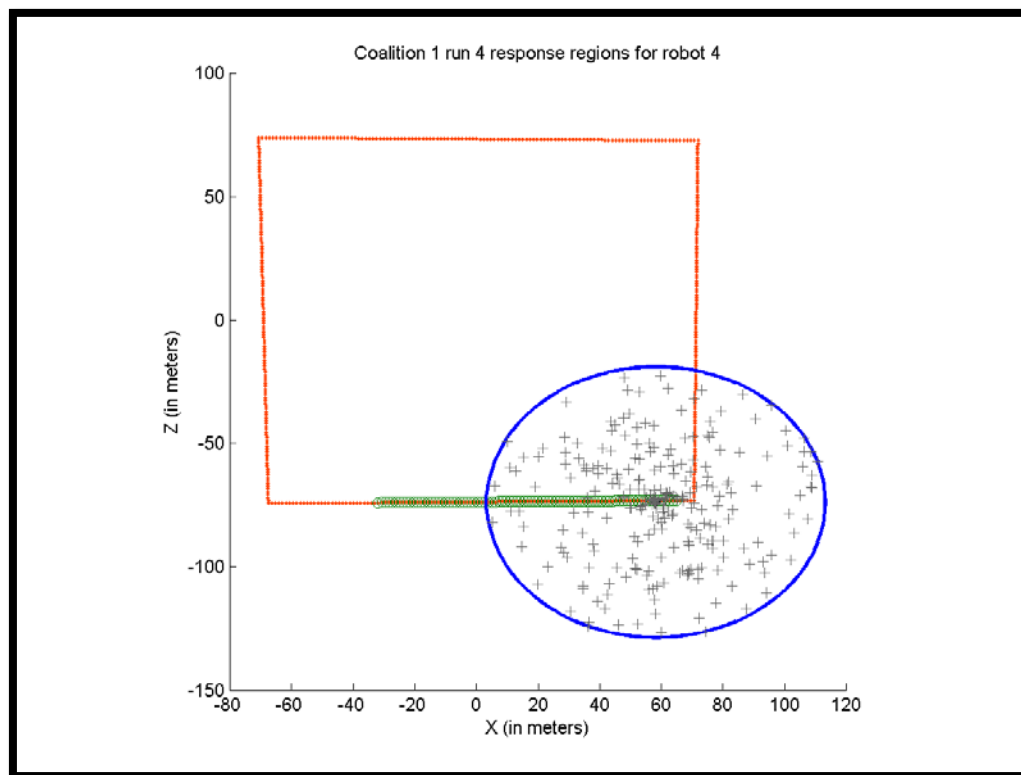


Figure 6.9: Illustration of a single robot's response region. Acceptable positions are represented by crosshair markers

To best illustrate the alternatives for the optimization problem, Figure 6.9 depicts a response region for a single robot within the coalition managed by robot

1. The response region (as per Table 6.7) will use a maximum radius of 55.0 meters. The origin of the response region is fixated to the robot's position. Each coalition is led by the auctioneer robot, which in this case is robot 1 and robot 27 for their respective coalitions. Robot 1 and robot 27 will generate response regions for their coalitions and then execute the optimization algorithm.

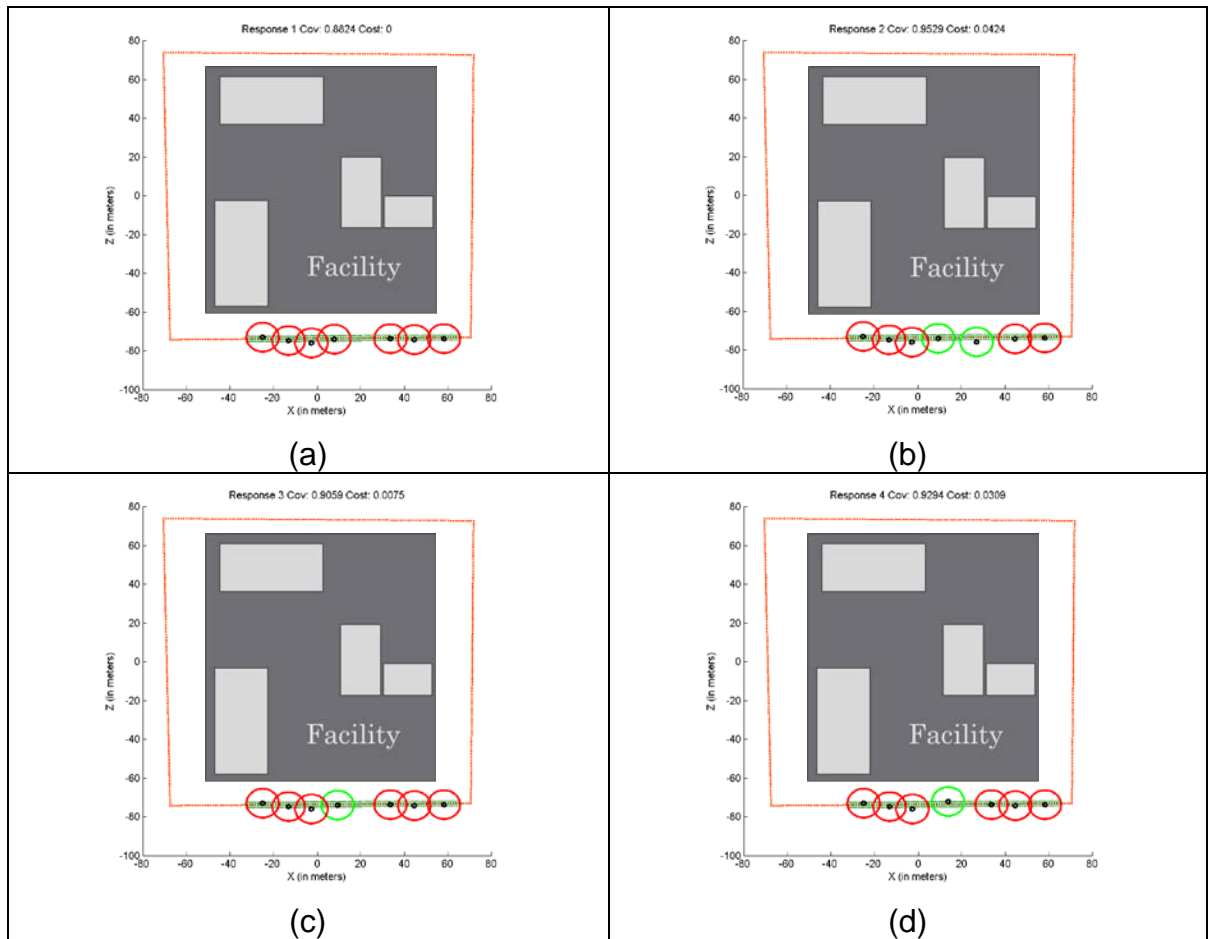
The NSGA-II is executed for 200 generations and produces a Pareto Archive Set (PAS) as shown in Table 6.8.

Solution ID	Robot Configuration	Coverage (%)	Cost (%)	ΔCoverage (%)	$\frac{\Delta \text{Coverage}}{\text{Cost}}$
1	-1,-1,-1,-1,-1,-1,-1	88.24	0.00	0.000	
2	200,-1,-1,175,-1,-1,-1	95.29	4.24	+7.05	1.663
3	200,-1,-1,-1,-1,-1,-1	90.59	0.75	+2.35	3.133
4	131,-1,-1,-1,-1,-1,-1	92.94	3.09	+4.70	1.521
5	119,-1,-1,-1,-1,-1,-1	89.41	0.42	+1.17	2.785
6	200,-1,-1,175,-1,236,-1	96.47	5.45	+8.23	1.510
7	3,-1,-1,-1,-1,-1,-1	91.76	2.35	+3.52	1.498
8	3,-1,-1,175,-1,236,-1	97.65	7.05	+9.41	1.334

Table 6.8: The PAS for robot 1 listing the solutions by ID. The robot configuration indicates the activated target point, except when -1 which implies the robot is not considered in the response. The coverage and the cost are normalized scalar quantities and cost is a normalized scalar quantity.

Figure 6.10 (a) through (h) illustrate the solutions listed in Table 6.8 with solution identifiers 1 through 8, respectively. Each optimization objective is taken with equal consideration. This explains why solution 1 has been added to the PAS, since it completely minimizes the coverage objective, even though it is very unlikely

this would be the selected response. Desirable solutions provide maximum coverage gain but with minimum energy expenditure. Since these are conflicting objectives, the selected response will likely be a compromise between the desirable coverage recovery and minimizing energy usage. For instance, solution 8 provides the maximum coverage recovery, but the coverage-cost ratio raises concerns as the solution is very inefficient. Compared to other solutions the energy used per unit coverage recovered is the highest. Solution 6 provides a more efficient response from the coalition, while sacrificing 1.18% coverage recovery. However, the sensor network's longevity was better preserved.



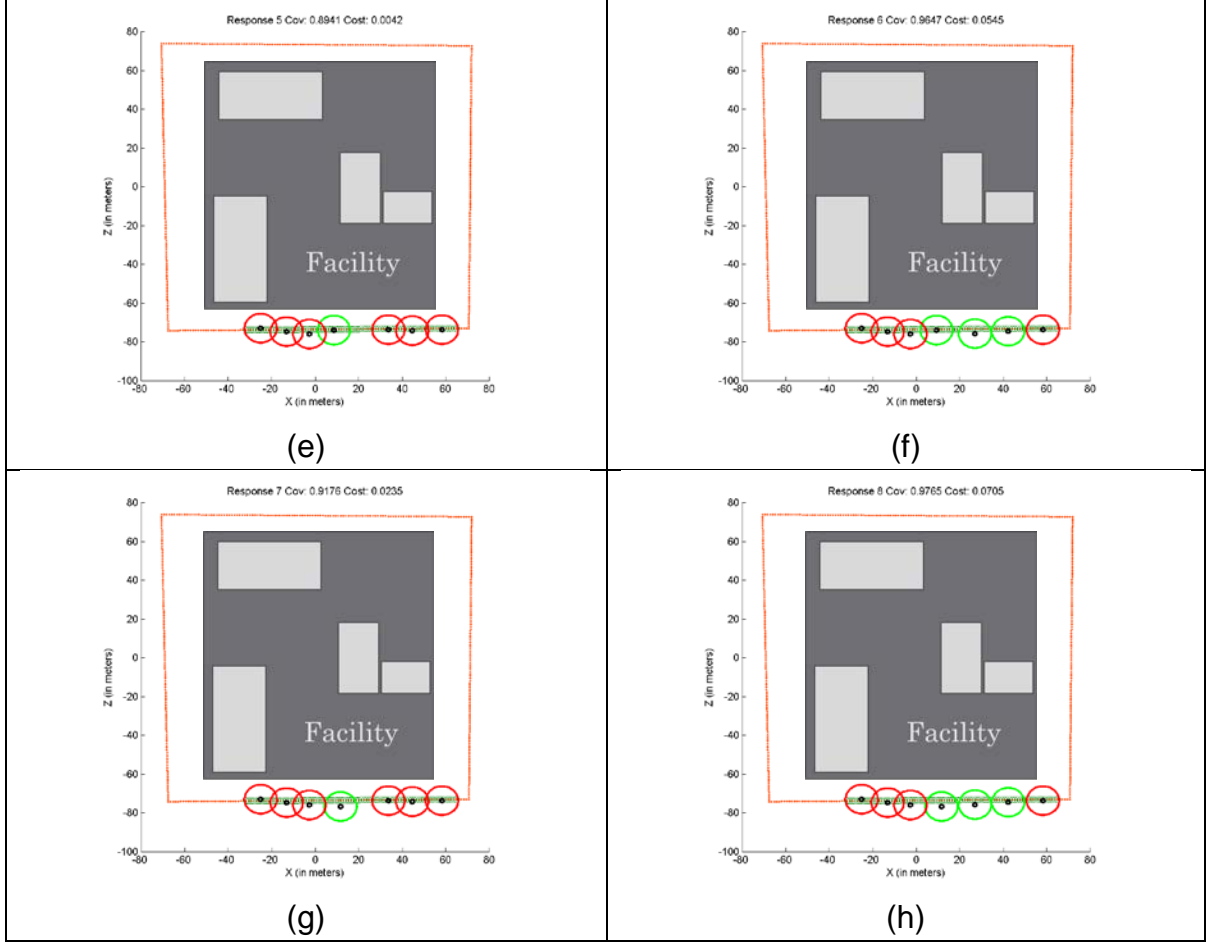


Figure 6.10: PAS Solutions for robot 1 coalition

The decision to choose solution 6 over solution 8 can also be justified by observing the response in Figure 6.10 (f) versus Figure 6.10 (h).

6.3. Chapter Summary

This chapter has evaluated the risk-aware RSN and its major components including: fuzzy risk assessment for RSN in CIP, fuzzy-auction based MRTA, and a self-organization technique using the NSGA-II. This evaluation was conducted using the factory scenario, which allows a large-scale deployment of robotic sensors. This was beneficial to permit multiple risk events varying in space and time and letting the risk-aware RSN handle this high-risk events concurrently. As a robotic member detected a high-risk event, immediately the fuzzy-auction MRTA

technique was able to allocate robotic members of the RSN to form a coalition. As shown in the results, these coalitions are capable of finding feasible solutions to respond to high-risk events. Thus, the risk-aware RSN is capable of perceiving risky situations and evolving its physical topology to safe-guard critical infrastructure.

Chapter 7. Conclusion

7.1. Summary

In this thesis many different technologies in the domain of risk assessment, MRTA, and self-organization have been discussed. Risk assessment conducted by robotic nodes have not been pursued by very many researchers, however a few examples do exist: [9], [10], [13], [14], [16], and [17]. The research conducted by [13] and [14] use a HMM to approach the challenge of effective risk assessment. For a real-time RSN, the requirement of a priori information in the context of the problem may be troublesome. A fuzzy-based approach to risk assessment can allow the system to begin sampling sensor data and extracting risk features based on the fuzzy model. This approach has been explored in this thesis as well as [9], [10], [16], and [17].

The MRTA problem was thoroughly surveyed in [18], where the MRTA problem can be classified into eight separate types. Based on this taxonomy, the work of this thesis falls under the category of ST-MR-IA. This problem is addressed by a multi-robot coordination technique called MURDOCH [19]. This thesis built upon the foundations of MURDOCH focusing attention to the bid evaluation process, named a fuzzy-auction MRTA. This technique effectively forms a coalition of robots, to mitigate a high-risk event. This technique permits the robots in the RSN to operate in a decentralized manner. This alone brings many benefits to the RSN, the most prominent being that the risk-aware RSN is capable of managing itself in the environment. No centralized server is required to delegate operations

and perform computations. During times of crisis in the RSN (i.e., high-risk events) the robotic nodes in the risk-aware RSN can take charge and manage smaller group of robots. These smaller groups are referred to coalitions, which are able to effectively mitigate detected risks in the network.

The self-organization process of the risk-aware RSN uses the NSGA-II to converge on a set of non-dominated solutions. This was an interesting and creative approach to obtaining optimal locations for robotic nodes while meeting the optimization criteria of maximum coverage of coordinates of interest and minimal moving energy expenditure. In Chapter 6, this behaviour of the RSN in the factory scenario met the design objectives, where a set of feasible of solutions were provided by the NSGA-II to maximum coverage of regions of interest while reducing the moving cost of the robots.

7.2. Contributions

This thesis proposed a risk-aware RSN to be used in a CIP scenario. This RSN uses a fuzzy-based risk assessment technique and employs a completely decentralized implementation. Once deployed the risk-aware RSN does not require any external management to assist in the coordination of the robots during a high-risk event. The individual risk-aware RSN members are capable of communicating with one another using a fuzzy-auction MRTA process used to join self-managed robot coalitions. Finally using the NSGA-II and specialized genetic encoding scheme, the robot coalition can seek new robot configuration to mitigate the detected risk. This work has been published in the form of two IEEE conference proceedings [7] and [11] and presented at a poster session [28].

7.3. Future Research

The various components presented in this thesis have currently only been tested in a virtual environment. The system has not yet been subjected to the risks and threats of a real environment. In the simulated world of MRDS, the individual robot sensor require more advanced communication models to better assess the energy cost of transmitting data from one robot to another. In addition, more advanced sensor models would produce more realistic detection. The current implementation utilizes a circular field of view where most real devices would be directional. The response regions of the robotic members, required for the self-organization process were also circular regions. Future work will investigate to define potential responses for robotic assets. Finally, it would be beneficial to begin implementing the described techniques of this thesis on real robots to create a real risk-aware RSN.

References

- [1] E. Whitman, "SOSUS: The "Secret Weapon" of Undersea Surveillance," *Undersea Warfare*, vol. 7, no. 2, 2005.
- [2] Public Safety Canada, "Action Plan for Critical Infrastructure," 2009.
- [3] J. Aubert, T. Schaberreiter, C. Incoul, D. Khadraoui and B. Gateau, "Risk-Based Methodology for Real-Time Security Monitoring of Interdependent Services in Critical Infrastructures," *Availability, Reliability, and Security, ARES '10 International Conference on*, pp. 262-267, 2010.
- [4] L. Coppolino, S. D'Antonio, L. Romano and G. Spagnuolo, "An Intrusion Detection System for Critical Information Infrastructures using Wireless Sensor Network," *Critical Infrastructure (CRIS), 2010 5th International Conference on*, pp. 1-8, 2010.
- [5] L. Gomez and C. Ulmer, "Secure Sensor Networks for Critical Infrastructure Protection," *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pp. 144-150, 2010.
- [6] A. Attwood, M. Merabti, P. Fergus and O. Abuelmaatti, "SCCIR: Smart Cities Critical Infrastructure Response Framework," *Developments in E-systems Engineering (DeSE)*, pp. 460-464, 2011.
- [7] 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," *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2013 IEEE International Conference on*, pp. 132-137, 2013.
- [8] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, pp. 1548-1557, 2001.
- [9] R. Falcon, A. Nayak and R. Abielmona, "An evolving risk management framework for wireless sensor networks," *Computational Intelligence for Measurement Systems and Applications (CIMS), 2011 IEEE International Conference on*, pp. 1-6, 2011.
- [10] R. Falcon and R. Abielmona, "A response-aware risk management framework for search-and-rescue operations," *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1-8, 2012.
- [11] J. McCausland, R. Abielmona, R. Falcon, A.-M. Cretu and E. Petriu, "Auction-Based Node Selection of Optimal and Concurrent

Responses for a Risk-Aware Robotic Sensor Network," *IEEE Int. Symp. Robotic and Sensors Environments (ROSE)*, pp. 136-141, 2013.

- [12] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182-197, 2002.
- [13] L. Paradis and Q. H., "A survey of fault management in wireless sensor networks," *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171-190, 2007.
- [14] X. Tan, Y. Zhang, X. Cui and H. Xi, "Using hidden markov models to evaluate the real-time risks of network," *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on*, pp. 490-493, 2008.
- [15] X. Cheng and Y. Ni, "The Research on Dynamic Risk Assessment Based on Hidden Markov Models," *Computer Science & Service System (CSSS), 2012 International Conference on*, pp. 1106-1109, 2012.
- [16] C. Lv, H. Gong and B. Wang, "Snow disaster risk assessment based on fuzzy comprehensive evaluation," *Geoinformatics, 2010 18th International Conference on*, pp. 1-6, 2010.
- [17] K. Haslum, A. Abraham and S. Knapkog, "DIPS: A Framework for Distributed Intrusion Prediction and Prevention Using Hidden Markov Models and Online Fuzzy Risk Assessment," in *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*, 2007.
- [18] B. P. Gerkey and M. J. Mataric, "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.
- [19] B. P. Gerkey and M. J. Mataric, "Sold!: Auction methods for multirobot coordination," *Robotics and Automation, IEEE Transactions*, pp. 758-768, 2002.
- [20] R. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *Computers, IEEE Transactions on*, vol. 100, no. 12, pp. 1104-1113, 1980.
- [21] I. Mezei, V. Malbasa and I. Stojmenovic, "Auction aggregation protocols for wireless robot-robot coordination," *Ad-Hoc, Mobile and Wireless Networks*, pp. 180-193, 2009.
- [22] C. Tovey, M. Lagoudakis, S. Jain and S. Koenig, "The Generation of Bidding Rules for Auction-based Robot Coordination," *Multi-Robot Systems. From Swarms to Intelligent Automata Volume III*, pp. 3-14, 2005.
- [23] T. Melodia, D. Pompili and V. Gungor, "Communication and Coordination in Wireless Sensor and Actor Networks," *IEEE*

Transactions on Mobile Computing, vol. 6, no. 10, pp. 1116-1129, 2007.

- [24] K. Ma, Y. Zhang and W. Trappe, "Managing the Mobility of a Mobile Sensor Network Using Network Dynamics," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 1, pp. 106-120, 2008.
- [25] J. Lizhong, J. Jia and D. Sun, "Node Distribution Optimization in Mobile Sensor Network Based on Multi-Objective Differential Evolution Algorithm," *Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on*, pp. 51-54, 2010.
- [26] Microsoft, *Microsoft Robotics Developer Studio (Version 4)*.
- [27] P. Selormey, *DotNetMatrix: Simple Matrix Library for .NET*, 2004.
- [28] J. McCausland, R. Abielmona, A.-M. Cretu, R. Falcon and E. Petriu, "A Proactive Risk-Aware Robotic Sensor Network for Critical Infrastructure Protection," in *hSite*, Montreal, 2013.

Appendix A. Fuzzy Membership Declarations

Fuzzy membership functions can be easily defined using a set of simple parameters to represent the characteristic of the function. For membership function, $f(x)$, let the domain of the function be the interval $[-\infty, \infty]$ and the range of the function be the interval $[0,1]$. For a trapezoidal function $f_{trap}(x)$ (see Figure A.1), a set of parameters A, B, C, and D can be used to define the shape of the function. The parameters and the function $f_{trap}(x)$ are related as follows:

$$f_{trap}(x) = \begin{cases} 0 & x \leq A \\ \frac{x - A}{B - A} & A \leq x < B \\ 1 & B \leq x \leq C \\ \frac{D - x}{D - C} & C \leq x \leq D \\ 0 & D \leq x \end{cases}$$

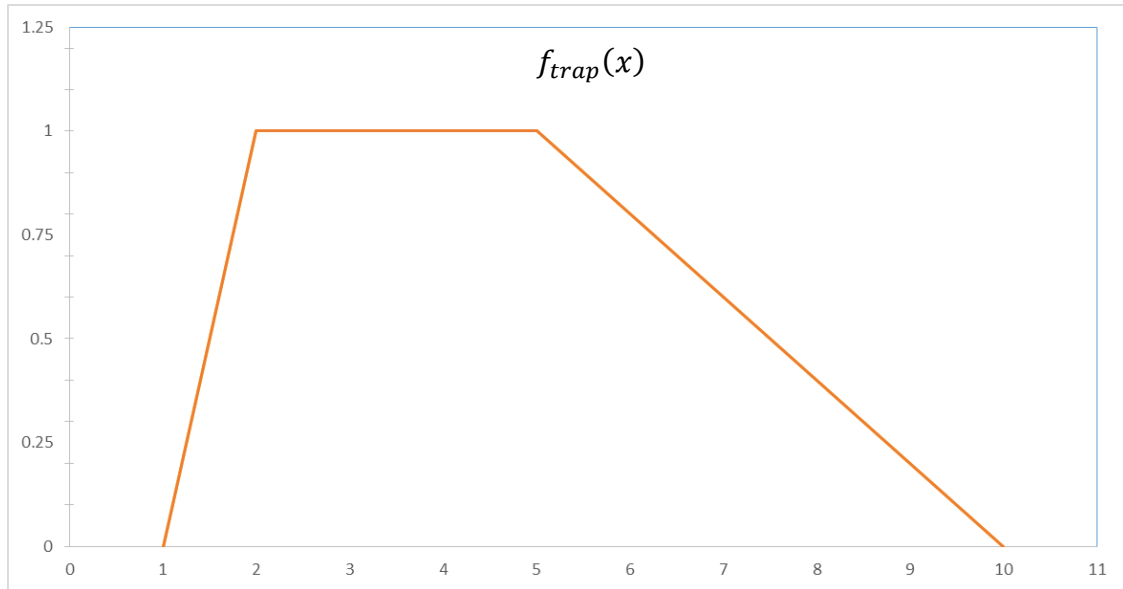


Figure A.1: An example of a trapezoidal membership function declarations with parameters $A=1$, $B=2$, $C=5$, and $D=10$.

For a triangular function $f_{tri}(x)$ (see Figure A.2), a set of parameters A, B, and C can be used to define the shape of the function. The parameters and the function $f_{tri}(x)$ relate as follows:

$$f_{tri}(x) = \begin{cases} 0 & x \leq A \\ \frac{x - A}{B - A} & A \leq x < B \\ \frac{C - x}{C - B} & B \leq x \leq C \\ 0 & C \leq x \end{cases}$$

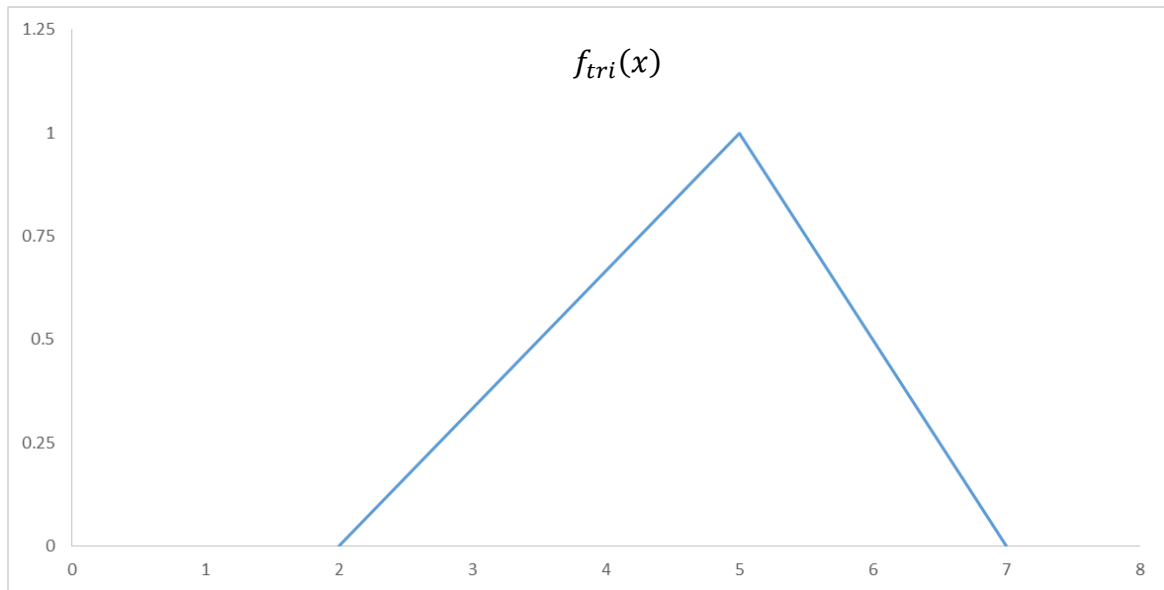


Figure A.2: An example of a triangular membership function declaration with parameters $A=2$, $B=5$, and $C=7$.